SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

# GREEN CLOUD COMPUTING: BALANCING LOAD FOR THE DEPLOYMENT OF SOFTWARE AS A SERVICE

PhD dissertation

School of Business and Social Sciences,
Department of Business Development and Technology
Aarhus University

**Araddhana Arvind Deshmukh**

Aarhus BSS
Aarhus University
Department
**2020**

# GREEN CLOUD COMPUTING: BALANCING LOAD FOR THE DEPLOYMENT OF SOFTWARE AS A SERVICE

A thesis submitted to the
School of Business and Social Sciences,
Department of Business Development and Technology
Aarhus University

In the partial fulfillment of requirement for the degree of

**DOCTOR OF PHILOSOPHY**


**AUGUST 2020**


## Araddhana Arvind Deshmukh

AARHUS UNIVERSITY

CTIF Global Capsule,
Department of Business Development and Technology,
School of Business and Social Sciences
Aarhus University, Herning, Denmark


AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES AARHUS UNIVERSITET

Thesis submitted:

PhD Supervisor: Dr. Albena Mihovska,
CTIF Global Capsule (CGC),
Department of Business Development and Technology, Aarhus University,
Herning, Denmark.

Co-Supervisor: Dr. Ramjee Prasad,
CTIF Global Capsule (CGC), Department of Business Development and Technology, Aarhus University,
Herning, Denmark.

PhD Committee: Sofoklis Kyriazakos,
Chairman,
Associate professor, Department of Business Development and Technology, Aarhus University
Professor, Vladimir Poulkov,
Technological University of Sofia
Professor, Walter Konhäuser,
TU BERLIN

PhD Series: Future Technologies for Business Ecosystem Innovation (FT4BI), Department of Business Development and Technology (BTECH), School of Business and Social Sciences, Aarhus University, Herning, Denmark



Aarhus BSS
Aarhus University
Department of Business Development and Technology
AUGUST 2020

**DEDICATED**

**TO**

**MY 'BABA' (FATHER) – LATE MR. ARVIND M. DESHMUKH**

**&**

**MY 'MA' (MOTHER) MANISHA ARVIND DESHMUKH**

# ABOUT THE AUTHOR

Araddhana Arvind Deshmukh graduated in Computer Engineering in 2004 from Vishwakarma Institute of Technology, Pune, Maharashtra (India) and received Masters from Pune Institute of Computer Technology in 2011, also one more Masters in Arts with specialization in industrial economics. She has completed AMIE degree of Computer Science Engineering from Institution of Engineers (India) with 2nd Rank holder. She had started pursuing Doctor of Philosophy (PhD) since March 2011, in the field of Wireless Communications at Aalborg University, Denmark. Later, she continued pursuing PhD from Aarhus University, Denmark.

She has been with VIT Engineering College Pune from 1999 to 2005 as Technical support, Network Administrator as well as Lecturer. She also worked with PVGCOET Engineering College, Pune as an Assistant Professor in Department of Computer Engineering, from 2005 to 2007. She has joined Sinhgad Institute, Smt. Kashibai Navale College of Engineering in 2007. Araddhana has 15 years of teaching experience and 6 years as other experience.

She has published 120 papers in reputed International Journals, Conferences etc. She has authored of 5 books on 'Cloud Computing', 'Human Computer Interaction', 'Object Oriented Programming', 'Distributed Systems', 'Principles of Programming Languages'. She is a reviewer of Elsevier, WPC, IJCSNS, IJCA, IJCSI, and in many international conferences like WIC 2013, 14,15,16,18, ICINC 2016, 18,19 and IETE journal. She has instrumental in organization of various workshops and short term training programs for students and faculties. She has delivered several talk on "Cloud Computing", "Next generation networking" In different institutes and engineering colleges. She received funding for research from BCUD, Pune and AICTE, India. She has also filed one patent in India.

She is a member of IEI(India),UACEE(USA), IAENG(Hong Kong),CSI (Pune), Currently she is unanimously elected as Secretary in the Institution of Engineers ( India) Pune Local Centre. Her area of research is Cloud Computing, Cloud Security, High Performance Computing, and Artificial Intelligence.

# DANSK RESUME

Araddhana Arvind Deshmukh uddannede sig inden for computerteknologi i 2004 fra Vishwakarma Institute of Technology, Pune, Maharashtra (Indien) og modtog kandidater fra Pune Institute of Computer Technology i 2011, også en kandidatgrad i kunst med speciale i industriel økonomi. Hun har afsluttet AMIE-grad af computervidenskabsteknologi fra Institution of Engineers (Indien) med indehaveren af 2. placering. . Hun var begyndt at forfølge doktor i filosofi (ph.d.) siden marts 2011 inden for trådløs kommunikation ved Aalborg Universitet. Senere fortsatte hun ph.d. fra Aarhus Universitet.

Hun har været hos VIT Engineering College Pune fra 1999 til 2005 som teknisk support, netværksadministrator og lektor. Hun arbejdede også med PVGCOET Engineering College som adjunkt ved Institut for Computer Engineering fra 2005 til 2007. Hun er ansat i Sinhgad Institute, Smt. Kashibai Navale College of Engineering i 2007. Araddhana har 15 års undervisningserfaring og 6 år som anden erfaring.

Hun har offentliggjort 118 artikler i anerkendte internationale tidsskrifter, konferencer osv. Hun har forfatter til 2 bøger om 'Cloud Computing', 'Human Computer Interaction', 'Object Oriented Programming', 'Distributed Systems', 'Principles of Programming Languages'. Hun er en anmelder af Elsevier, IJCSNS, IJCA, IJCSI og i mange internationale konferencer som WIC 2013,14,15,16,18, ICINC 2016,18,19og IETE tidsskrift. Hun har været med til at organisere forskellige workshops og kortvarige træningsprogrammer for studerende og fakulteter. Hun har holdt flere foredrag om "Cloud Computing", "Next generation Networking" på forskellige institutter og ingeniørhøjskoler. Hun modtog finansiering til forskning fra BCUD, Pune og AICTE, Indien. Hun har også indgivet et patent i Indien.

Hun er medlem af IEI (Indien), UACEE (USA), IAENG (Hongkong), CSI (Pune). Hendes forskningsområde er Cloud Computing, Cloud Security, High Performance Computing, Kunstig intelligens.

# ENGLISH ABSTRACT

Every day there is more need for computing and storage devices. Enormous amounts of information are produced and traded over the system, which further requires the need of increasingly processing assets. Virtualization of resources and cloud computing have become key enabling technologies.

Cloud computing is integrating personal computers into a single global computer. The term cloud computing stands for the type of service that allows any organization to deliver an application or service to their employee, customer, or end user through the Internet. The term cloud computing remains for the sort of administration that enables any association to convey an application or administration to their worker, client through the Internet.

There is an extraordinary relationship between the moves from the idea of customary PCs to cloud PCs. There is a great analogy between the shifts from the concept of traditional computers to cloud computers.
A large portion of endeavours are aimed at decreasing the computing cost through the methods of virtualization. Cloud Registering offers better processing through improved use and diminished organization and framework costs.

Due to the usage and expansion of web applications, dynamic manipulations such as online banking, online shopping, e-commerce and various service web sites, reason in ever-growing server resource load and overhead. Load alludes to the site traffic as well as it incorporates CPU load, network load and memory limit of every server. A load balancing method ensures that every framework in the system has same amount of work at any moment of time. This suggests neither any of them is unnecessarily over-stacked, nor under-used.

Load balancing circulates information relying on how demanding every server or hub is. Without a load balancer, the customer must pause while his procedure gets prepared, which may be excessively tiring and demotivating for him.

The load balancer handles different data by managing a queue, the job arrival rate of the CPU processing, and so forth. Without a proper use of a load balancer loss in information may occur.

Software as a service (SaaS) is a newly emerging computing paradigm, where the remote and virtualized computing resources are utilized by users by means of software access to the Internet. Cloud platforms offer resource utilization as on-demand service, which lays the foundation for applications to scale during runtime. However, just-in time scalability is not achieved by simply deploying applications to cloud platforms. Existing approaches require developers to rewrite their applications to leverage the on-demand resource utilization, thus bind applications to specific cloud infrastructure. So the single server is no longer able to balance the growing number of service requests, and a more expensive server with better performance with the help of cloud computing can be used instead.

This research work proposes novel ubiquitous strategy for load balancing useful to scale different types of applications. This strategy-based approach automates the deployment and scaling of applications in the cloud. Just-in-time scalability is achieved without binding to specific cloud infrastructure. A real case will be used to demonstrate the process and feasibility of this strategy-based approach.
The Cloud has helped endeavors influence the advantages of figuring assets, which are shared over a virtualized domain. A ton of ventures are now utilizing cloud-based administrations in either structure. This conveys us to the idea of burden balancing in the cloud.
One issue is how to implement an algorithm for load balancing with dynamic services. But very few results are available related to the virtualization of the content aware traffic, utilization, storage, name space and allocation.

This research will bring about with the help of simulation and using mathematical and analytical modelling novel approach to virtualized load balancing. A methodology for handling the following issues related to load balancing will be developed: 1. Content aware traffic 2. Service utilization 3. Name space virtualization. The envisioned methodology will automate the deployment and scaling of the applications in the cloud. The SaaS is also reliable and takes care of secured computing schemes, which have combined properties of threshold group-oriented signature schemes and multi-signature scheme. To ensure the load-balanced data remains secure with threshold multisignature. At the time of subscribing by cloud user's computation resources, it is of critical importance to collaborate and contribute equally to produce a valid multiparty signature.

*Keywords: cloud computing, virtualization, data security, content aware traffic,*

# Dansk SUMMARY

Hver dag er der mere behov for computere og lagerenheder. Enorme mængder information produceres og handles over systemet, hvilket yderligere kræver behovet for i stigende grad at behandle aktiver. Virtualisering af ressourcer og cloud computing er blevet nøglen muliggør teknologier.

Cloud computing integrerer personlige computere i en enkelt global computer. Udtrykket 'cloud computing' står for den type tjeneste, der giver enhver organisation mulighed for at levere en applikation eller service til deres medarbejder, kunde eller slutbruger via Internettet. Udtrykket cloud computing forbliver for den form for administration, der gør det muligt for enhver forening at formidle en applikation eller administration til deres arbejdstager, klient via Internettet.

Der er et ekstraordinært forhold mellem bevægelserne fra ideen om sædvanlige pc'er til sky pc'er. Der er en stor analogi mellem skiftet fra konceptet traditionelle computere til skycomputere.
En stor del af bestræbelserne er rettet mod at reducere beregningsomkostningerne ved hjælp af virtualiseringsmetoder. Cloudregistrering tilbyder bedre behandling gennem forbedret brug og formindskede organisations- og rammeomkostninger.

På grund af brugen og udvidelsen af webapplikationer forårsager dynamiske manipulationer såsom online bank, online shopping, e-handel og forskellige servicewebsteder, stadig større serverbelastning og overhead. Belastning henviser til stedstrafikken såvel som det indbefatter CPU-belastning, netværksbelastning og hukommelsesgrænse på hver server. En belastningsbalanceringsmetode sikrer, at alle rammer i systemet har samme mængde arbejde på ethvert tidspunkt. Dette indebærer, at hverken af dem er unødvendigt overdreven eller heller ikke underbrugt.

Belastningsbalancering cirkulerer oplysninger, der er afhængige af, hvor krævende hver server eller hub er. Uden en belastningsafbalancering skal kunden holde pause, mens hans procedure forberedes, hvilket kan være for træt og demotiverende for ham.

Belastningsafbalanceren håndterer forskellige data ved at styre en kø, jobprisen for CPU-behandlingen osv. Uden en ordentlig brug af et belastningsbalancer kan der opstå tab i information.

Software as a service (SaaS) er et nyopstået databehandlingsparadigme, hvor de eksterne og virtualiserede databehandlingsressourcer udnyttes af brugerne ved hjælp af softwareadgang til Internettet. Cloud-platforme tilbyder ressourceudnyttelse som on-demand-service, der lægger grunden til, at applikationer skaleres i løbet af runtime. Imidlertid opnås ikke-i-tiden skalerbarhed ved blot at implementere applikationer til skyplatforme. Eksisterende tilgange kræver, at udviklere omskriver deres applikationer for at udnytte ressourceudnyttelsen on-demand og dermed binde applikationer til specifik sky-infrastruktur. Så den enkelte server er ikke længere i stand til at afbalancere det stigende antal serviceanmodninger, og en dyrere server med bedre ydelse ved hjælp af cloud computing kan i stedet bruges.

Dette forskningsarbejde foreslår en ny allestedsnærværende strategi til belastningsbalancering, der er nyttig til at skalere forskellige typer applikationer. Denne strategibaserede tilgang automatiserer implementering og skalering af applikationer i skyen. Just-in-time skalerbarhed opnås uden binding til specifik sky-infrastruktur. En reel sag vil blive brugt til at demonstrere processen og gennemførligheden af denne strategibaserede tilgang.
Skyen har hjulpet bestræbelser på at påvirke fordelene ved at finde aktiver, der deles over et virtualiseret domæne. Et ton venturer bruger nu skybaserede administrationer i begge strukturer. Dette formidler os ideen om byrdebalancering i skyen.

Et spørgsmål er, hvordan man implementerer en algoritme til belastningsbalancering med dynamiske tjenester. Men meget få resultater er tilgængelige relateret til virtualisering af den indholdskendte trafik, anvendelse, opbevaring, navneplads og tildeling.

Denne forskning vil skabe ved hjælp af simulering og ved hjælp af matematisk og analytisk modellering af nye metoder til virtualiseret belastningsbalance. En metode til håndtering af følgende problemer, der er relateret til belastningsbalancering, vil blive udviklet: 1. Indholdsbevidst trafik 2. Serviceudnyttelse 3. Navn på plads virtualisering. Den planlagte metode vil automatisere implementeringen og skaleringen af applikationerne i skyen. SaaS er også pålidelig og tager sig af sikrede computerskemaer, der har kombinerede egenskaber ved tærskelgruppeorienterede signaturskemaer og multisignaturskema. for at sikre, at de belastningsbalancerede data forbliver sikre med tærskel-multisignatur. På tidspunktet for abonnement fra skybrugerens beregningsressourcer er det af kritisk betydning at samarbejde og bidrage lige så til at producere en gyldig flersportsignatur.

Nøgleord: cloud computing, virtualisering, datasikkerhed, indholdsbevidst trafik,

# ACKNOWLEDGEMENTS

In Indian philosophy, God, Guru (teacher), family and friends are the three principle factors behind the success of any endeavor. First and foremost, I wish to thank the Almighty God Ganesh, who has guided me through the good and bad periods.

I express my deep sense of gratitude and respect to Dr. Albena Mihovska as my supervisor. It was not possible to complete the work without her support and guidance. My thank goes to her for keenly observing and participating in my progress. She helped me in the formulation of the problem and deciding the path. Dr. Albena Mihovska is an exceptionally enthusiastic path director with a positive air. In spite of her bustling timetable, she was constantly accessible to illustrate my questions. She has upheld exceptionally well amid my high points and low points in the voyage of the Ph.D. program. I consider it as a stunning chance to do my Ph.D. under her important direction and to gain from her exploration skill. Much obliged to you, Dr. Albena, for all your genuine help and support.

I am very much blessed to have Prof. Ramjee Prasad, Founder Chairman GISFI, and Founder President CGC Aarhus University, Denmark as my Co-supervisor. I took in a ton from him, and my learning was not constrained to the theory work. He instructed me the estimation of diligent work and the correct importance of research morals.

I am appreciative to the administration of STES, Pune. I am obliged to Hon. Organizer President of STES, Prof. M. N. Navale, Founder Secretary of STES, Dr. Mrs. S. M. Navale, Vice President (HR), Mr. Rohit M. Navale, Vice President (Admin), and Mrs. Rachana Navale-Ashtekar. The Institution has given a doorstep chance to improve specialized information and overhaul the capability with full subsidizing.

I offer my regard to Dr. A.V. Deshpande, Dr. S. S. Inamdar and Dr. S. D. Markande - the wellspring of motivation, Dr. M.S. Gaikwad - solid supporter, and Dr. K. R. Borole - Motivator, and well-wisher and all my GISFI individual associates .My special thanks to Mrs. Jyoti Prasad, Mr. Rajiv Prasad and Mrs. Mayuri Prasad for making my visit much comfortable with their love and support. Their affection and care were memorable.

My exceptional gratitude to Mrs. Jyoti Prasad, Mr. Rajiv Prasad and Mrs. Mayuri Prasad for making my visit much alright with their affection and backing .Their fondness and care were paramount.

At last, I recognize favors of dad late Shri Arvind, mother Manisha Deshmukh, for their consideration. I perceive the consolation, and good help, given by my sisters Manjusha and Sandhya with jiju Vivek Kulkarni and Vivek Giridhar I unique gratitude to my nephew Nachiket, Saket, Aniket, Amruta, Shweta and my naughty granddaughter ANVII from the base of the heart, their smile dependably bond me to leave challenges – Good Motivators.

My exceptional much appreciation goes to Prof.Mrs.M.A.Shukla and Dr. N.P.Kulkarni for their valuable appeal, predictable help, and backing. I similarly thank all my specialized topic partners at SKNCOE, Pune for their motivation, kind help. Last anyway not the base, I should need to thank all help drew in with the voyage of proposition and research work

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ACRONYMS

| Acronym | Expression |
|---|---|
| VM | Virtual Machines |
| SLB | Static Load Balancing |
| DLB | Dynamic Load Balancing |
| CIS | Cloud Information Services |
| VMM | Virtual Machine Monitor |
| CPU | Central Processing Unit |
| I/O | Input Output |
| SaaS | Software As a Service |
| IaaS | Infrastructure as a Service |
| etc | excetra |
| QoS | Quality of Service |
| PaaS | Platform as a Service |
| VMMC | Virtual Machine Monitor Control |
| St | Overall smoothing |
| Bt | Trend Component |
| Rt | Estimate of the Depersonalize Level |
| Gt | Estimate of the trend |
| VoD | Video on Demand |
| CDN | Content Delivery Network |
| SLA | Service Level Agreement |
| Awt | Average Waiting Time |
| OS | Operating Systems |
| GUI | Graphics User Interface |
| LAN | Local Area Network |

| | |
|---|---|
| VDI | Virtual Desktop Infrastructure |
| AUT | Application Under Test |
| Mu | No. of Monitors Requested |
| $M_{max}$ | Maximum no of Monitors per instance |
| Mk | Number of active monitors on instance |
| RAM | Random Access Memory |
| $RAM_k$ | Ram of k'th instance |
| $CPU_k$ | CPU usage of k'th instance |
| StaaS | Storage as a Service |
| CaaS | Computation as a Service |
| ES | Exponential Smoothing |
| CCA | Canonical Correlation Analysis |
| CASP | Content Aware Service Provider |
| CAP | Content Aware Provider |
| TPA | Third Party Auditor |
| PDP | Provable Data Possession |
| PoR | Proof of Retrievability |
| CSP | Cloud Service Provider |
| CSS | Cloud Storage Server |
| CS | Cloud Server |
| ZKP | Zero Knowledge Proof |
| CRC | Cyclic Redundancy Check |
| ECC | Elliptical Curve Cryptography |
| CA | Computer Associates |
| NLB | Network Load Balancer |
| SLB | Server Load Balancer |

# LIST OF PUBLICATIONS

1. Araddhana A. Deshmukh, Dr. Albena D. Mihovska, Dr. Ramjee Prasad, “Zoneminder as ‘Software as a Service’ and Load Balancing of Video Surveillance Requests”, Networking and Electronic Commerce Research Conference 2012 (NAEC 2012), http://www.atsma.org/Conferences/NAEC2012,Italy,Riva del Garda Italy, October 11-14 2012
2. Araddhana A. Deshmukh, Dr. Albena D. Mihovska, Dr. Ramjee Prasad “Cloud computing security schemes:- TGOS [Threshold Group-Oriented signature ] and TMS [Threshold multisignature”, 2 nd World Congress on Information and Communication Technology http://www.mirlabs.org/wict12/#cfp, WICT 2012, Trivandrum India, October 30- November 2, 2012 [ The paper also shortlisted by www.mirlabs.net/WICT 2012 for journal http://www.mirlabs.net/wict12/#journal .]
3. Araddhana A. Deshmukh, Dr. Albena D. Mihovska, Dr. Ramjee Prasad, “Optimized Resource Stipulation for Video on Demand Via Zoneminder”, Global Wireless Summit 2014 in Aalborg from May 11-14, 2014 Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2014 4th International Conference on, Aalborg, Denmark, May 11-14, 2014 [Based on the GWS2014 paper we have been invited to submit a journal paper in the Special issue on "Advances in Control for the provision of Smart Services", which will be published on the novel CONASENSE journal ]
4. Araddhana A. Deshmukh, Pankaj Chaudhary , ,Dr. Albena Mihovska, Dr. Ramjee Prasad,’ NUNI ( New User New Item) Problem for SRSs Using Content aware multimedia based Approach’ 3rd International Conference on Innovations in Computer Science & Engineering Hosted by: Guru Nanak Institutions In Association with CSI Hyderabad Chapter, Division 5, Education and Research, CSI –India Dates: 7th – 8th August 2015, Venue: Guru Nanak Institutions, Hyderabad, India, Springer publication, ://www.springer.com/series/11156
5. Dipti Madankar , Araddhana A. Deshmukh, Dr. Albena Mihovska , Vaishali Maheskar , ‘ SQAMPS: Secure Quorum Architecture for Mobile Presence Services with End-to-End Encryption, WCI '15 Proceedings of the Third International Symposium on Women in Computing and Informatics , August 10-13, 2015, Kochi, India, ACM ISBN 978-1- 4-4503-3361-0/15/0
6. Shubhangi Opale , ,Araddhana A. Deshmukh, Dr. Albena Mihovska, Dr. Ramjee Prasad , ”'Privacy-Preserving Friend Matching Protocol approach for Pre-match in Social Networks”, WPMC 2015 18th International Symposium on Wireless Personal Multimedia Communications, Global Wireless Summit 2015, Hyderabad , Dec 13-16, 2015
7. Araddhana A. Deshmukh,Snehal Sawant ,Dr. Albena Mihovska, Dr. Ramjee Prasad ,”'Public Auditing and Data Dynamics in Cloud with Performance Assessment on Third Party Auditor'” WVITAE 2015, 5th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), Global Wireless Summit 2015, Hyderabad , Dec 13-16, 2015
8. Araddhana A. Deshmukh, Reshma Gade, Dr. Albena Mihovska, Dr. Ramjee Prasad,” Inference attack on URL visiting history of the social networking”, Springer International Conference on Intelligent Computing and Application, ICICA 2016, Department of computer Engineering, D.Y.Patil College of Engineering, Akurdi, Pune44
9. Araddhana A. Deshmukh, Reshma Gade, Dr. Albena Mihovska ,Dr. Ramjee Prasad,” Inference attack on URL visiting history of the social networking”, International Journal of Control Theory and Applications., I J C T A, 9(10), 2016, pp. 1-11 © International Science Press, ISSN: 0974-5572, PP 1111-1118, Dec 2016

10. Araddhana A. Deshmukh, Dr.Albena Mihovska, Dr. Ramjee Prasad," Cloud Security using COSDACC [ Cluster of Servers and Access Control], IEEE 5G Summit and 30th GISFI Standardization Series Meeting, ,SRM University, Kattankulathur, Chennai. *Also shortlisted for Conasense Journal*, 16-17 November 2017

11. Araddhana A. Deshmukh, Smita Ashok Yeole, Dr.Albena Mihovska, Dr. Ramjee Prasad, "An Approach: A Novel Protected Data Deduplication Scheme with Dynamic Ownership Management to Secure Cloud Data", International Journal of Pure and Applied Mathematics, Volume 118 No. 24 2018, ISSN: 1314-3395 (on-line version),url: http://www.acadpubl.eu/hub/Special Issue,http://www.acadpubl.eu/hub/

12. Araddhana A. Deshmukh, Dr.Albena Mihovska, Dr. Ramjee Prasad," Cloud Security using COSDACC [ Cluster of Servers and Access Control], IEEE 5G Summit and 30th GISFI Standardization Series Meeting, ,SRM University, Kattankulathur, Chennai. *Also shortlisted for Conasense Journal*, 16-17 November 2017

13. Araddhana A. Deshmukh, Dr.Albena Mihovska, Dr. Ramjee Prasad" Secured Outsourced Integrity Test for Cloud Storage", Fourth International conference on Information and Communication Technology for Sustainable Development ( ICT4SD 2019) with World Publication and Conference Summit PubCon2019 Goa, India, as per the guidelines of Springer, 5-6 July 2019 ISBN: 978-981-15-0936-0

14. Aradhana A. Deshmukh, Dr. Girish S. Mundada, Dr. Albena Mihovska, " An Amendment of 3T (Technology, Trend, Target) using IT (Indian Talent) for Indian Engineering: A vital technocracy of Smart City", 34th Indian Engineering Congress, Hyderabad International convention Centre, NOVOTEL Complex, Hyderabad, Dec 27-29, 2019 PP 325-331 ISBN 978-811-942561-5-1

15. Aradhana A. Deshmukh, Albena Mihovska , Ramjee Prasad, "Secure Outsourced Integrity Test For Cloud Storage", Advances in Intelligent Systems and Computing", 1077 ICT Systems and Sustainability, http://www.springer.com/series/11156, Proceedings of ICT4SD 2019, Volume 1, ISSN 2194-5357 ISSN 2194-5365 (electronic) Advances in Intelligent Systems and Computing ISBN 978-981-15-0935-3 ISBN 978-981-15-0936-0 (eBook), https://doi.org/10.1007/978-981-15-0936-0, © Springer Nature Singapore Pte Ltd. 2020, PP 309-320

16. Aradhana A. Deshmukh, Albena Mihovska , Ramjee Prasad," Anatomization and Perception of Mental Disorder Because Usage of Online Social Network Data", *"International Symposium on 5G & Beyond for Rural Upliftment*" BIT Sindri, IIT(ISM) Dhanbad, the IEEE 5G summit and 35th GISFI Standardization Series Meeting (GSSM) *on February 8-9.*2020, https://www.riverpublishers.com/pdf/ebook/chapter/RP_9788770222174C17.pdf

# CHAPTER 1 - INTRODUCTION

*"Everything is going to be connected to Cloud and data. All of this will be mediated by software"- Satya Nadella-1967*

गतेर्भंग: स्वरो हीनो गात्रे स्वेदो महद्भयम्ज्ञन्ब्स्प; |
मरणे यानि चि*नानि तानि चि*नानि याचके ||

Losing balance while walking, talking in low voice (not able to talk properly), sweating, and fear, are signs found in a person who is about to die, same signs are found in yAchaka, i.e. a person who is asking help from others (a person who is dependent on others). Balancing is necessity in every step of life. [ Subhashit 208, Rigveda].

This Chapter shows the typical state outline of the thesis. It clarifies the ideas and method of the thesis by research commitments and research questions. The section additionally defines the hypothesis of work. The Chapter closes by giving a framework review of the existing chapters.

### 1.1 PROBLEM BACKGROUND

Today, the Internet offers enormous measure of information with administrations accessible at no cost to clients. Advances in distributed and cloud computing have allowed for giving cloud-based administrations. Cloud storage capabilities are an innovative way to achieve more capacity, which is essential in a world of rapidly increasing digital data in all spheres of life. The cloud enables the availability of sufficient computing requirements as a service to a multitude of diverse end users and has become an integral part of data processing and storage for both, business and private users. However, larger amounts of information to be processed also demand more resources, which if not managed efficiently lead to a heavier carbon footprint and increased operational costs.

Virtualization is a key element of cloud computing allowing for maximization of the resources, while reducing the number of physical machines, by running multiple systems and applications on the same hardware. Virtualization, thus, contributes to the flexibility, multitenancy and scalability of cloud computing. However, the capabilities of sharing and migration make security an open challenge in virtualization and cloud computing.

Load balancing allows for smooth operation without overloading/under loading the virtual machines of the cloud system, which is crucial when large amounts of data need to be processed at the same time. Load balancing, thus, is crucial for the availability of cloud-based

applications, for fast responses to service requests and for preventing downtime of the server, which are critical for maintaining the service level agreements (SLA).

The research leading to this PhD dissertation focuses on the design and evaluation of energy efficient load balancing techniques to be applied in support of cloud-based operations and for sustaining faster response times to service requests, maintaining the required SLAs and security requirements, and for serving clients as possible in an energy-efficient manner.

### 1.1.1 THE GENERAL CONCEPT OF CLOUD COMPUTING

There have been many definitions of cloud computing by different researchers. Barkley RAD defines cloud computing as: “Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centre that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The data center hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, we call it a Public Cloud; the service being sold is Utility Computing. We use the term Private Cloud to refer to internal data centre of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds. People can be users or providers of SaaS, or users or providers of Utility Computing.” [27]

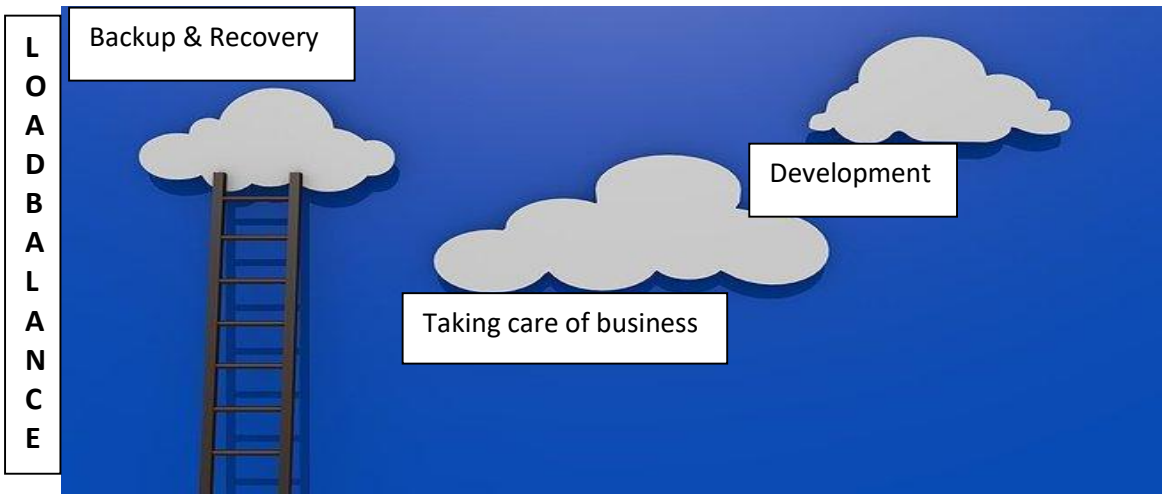



FIGURE 1.1 FUTURE APPLICATIONS OF CLOUD COMPUTING

Cloud computing as shown in Figure 1.1 is a future application of cloud computing, in which dynamic, versatile and virtual resources are given over the Internet, with huge data storage. Cloud computing alludes to administrations that give normal business applications on the web, which are obtained from a Web program, while the product and information are put

away on the servers. To satisfy 'n' demands from 'n' number of clients for 'n' servers and group of systems, requires exact and proficient balancing, this can be by means of load balancing [7,8]. Load balancing algorithms distribute resources and workloads across various servers, thus enabling more virtualized capacity for handling any request yielding, among others, also a faster response time. Cloud computing provides Database as a Service (DaaS), Expert as a Service (EaaS) , Storage as a Service (SaaS), Network as a Service (NaaS), Security as a Service (SecaaS), Communication as a Service (CaaS), Monitoring as a Service (MaaS), and Testing-as-a-service (TaaS), which are accessed by the user for different application purposes [3].

A cloud comprises of a few components, for example, customers, server farm and appropriate servers. A load balancer allows for high-quality performance and applications.
A load balancer has many assessments in cloud computing like Server Performance Advisor [ SPA ] provided by Microsoft, Performance Hyper v cluster test , seesaw test by Google, HAProxy in Github, Zevenet  etc. However, in the existing performance tests there were very few load balancers that test security issues such as integrity, agility, and so forth.  Chapter 6 deals with the testing of security issues. In the continuation other concerns are also equally biased like steps i.e. on-request resource designation, resource use, resource synchronization, resource pooling with adaptable metered administrations [28].

Today the main challenge faced by organizations, which have moved their resources to the cloud is how to track and keep intact their information there, in particular due to the continuous increase of available information, also coming from the current digitization trend. IBM says, that 90% of the information on the planet today has been made over the most recent two years alone and that is on the grounds that consistently we are making around 2.5 quintillion bytes of information [27, 28].

There are information sources everywhere: Internet-based life posts, purchase exchange records, mobile phones GPS flag, blog entries, advanced pictures and recordings and environmental sensors are just few examples. These huge data when ordered, examined and sorted out are able to reveal a lot of personal and confidential information, and may have significance for the protection of the critical infrastructure.

It is important to differentiate and understand the process of backup and recovery, with both having its own unique characteristics. Backup, or ‘reinforcement’ is only a duplicate of a document or data put away some place in the cloud (e.g., any non-basic information, such a PowerPoint, etc) and that can be reproduced.

Cloud information recovery (also, known as ‘recuperation’), incorporates making a framework that is equipped for discovering data and replicating them, when required. It incorporates backing up basic information like client data and reestablishing the data if necessary. In terms of the current trend of frequent digital threats/attacks, the cloud information recuperation is of most extreme significance. Organizations do not become acquainted that they are hacked until days or weeks after the fact. Controllers anticipate that they should recover the records; however the proper separation of the data of one client from a crowd of millions requires complex processing.

Recuperation and reinforcement are main challenges for the IT groups. Distributed computing has its very own arrangement of difficulties and the major being reclamation of data that endeavors store in the open cloud. Because of this, open cloud spending is swelling at a high rate.

Cloud services benefits that are being built today are located in storehouses because of the fact that the market is moderately new and still needs development. The current most popular cloud computing providers are Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, with a trend towards multi-cloud platforms with the recent acquirement of Redhat by IBM.

There are three key attributes related to cloud computing that have been considered for the proposed here research:

- On Demand: Resources can be stipulated immediately whenever essential. Resources will be unconfined when no longer vital. They will be payable only when it will be used by the client. ‘On Demand’ includes a business model in which processing assets are made accessible to the client on an "as required" premise. Instead of at the same time, on-request processing permits cloud facilitating organizations to give their customers access. This key issue needs load balancing techniques. An analysis of load balancing techniques has been given in chapter 2.
- Scalability: It is an amenity to afford the impression that infinite resources will be accessible to fulfill the queries of clients. Scalability with regards to cloud computing can be characterized as the capacity to deal with developing or lessening demands to fulfill business needs in a competent manner. Basically, scalability is an arranged degree of limit that can develop or shrivel as required.
- Multitenant environment: Resource stipulated for many clients with saving significant cost.

### 1.1.2 LOAD BALANCING CHALLENGES

Load balancing is required within the cloud-based interactions for the following general purposes:

- Automated System service
- The center component identified with cloud computing is flexibility; administrations might be doled out or conveyed consequently. How then we would almost certainly use or release the resources of the cloud, simply holding the indistinguishable efficiency as conventional frameworks and using perfect resources. Proposed solution has been given in chapter 4.
- A number of algorithms are formulated to be actually beneficial for an intranet in which communication waiting time are minor. Nonetheless, it is a test to layout a load balancing algorithm which can easily function with regards to divided nodes. This is because alternative aspects will have to be accepted into consideration like the network links speed amount most of the nodes, the length between the consumer and the task filtering nodes, and the ranges among the nodes concerned with delivering the service.
- Virtual Machines Migration allows that, the whole machine can be seen like a file or group of files, to empty a physical machine, or if it is over loaded, it is achievable to swap a virtual machine inside physical machines. The most important goal is to disperse the load in a data center or group of data centers
- In this PhD research work, load balancing raises to website traffic, CPU load, network load and server capacity. Each cloud-based framework incorporates its own complex design and thus, a different set of load balancing algorithms. This makes it difficult to build up a standard outline for load balancing administration. Consequently, load balancing improvement approaches would be motivated from various viewpoints but with the common essence to allow for uniformity of distribution among every single accessible asset.
- In this thesis cloud load balancing focuses on following challenges:
- Content delivery network traffic management: - The thesis, elaborates the deliberation of a content delivery, to create calculations for content circulation. This is a main topic detailed in Chapter 2.

- Optimal Scheduling algorithm for load balancing: - Apply the size and heading of the heap between interconnected undertakings for advancing the heap adjusting conduct. This is a main topic of Chapter 5 of this thesis.
- Cloud Security: At the point when business basic data are moved into the cloud, it is justifiable to worry about its security. Losing information from the cloud, either by inadvertent cancellation, a demonstration of nature cuts down a cloud specialist organization, could be shocking for an undertaking business

**1.2 RELATED WORK**

Load balancing algorithms have been a topic of large research. One open issue is how application structures and related interrelationship can be verified.

As of late, a lot of research has been centered on the load balancing calculations. Prime precedents incorporate static planning, dynamic also, half and half calculations. There have been, in any case, few investigations on burden adjusting calculations that emphasize on their subordinate examples.

- Work process planning is a significant apparatus for applications utilized in e-business and ‘e’science .Handling of information in such applications requires a complex technique, in this way stressing the significance of the work process planning calculations. Distributed computing stages can encourage use of e-science and e-business applications. Thinking about the information and the serious nature of these applications, for example, climate gauging and web based booking, complex information handling is required.

Furthermore, to improve unwavering quality and execution of the e-science applications and in accordance with the cloud framework engineering plans, underlined the improvement of a successful work process planning calculation. Thinking about work process applications, consummation of one errand is required for the execution of interconnected assignments and in this way for the execution of interconnected employments. Consequently, creating keen calculations, which are able to do perceiving the conduct of the interconnected errands and occupations is essential. Such keen computational models would then be able to address the framework imperatives and client necessities, empowering better execution

Other work has been centered on burden adjusting systems and their application to cloud computing. One of these methods is improving information recovery, which majorly affects load adjusting. The information replication procedure, nonetheless, as a significant information recovery technique, has not been concentrated profoundly in the field of cloud

computing. Unpredictability in implanting replication philosophy in cloud-based frameworks might be the explanation behind the absence of consideration regarding this subject [17]. Such an absence of consideration stresses the requirement for building up a brilliant unique replication approach, equipped for improving burden adjusting through replication. During resource provisioning, the cloud organizes requirements to balance the continuous load. Load balancing in the cloud permits a contribution foundation where substantial pools of framework going together to give administrations with Quality of Services (QoS).

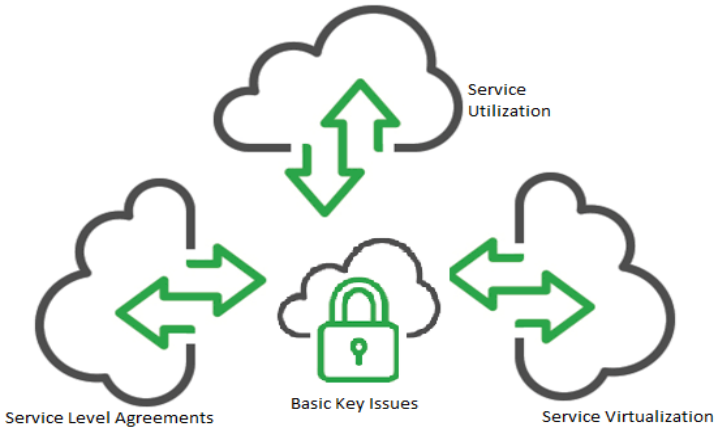



FIGURE 1.2 THREE BASIC KEY ISSUES AS CONTRIBUTION

While focusing on load balancing research, the following key issues need to be considered [13, 14, 15, 16, 19]:

- Service Utilization: cloud computing is a coordinated arrangement of different equipment, programming and intelligent connections cooperating to convey information to the end-client.
- Asset usage inside a cloud – and how the user can adjust key cloud administrations with association's objectives.
- Service Virtualization: By offloading equipment prerequisites and utility costs, it can quickly change an organization's framework and enhance its effectiveness independent from anyone else. Virtualization in cloud computing enables to run numerous applications and working frameworks on a similar server, in this way accommodating proficient asset use and decreasing expenses. Virtualization allows for cost-effectiveness and asset-effectiveness. The following advantages can be listed:
    - Expulsion of exceptional equipment and utility prerequisites;
    - Successful administration of assets;
    - Expanded worker efficiency because of better availability;

- Diminished danger of information misfortune, as information is gone down over numerous capacity areas;
- Advantages for Data Canters
- Expansion of server abilities, accordingly lessening and keep operation costs;
- Little impression because of lower equipment, vitality and labor prerequisites
- Service Level Agreements : Is an agreement between a specialist organization and its inner or outside clients that reports what benefits the supplier will outfit and characterize the execution guidelines the supplier is committed to meet.
- Service Level Agreement (SLA) describes the agreement on non-functional requirements between provider and customer.
- SLA consists of Service Level Objectives (SLOs) that are evaluated according to measurable Key Performance Indicators (KPIs).
- Automatic SLA protection enables further increase of the system utilization and system profit

A cloud stage would offer asset use as on request, which establishes the framework for application to scale amid runtime. Existing methodologies expect designers to rework their applications to use the on-request asset usage with improved arrangement. The single server is not any more ready to adjust developing number of administration asks for, a more costly server with better execution can be utilized [7, 8, 13].

### 1.3 PROBLEM STATEMENT

This research proposes to develop green cloud computing feature in SaaS using load balancing approach with optimization to traffic request at initial stage and then segregate. It improves the QoS (Quality of Services) issues like name space virtualization, optimization algorithms for resource utilization, service level agreements and security. Cloud computing devours huge amount of power and energy. It results in deficiency of energy, which also makes effect on global solution.

To prove cloud technology should be Green it desires to observe two ways of solution:-

- 1) Usage of energy efficiency with minimal cost of operation for resource management and scheduling.
- 2) Provide solution with optimization at the time of cloud requests and reply for resource utilization.

In this thesis, Author offer an optimization method using virtualization solution. With the help of virtualization, the utilization and energy of resources can be maximized, which results in runtime demand fulfillment, as well as scaling performance of server based tasks. Scale performance is used to calculate the service utilization with the objective to prove the green sustainability. This research contributes to the detection of the server failure and the repartitioning of the client traffic like severance. To qualify as be 'GREEN' computing, the load balancing should be aware of how it selects, approximates, evaluates, and calculates the performance of the system, the communication between the nodes, and the nature of work to be transferred and tested.

Load balancing with 'GREEN' gauge additionally needs to upgrade the load and to manage security challenges with advance in collaboration of the resource use and occupation reaction time. Load forecasting plays an imperative role for load balancing. Surveying of the forecasting methods provides the scheme about the client's request to providers. Load anticipating assumes a basic part for stack adjusting. Looking over estimating strategies gives the plan about the customer's demand to suppliers.

This thesis further proposes load balancing based on spread over optimization and virtualization techniques with security objectives for cloud providers and subscribers. Finding a green cloud computing load balancing algorithm is a challenge. A main problem lies in how to split the computational task across many different nodes in the cluster, so that the whole cluster system deliver augmented, energy efficiency, and performance for wired and wireless network operation centers.

## 1.4 OVERVIEW OF CONTRIBUTION

Figure 1.3 shows the scope of this PhD thesis, which relates to advancing the state of the art in green load balancing for cloud computing.

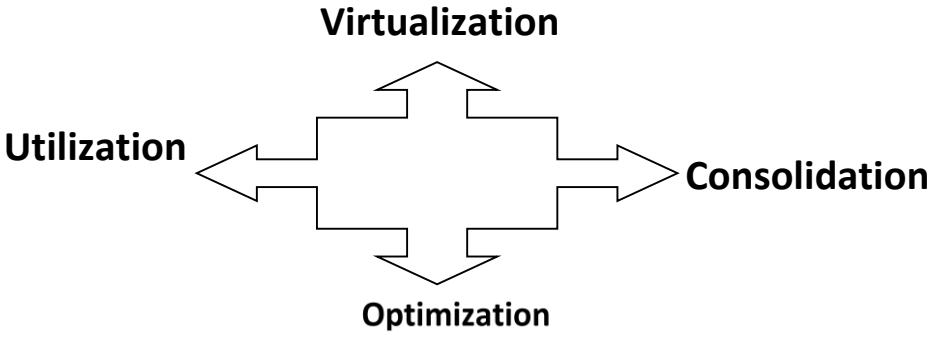



FIGURE 1.3 SUPPORT OF THIS PHD THESIS

Optimization applied with Virtualization, Consolidation, Utilization and Multitenancy approach has been carried out provided to confirm green concepts in cloud load balancing [1,3].

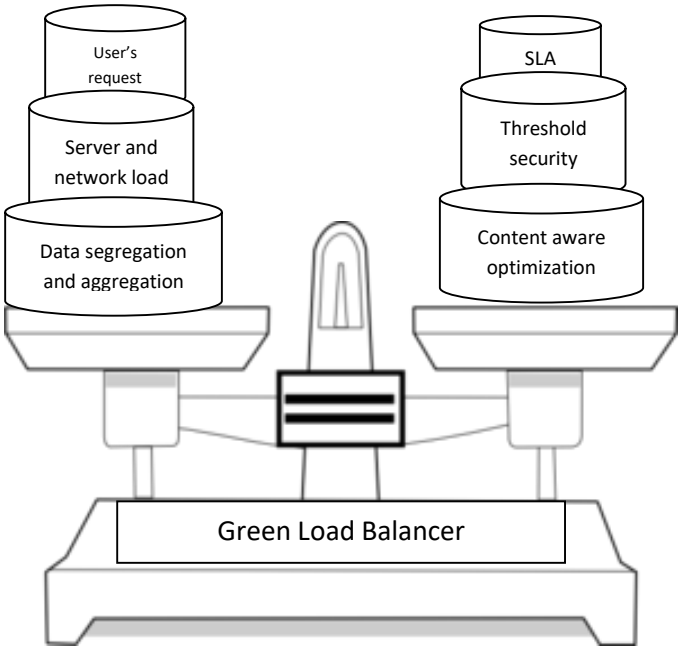



FIGURE 1.4 THESIS PURPOSE

Figure 1.4 shows the objectives of the thesis. The cloud computing needs issues such as data aggregation and segregation. Subsequently, as appeared in the balancer information conglomeration and isolation connected with content mindful enhancement enhances perception of 'Green'. At the point when information storing required server and system speed execution it additionally accompanies edge security approach likewise a piece of Quality of Service issue.

To prove the 'Green' concepts in cloud computing, the following research objectives have been defined:-

- Improve the performance substantially
- Maintain the system steadiness
- Improve response time

Load balancing distributes the network traffic for handling million requests by using scheduling. Load balancing is a process where the load is divided among various nodes and each node is compatible to handle required amount of load either dynamically or statically. Typically, a specified amount of load is loaded and fixed amount of traffic is to be divided

equally among the various layers in the servers [9, 12, 14]. But during some time the load may get increased or decreased with an imbalance may get created. Static load balancing the load sometimes may get unevenly cloud and may raise some issues.

Load balancing makes sure that all the processor in the system or every node in the network does approximately the identical amount of occupation at any instances of time.

Many algorithms for load balancing have been proposed up till now by various authors. Some of the algorithms for load balancing have been overviewed in this thesis. The performance of various algorithms has been deliberated, equated in Chapter 5 to accomplish the aim mentioned in introduction.

The cloud has emerged as a main optimality for service vendors ordinarily due to their support for virtualization and service-oriented methodology. Esoteric cloud entirely can be offered *'As A Service'.* Accordingly scheduling tasks on services becomes even more challenging as inside cloud environments each member uses its peculiar policies and is not gratified to adhere outside rules [1-5].

The load balancing algorithm ensures that every computing resource will be used by the cloud efficiently and in the end would improves the resource utilization. Content aware dynamic scheduling algorithm emphasizes load balancing and load sharing in task distribution. It also provides a monitoring and control system that understands the distribution of requests, maintains technical performance metrics. It also notifies the information of alters to administrators quickly, regarding the problem occurrence.

While balancing the load, which is content aware workflow scheduling is necessary belongings lengthwise with virtualization. A main problem lies in how to split the computational task across many different nodes in the cluster. In this thesis this proposal is present in Chapter 2 and 3 respectively.

Cloud computing can be used for dynamic load balancing. With Software as a Service (SaaS) on the server is essential to load for many NOCs [Network Operation Centers] in real time experiment. As shown in Figure 1.3 the load balancer has two major sides of control, i.e. client to server communication, and server too many server [Server Farm] communication. Client requests will be controlled by the server load balancer. Each cluster of servers has control of the cloud controller load balancer.

After data aggregation the load will be distributed among a group of servers. Figure 1.5 shows a content aware secure load balancing model. On another side of the cloud i.e. the cloud load balancer will mainly work for data segregation, because each

web site server is also connected to the server farm. The redundancy segregated balancer will fulfill the execution of content aware request.

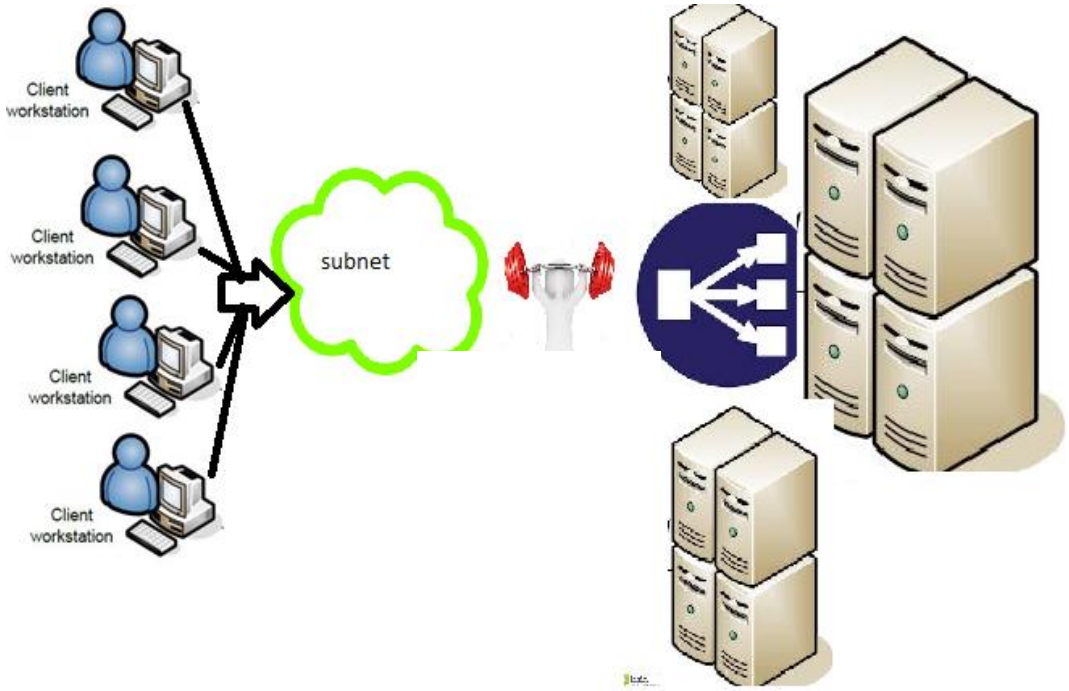



FIGURE 1.5 PROPOSED ARRANGEMENTS FOR LOAD BALANCER

There are various load balancing techniques and algorithms available [7-13]. One basic issue is the load modifying with optimized establishments. There are very less number of results are available related to the virtualization of the substance careful movement, use, storing, name space and portion [14, 17,19]. As shown in Figure 1.5, this research will bring about with the help of a Cloudsim simulator and using mathematical and analytical modeling a novel approach to virtualized load balancing.

Cloud computing is partitioned into three basic parts; provider, user (subscriber) and patron. The provider provides services to the end user with disturbed physical resources using virtualization technology and data centers. This is useful for the patrons from investment on physical hardware and which will take it on rent from providers. Now a days cloud services designated as 'X', as a service (XaaS) [2,5]. 'X' can be castoff as access (AaaS), data( DaaS), infrastructure (IaaS), software (SaaS), platform (PaaS), storage (Staas) and many more[18-26].

## 1.5 METHODOLOGY

The thesis adopts a public cloud model, which has various hubs with circulated registered assets in a wide range of geographic areas. Along these lines, this model

partitions public cloud into a few cloud allotments. The cloud has a fundamental controller that picks the reasonable allotments for arriving employments while the balancer for each cloud parcel picks the best load adjusting procedure. The load balancing system depends on the cloud apportioning idea. In the wake of making the cloud segments, the heap adjusting then begins.

### 1.5.1 HYPOTHESIS:

This research work proposes novel entire system for load balancing, which was helpful to scale assorted kinds of utilizations. This procedure based approach computerizes the arrangement and scaling of uses in the cloud utilized case study has been developed, to exhibit the procedure and plausibility approach.

According to the hypothesis, two main crucial points will be considered in details as to limit servers' resource utilization, we recognized two principle levels of arrangements in

- The server level—limiting the force utilization of a server and
- The server farm level—improving the force utilization of a pool of servers [50].

For the primary level, explicit strategies and methods have been proposed for lessening vitality utilization at the compiler layer, the operational layer, and the application layer [21,22,27, 29]. These procedures include: controlling off bits of the chips for usage of power in load balancer in SSL Secure Socket Layer load balancer, frustrating CPU clock speeds.

It happens that if the record sizes are very huge, the system will support of energy utilization; hence it will be greener to run the application locally than in the clouds. Moreover, numerous work concentrated on simply specific segment of Cloud figuring while at the same time dismissing impact of other, which may not bring about general vitality proficiency.

For instance, VM combination may decrease the number of dynamic servers however it will put unnecessary load on a couple of servers where warm dispersion can turn into a noteworthy issue. Some different works just center on redistribution of workload to help vitality effective cooling without considering the impact of virtualization. Furthermore, Cloud suppliers, being benefit arranged, are searching for arrangements which can diminish the power utilization and accordingly, carbon emanation without harming their market. In this manner, we give a bound together answer for empower Green Cloud processing. We propose a Green Cloud system, which

considers these objectives of supplier while checking the vitality utilization of the clouds. The abnormal state perspective of the green Cloud computing. The objective of this engineering is to make the cloud green from both client and providers point of view.

The following methodologies are identified for load balancing [6, 7, 9,13,14,15]:

- Content aware traffic: In cloud computing content aware based routing and content aware based security are two efficient event delivery issue in load balancing. As aware in a cloud computing communication is client server type. So issues related with client server with content based are an important paradigm for asynchronous communication between entities in a cloud network [7]. While storing data on cloud comprises content aware request applied with optimized way improves performance of system which is more focused in Chapter No 2.
- SLA based load balancing for video surveillance solution using virtualized instance i.e. basically user's request.
- SLA based instances provide enhanced service offerings that are [AD2T2] Automated, Dynamic, Dependable, Transparent, and Tractable.
- More **A**utomated and thus cost-efficient (automated service management procedures).
- More **D**ynamic (reduced preparation and setup time).
- More **D**ependable (explicitly specified SLAs and the confidence to meet them).
- More **T**ransparent (clearer understanding the cost drivers and non-functional properties of their offering).
- More **T**ractable (simplified adjustment or provisioning, possibly relying on third party providers).
- Secure quorum based solution for Name space virtualization.
- Threshold Group Oriented Multi signature cloud.
- The imagined technique will mechanize the organization and scaling of the applications in the cloud. The SaaS is additionally dependable and deals with secured figuring plans, which has combined properties of threshold bunch situated mark plans and multi-signature conspire. Thus to completely guarantee the information security and spare cloud client's control resources, it is of basic importance to team up and contribute similarly to deliver a genuine multi signature.

The results of the research includes:

- Test cases for an implemented application based content aware on the parameters scalability, User response time, data segregation.
- New content aware dynamic load balance techniques to make the application more efficient. Cloud Computing is cheaper for small and medium sized enterprises than large enterprises. Elasticity is another factor for enterprises to adapt Cloud Computing as they can use their resources dynamically.
- New load balancing technique, like computing framework permits endeavors to accomplish increasingly effective utilization of their IT equipment and programming ventures. This is accomplished by separating the physical boundary intrinsic in disconnected frameworks, robotizing the administration of the gathering of the frameworks as a solitary element. Cloud computing can likewise be depicted as at last virtualized framework and a characteristic advancement for server farms.

### 1.5.2 OBJECTIVES:

The objectives of the PhD research can be summarized as follows:

- Amend and recommend the strategy of load balancing efficient scalability, throughput, latency and utilization. Scale the performance of the server based tasks to calculate the service utilization issue.
- Aggregate HTTP requirements and then distributes traffic [segregation] with content aware dynamic requests.
- Define classification methods of load balancing for name space virtualization.
- Reply to the client with checking the availability in the server farm where the actual name space gets assigned.
- Recommend threshold group oriented signature for cloud security.

Resource stipulation by allowing only traffic that is destined for the desired services. Overall the main objective is to design, develop and test a cloud using the high performance computers and hardware along with sustainable and fully equipped software. Moreover, the aim is to simulate a balancing load for SaaS. The cloud has a virtualized environment for running various kinds of applications. It is a conceived that decision of resource allocation is taken by the cloud environment itself. It is also conceived for applying optimized virtual method to prove Green cloud that the proposed design of new secured scheme i.e. threshold group oriented multsignature scheme.

### 1.5.3 CONTRIBUTIONS

Figure 1.6 shows the contribution in the development of a cloud based load balancing scheme using a strategy based approach. This strategy-based approach automates the deployment and scaling of the applications in the cloud. Just-in-time scalability is achieved without binding to specific cloud infrastructure. A real case has been used to demonstrate the process and feasibility of this strategy-based approach.


FIGURE 1.6 PILLARS OF LOAD BALANCING USED AS AUTHORS' CONTRIBUTION [29-39].

Cloud platforms offer resource utilization as on demand service, which lays the foundation for applications to scale during runtime. Existing approaches require developers to rewrite their applications to leverage the on-demand resource utilization, thus bind applications to specific cloud infrastructure [5-10]. So single server is no longer able to balance growing number of service requests, a more expensive server with better performance can be used instead.

The research is based on theory-assisted design and application to practical situations. Powerful network simulators like Cloudsim, jmeter and opnet are available to design and analyses the converged system according to scenario (application, routing, and analysis). Opnet [optimized Network Engineering Tool], which can simulate the capabilities and characteristics of any information network.

So testing the preliminary level of load balancing preference is given to jmeter: open source software, a 100 % pure Java application designed to load test functional behavior and measure performance.

1.6 THESIS OUTLINE

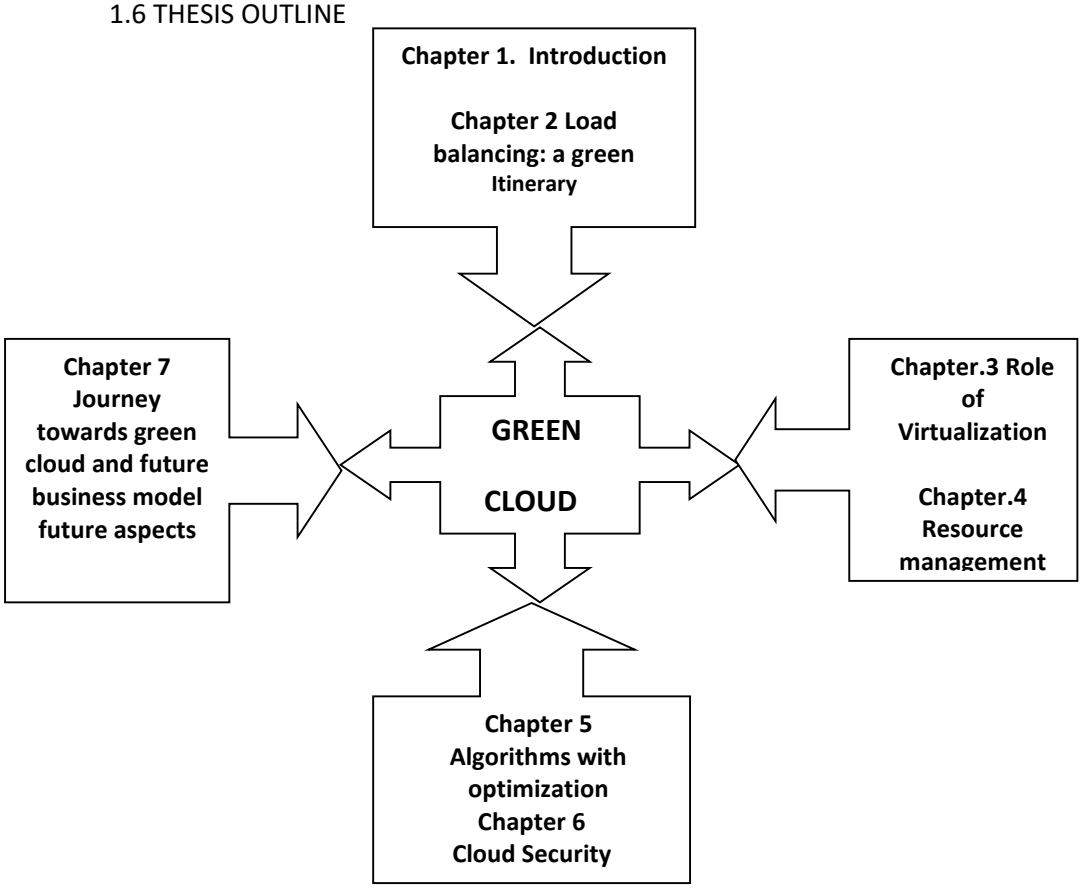



FIGURE 1.7 OUTLINE OF THESIS

The following section present an outline of the thesis with a brief narrative of the individual chapters as shown in Figure 1-5 with individual chapter contributions.

- **Chapter 2- Load Balancing:** proposes forecast algorithm for load balancing. The challenges in load balancing for cloud computing environment have been outlined. Cloud provides a dynamically scalable, abstracted computing and storage infrastructure that is typically based on a virtualized, fault tolerant, parallel computing architecture. Cloud computing refers to computation, software, data access, and storage services that do not require any end-user knowledge of the physical location and configuration of the system that delivers the services. Chapter performed a comparative analysis of load balancing algorithms. The metrics such as throughput, VM ratio, utilization needs to prove how load has been balanced in critics.
- **Chapter 3 Role of Virtualization :** Reflects the idea of virtualization, in computing, refers to the act of creating a virtual (rather than actual) version of something, including but not limited to a virtual computer hardware platform, operating system (OS), storage device, or computer network resources. The chapter gives a detailed description about the development and usage of Virtual Box conventions system.

- **Chapter 4 Resource management:** Cloud computing is another registering model utilizing which applications, information and IT administrations are given over the Web. Distributed computing shares information and offers administrations straightforwardly among its clients. With the expansion in number of clients of cloud the errands to be booked increments. The execution of cloud relies upon the errand booking calculations utilized as a part of the planning segments or handling segments. Planning parallel applications displayed by Directed Acyclic Graphs onto a system of heterogeneous PCs is an NP-complete issue. Various calculations have been projected in the past to take care of the undertaking booking issue for heterogeneous system of PCs, however, none of these calculations can be stretched out to cloud computing frameworks, which are additionally heterogeneous figuring frameworks. Since cloud computing frameworks have a high level of eccentrics as for asset accessibility and system data transmission, undertaking reservation designs for cloud computing frameworks should join the inertness caused by capricious asset accessibility. The present contemplate includes studying the distinctive
  - Undertaking planning calculations created for cloud condition with different stages like
  - Resource requirement, resource monitoring, resource scheduling etc.. Offices ought to build up an organized way to deal with cloud-based administrations as a fundamental part of their ICT technique and guide.
- **Chapter 5 Algorithms for Green Cloud Computing:** Cloud computing is made up by collecting two terms in the field of innovation. To start with term is Cloud and the second term is figuring. Cloud is a pool of heterogeneous assets. It is a work of tremendous foundation and has no significance with its name, alludes to both the applications conveyed to end clients as benefits over the Internet and the equipment and framework programming in datacenters that is in charge of giving those administrations. Keeping in mind the end goal to make proficient utilization of these assets and guarantee their accessibility to the end clients "Registering" is done in light of specific criteria indicated in SLA. Framework in the Cloud is made accessible to the client's On-Demand premise in pay-as-you-say-way. Calculation in cloud is finished with the plan to accomplish most extreme asset usage with higher accessibility at limited cost.
- **Chapter 6 Cloud Security**: Because of this adaptability everybody is exchanging information to cloud. To store information on cloud client needs to send their

information to the outsider who will oversee and store information. So it is critical for the organization to secure that information. Information is said to be secured if privacy, accessibility, respectability is available. To secure information we have diverse calculations.

- **Chapter 7 Conclusion and future aspects:** The principle preferred standpoint of Cloud figuring is that it takes out the need for client to be in same area where equipment programming and storage room is physically displayed. Cloud makes it conceivable to store and access your information from any place whenever without agonizing over support of equipment programming and storage room. All these administrations are given to client requiring little to no effort. Client needs to pay as indicated by storage room he/she is utilizing. Because of this adaptability everybody is exchanging their information on cloud. With conclusion author also presents business model for GREEN growth.

## 1.7 SUMMARY

Cloud computing is an ongoing landing to the universe of IT framework. The idea enables organizations to amplify use of their possibilities and thus help their execution. One of the principle advantages of Cloud Computing is the critical increment in productivity of executing marketable strategies. Moreover, Cloud Computing furnishes huge scale applications with amazing registering power crosswise over worldwide areas. However the cloud clients can share their information effectively by utilizing replication strategies.

Cloud computing structure has been created dependent on a multi-tenure idea. Along these lines, accessibility and effectiveness of the assets are vital factors in the Cloud engineering. As the quantities of clients are expanding quickly, the heap will significantly affect execution and activity of the cloud frameworks. In like manner, advanced load adjusting calculations that can deal with the Cloud stack in a period and cost-effective way are required.

Considerable research has been committed to upgrading load adjusting in cloud computing. This improvement is exhibited through a fair system of associating assets. The objective of this system is to limit the holdup time and augment usage of the throughput. This postulation gives a lot of arrangements which moderate the issue of load balancing in the cloud. The thesis examines a novel class of heuristic planning calculations that

enhances the target of enhancing information accessibility to enhance the heap adjusting between the cloud sites.

In synopsis, this examination development entangles the structure of streamlined load adjusting calculations that think about the extent and bearing of the stack in work process applications. Moreover, by architecting the expectant replication calculation, it limits the quantities of the copies and upgrades the compelling system use in Cloud-based frameworks.

**REFERENCES**


[1] http://www.auerbach-publications.com/dynamic_data/2696_1572_Held_load_balancing.pdf "Network Load balancing" by Jonatahan Held

[2] Trieu C. Chieu, Ajay Mohindra, Alexei A. Karve and Alla Segal 2009 IEEE International Conference on e-Business Engineering2009 IEEE International Conference on e-Business Engineering2009 IEEE International Conference on e-Business Engineering Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment ,IBM T. J. Watson Research Center 19 Skyline Drive Hawthorne, NY, USA

[3] http://www.tcpipguide.com/free/t_DNSNameServerLoadBalancing.htm

[4] http://blog.ioshints.info/2006/11/perfect-load-balancing-with-mpls.html

[5] http://technet.microsoft.com/en-us/library/ff755052.aspx

[6] "Design and implementation of an MPLS based load balancing architecture for Web switching", Radu Dragos, Sanda Dragos & Martin Collier, School of Electronic Engineering – DCU, Switching and Systems Laboratory,Dublin City University, Glasnevin, Dublin 9, Ireland

[7] A. Riabov, 2. Liu, J. Woif, P. Yu, L. Zhang, "Clustering Algorithms for content-based publication - subscription systems," In KDCS 2002.

[8] http://www.radware.com/Resources/link_load_balancing.aspx & http://kb.linuxvirtualserver.org/wiki/Layer-4_load_balancing

[9] http://www.itedge.net/published/Microsoft%20SQL%2Database%20Clustering%20and%20Load%20Balancin B.pdf

[10] http://old.iptel.org/voipsecurity/doc/14%20-%20Kambourakis%2 Towards%20Effective%20SIP%20load%20Balancing.pdf

[11] K. Xiong and H. Perros, "Service Performance and Analysis in Cloud Computing," in Proceedings of the 2009 Congress on Services-I-Volume 00, pp. 693–700, IEEE Computer Society, 2009.

[10] R. Gonzalez and M. Horowitz, "Energy dissipation in general purpose microprocessors," IEEE Journal of Solid-State Circuits, vol. 31, no. 9, pp. 1277–1284, 1996.

[11] D. Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: Eliminating Server Idle Power," in Proceeding of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '09, pp. 205–216, 2009.

[12] S. Pelley, D. Meisner, P. Zandevakili, T. F. Wenisch, and J. Underwood, "Power Routing: Dynamic Power Provisioning in the Data Center," in Proceeding of the 15th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '10, pp. 231–242, 2010.

[13] PassMark, “CPU Benchmarks.” http://www.cpubenchmark.net.

[14] R. Nathuji, C. Isci, and E. Gorbatov, “Exploiting Platform Heterogeneity for Power Efficient Data Centers,” in Proceedings of ICAC’07: Fourth International Conference on Autonomic Computing, 2007.

[15] S. Ghiasi, T. Keller, and F. Rawson, “Scheduling for Heterogeneous Processors in Server Systems,” inProceedings of the 2nd Conference on Computing Frontiers, pp. 199–210, 2005.

[16] L. A. Barroso, “The Price of Performance,” ACM Queue, vol. 3, no. 7, pp. 48–53, 2005.

[17] M. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel, “Amazon S3 for Science Grids: A Viable Solution?,” in Proceedings of the 2008 International Workshop on Data-aware Distributed Computing, pp. 55–64, 2008.

[18] K. Xiong and H. Perros, “Service Performance and Analysis in Cloud Computing,” in Proceedings of the 2009 Congress on Services-I-Volume 00, pp. 693–700, IEEE Computer Society, 2009.

[19] R. Gonzalez and M. Horowitz, “Energy dissipation in general purpose microprocessors,” IEEE Journal of Solid-State Circuits, vol. 31, no. 9, pp. 1277–1284, 1996. “

[20] D. Kotz and K. Essien. “Analysis of a campus-wide wireless network. In ACM MobiCom, pages 107{118, 2002.

[21] www.rightscale.com

[22] N.A. Fikouraas,A.Udugama,C.Gorg in proceeding “ Experiemnta evaluation of load balancing for mobile internet real time communication” Eoropean Commision with IST project Noaman :

[23] “At a glance” 2011 Cisco Systems Inc Cisco Systems third party trade mark

[24] Krzysztof Zajda “Evaluation of possible applications of dynamic routing protocol for load balancing in computer networks. IEEE proceedings HET-NET’s 2010, ISBN 978-83-9260544-4-7,pp 285-296.

[25] Hui Chen,Chunjie Zhou, Yuan quing Qin , “Petrinet modelling of reconfigurable protocol stack for cloud commuting control systems” 2nd IEEE International Conference on cloud computing Technology and Science.

[26] Chuney Gong,Jie Liu, Qiang Zhang, Zhenghu Gong “ the characteristics of Cloud Computing”. IEEE 1530-2016/10 39th International Conference on Parallel Processing Workshop

[27] M. Armbrust, A.Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia., "Above the Clouds: A Berkeley View of Cloud computing," University of California at Berkley, USA, Technical Report , Technical Report No. UCB/EECS-2009-28, 2009

[28] “Hyrax Cloud computing on mobile devices using Map Reduce” thesis by Eugene E. Mariela, School of computer science, Pittsburgh, PA 15213.

[29] Araddhana A. Deshmukh, Dr. Albena D. Mihovska, Dr. Ramjee Prasad, “Zoneminder as ‘Software as a Service’ and Load Balancing of Video Surveillance Requests”, Networking and Electronic Commerce Research Conference 2012 (NAEC 2012), http://www.atsma.org/Conferences/NAEC2012,Italy,Riva del Garda Italy, October 11-14 2012

[30] Araddhana A. Deshmukh, Dr. Albena D. Mihovska, Dr. Ramjee Prasad “Cloud computing security schemes:- TGOS [Threshold Group-Oriented signature ] and TMS [Threshold multisignature”, 2nd World Congress on Information and Communication Technology http://www.mirlabs.org/wict12/#cfp, WICT 2012, Trivandrum India, October 30- November 2, 2012 [ The paper also shortlisted by www.mirlabs.net/WICT 2012 for journal http://www.mirlabs.net/wict12/#journal .]

[31] Araddhana A. Deshmukh, Dr. Albena D. Mihovska, Dr. Ramjee Prasad, ”Optimized Resource Stipulation for Video on Demand Via Zoneminder”, Global Wireless Summit 2014 in Aalborg from May 11-14, 2014 Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2014 4th International Conference on, Aalborg, Denmark, May 11-14, 2014 [Based on the GWS2014 paper

we have been invited to submit a journal paper in the Special issue on "Advances in Control for the provision of Smart Services", which will be published on the novel CONASENSE journal ]

[32] Araddhana A. Deshmukh, Pankaj Chaudhary , ,Dr. Albena Mihovska, Dr. Ramjee Prasad,' NUNI ( New User New Item) Problem for SRSs Using Content aware multimedia based Approach' 3rd International Conference on Innovations in Computer Science & Engineering Hosted by: Guru Nanak Institutions In Association with CSI Hyderabad Chapter, Division 5, Education and Research, CSI –India Dates: 7th – 8th August 2015, Venue: Guru Nanak Institutions, Hyderabad, India, Springer publication, ://www.springer.com/series/11156

[33] Dipti Madankar , Araddhana A. Deshmukh, Dr. Albena Mihovska , Vaishali Maheskar , ' SQAMPS: Secure Quorum Architecture for Mobile Presence Services with End-to-End Encryption, WCI '15 Proceedings of the Third International Symposium on Women in Computing and Informatics , August 10-13, 2015, Kochi, India, ACM ISBN 978-1- 4-4503-3361-0/15/0

[34] Shubhangi Opale , ,Araddhana A. Deshmukh, Dr. Albena Mihovska, Dr. Ramjee Prasad , "'Privacy-Preserving Friend Matching Protocol approach for Pre-match in Social Networks", WPMC 2015 18th International Symposium on Wireless Personal Multimedia Communications, Global Wireless Summit 2015, Hyderabad , Dec 13-16, 2015

[35] Araddhana A. Deshmukh, Snehal Sawant ,Dr. Albena Mihovska, Dr. Ramjee Prasad ,"'Public Auditing and Data Dynamics in Cloud with Performance Assessment on Third Party Auditor'" WVITAE 2015, 5th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), Global Wireless Summit 2015, Hyderabad , Dec 13-16, 2015

[36] Araddhana A. Deshmukh, Reshma Gade, Dr.Albena Mihovska, Dr. Ramjee Prasad," Inference attack on URL visiting history of the social networking", International Journal of Control Theory and Applications., I J C T A, 9(10), 2016, pp. 1-11 © International Science Press, ISSN: 0974-5572, PP 1111-1118, Dec 2016

[37] Araddhana A. Deshmukh, Dr.Albena Mihovska, Dr. Ramjee Prasad," Cloud Security using COSDACC [ Cluster of Servers and Access Control], IEEE 5G Summit and 30th GISFI Standardization Series Meeting, ,SRM University, Kattankulathur, Chennai. *Also shortlisted for Conasense Journal*, 16-17 November 2017

[38] Araddhana A. Deshmukh, Smita Ashok Yeole, Dr.Albena Mihovska, Dr. Ramjee Prasad, "An Approach: A Novel Protected Data Deduplication Scheme with Dynamic Ownership Management to Secure Cloud Data", International Journal of Pure and Applied Mathematics, Volume 118 No. 24 2018, ISSN: 1314-3395 (on-line version),url: http://www.acadpubl.eu/hub/Special Issue,http://www.acadpubl.eu/hub/

[39 ] Araddhana A. Deshmukh, Dr. Albena Mihovska, Dr. Ramjee Prasad, " Secured Outsourced Integrity Test for Cloud Storage", Fourth International conference on Information and Communication Technology for Sustainable Development ( ICT4SD 2019) with World Publication and Conference Summit PubCon2019 Goa, India, 5-6 July 2019.

# CHAPTER 2 LOAD BALANCING: A GREEN ITINERARY

Chapter 2 explores the main concepts and issues of cloud load balancing. Cloud scalability is related to enabling multiple virtual servers and the ability to support the running of multiple applications instances. Load balancing is an essential component to distribute traffic among these instances, which then ensures maximum throughput in minimum response time.

This chapter proposes optimized load balancing that also caters for making the cloud operations energy efficient. The chapter evaluates the impact of this for a real time application like 'VoD' i.e. Video on Demand, which may involve huge number of clients requesting the same service simultaneously.

This Chapter is organized as follows Section 2.1. is an existing survey i.e. appreciative occurrence Section 2.2 focuses on existing approaches. Section 2.3 outlines all application oriented research for business products.

## 2.1 APPRECIATIVE OCCURRENCE

The web servers of famous sites are regularly founded on appropriated or parallel design, while saving a virtual single interface. This results into little inertness time and less weight on every server. Distinctive sites utilize diverse systems to appropriate load among web servers yet most of the plans focus on just a single factor that is number of solicitations, however none of the plans consider the point that:

- Different sort of solicitations will require distinctive level of preparing endeavors to reply.
- Status record of all the web servers that are related with one space name must be considered.
- Mechanism to deal with a circumstance when one of the servers is not working.

Load balancing can be characterized as a technique for conveying the load on the numerous PCs or a PC group through system connects to accomplish ideal asset usage which augments throughput and limits the general reaction time. It limits the aggregate holding up time of the assets as well as maintains a strategic distance from an excess of over-burden on the assets. In this method the activity is partitioned among servers, with the goal that information can be sent and got immediately.

One of the essential issue in cloud computing is to isolate the load dynamically. Workload of a machine implies to add up to preparing the time it requires for executing every one of the undertakings allotted to the machine. Load balancing is the procedure of enhancing the

execution of the framework by moving of the workload among the processors. The advantages of circulating the workload incorporates a higher asset use proportion which additionally prompts improving the general execution in this manner accomplishing most extreme customer fulfilment. Through load balancing each virtual machine in the cloud framework does the same measure of work that expands the throughput and limits the reaction time. Consequently applying optimization is one of the imperative factors to increase the working execution of the cloud specialist organization.

### 2.1.1 LOAD BALANCING PROSPECTS

There are three basic service models of cloud computing, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Infrastructure as a Service is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. The service provider uses this equipment and is responsible for space, running and maintaining it. The client typically pays on a per-use basis. In this service model, cloud providers offer resources to users/machines that include computers as virtual machines, raw (block) storage, firewalls, load balancer and network devices. The resource management has the following characteristics and components of include:

- Utility computing service and billing model.
- Automation of administrative tasks.
- Dynamic scaling.
- Policy-based services
- Expectation

Providing virtual machines (VMs) to users to satisfy their demand is a challenge in load balancing – because of the need for the proper deployment of the virtual machine onto available hosts. At present, many load-balancing methods are based on the load forecasting model that could predict the resource requirements of the virtual machine [11,12,16,19]. However, the resource requirement of the virtual machine in the cloud is hard to predict because there will be variety of load types numbers of VMs. Moreover, a variety of heterogeneous hardware environments and virtualization technologies make it hard to predict the requirement of the VM based on the workloads.

2.1.2 STEPS FOR PERFORMANCE CHECKING IN LOAD BALANCING

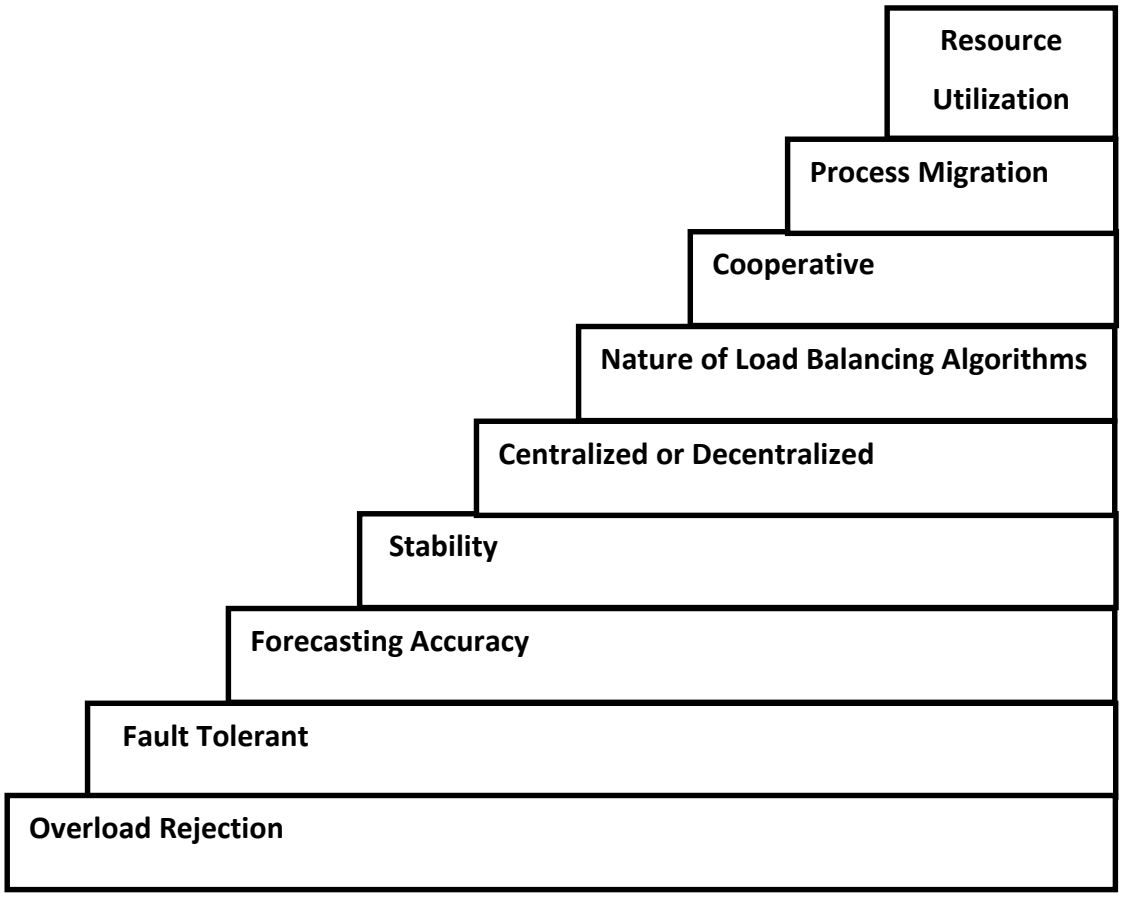



FIGURE 2.1 STEPS FOR PERFORMANCE CHECKING IN LOAD BALANCING

Figure 2.1 shows the situational steps to increase the performance of load balancing:

- **Overload Rejection**

  If load balancing is not possible, additional overload rejection measures are needed. When the overload situation ends then first the overload rejection measures are stopped. After a short guard period load balancing is also closed down [5].

- **Fault Tolerance**

  This parameter shows whether the algorithm is able to tolerate secondary faults or not. It enables the algorithm to continue operating properly in the event of some failure. If the performance of the algorithm decreases, the decrease is proportional to the seriousness of the failure, even a small failure can cause total failure in load balancing [5],[6].

- **Forecasting Accuracy**

  Forecasting is the degree of conformity of calculated results to its actual value that will be generated after execution. The static algorithms provide more accuracy than of dynamic

algorithms as in former most assumptions are made during compile time and in later this is done during execution [6],[7],[8].

- **Stability**

  Stability can be characterized in terms of the delays in the transfer of information between processors and the gains in the load balancing algorithm by obtaining faster performance by a specified amount of time [6],[9],[10].

- **Centralized or Decentralized**

  Centralized schemes store global information at a designated node. All sender or receiver nodes access the designated node to calculate the amount of load-transfers and also to check that tasks are to be sent to or received from. In a distributed load balancing, every node executes balancing separately. The idle nodes can obtain load during runtime from a shared global queue of processes.

- **Nature of Load Balancing Algorithms**

  Static load balancing assigns load to nodes probabilistically or deterministically without consideration of runtime events. It is generally impossible to make predictions of arrival times of loads and processing times required for future loads. On the other hand, in a dynamic load balancing the load distribution is made during run-time based on current processing rates and network condition. A DLB policy can use either local or global information [4],[5],[6].

- **Cooperative**

  This parameter gives that whether processors share information between them in making the process allocation decision other are not during execution. What this parameter defines is the extent of independence that each processor has in concluding that how should it can use its own resources. In the cooperative situation all processors have the accountability to carry out its own portion of the scheduling task, but all processors work together to achieve a goal of better efficiency. In the non-cooperative individual processors act as independent entities and arrive at decisions about the use of their resources without any effect of their decision on the rest of the system.

- **Process Migration**

  Process migration parameter provides when does a system decide to export a process? It decides whether to create it locally or create it on a remote processing element. The algorithm is capable to decide that it should make changes of load distribution during the execution of process or not.

- **Resource Utilization**

Resource utilization include automatic load balancing A distributed system may have unexpected number of processes that demand more processing power. If the algorithm is capable to utilize resources, they can be moved to under loaded processors more efficiently.

## 2.2 EXISTING SYSTEM APPRAISAL

Figure 2.2 shows ranked classification of existing load balancing methods

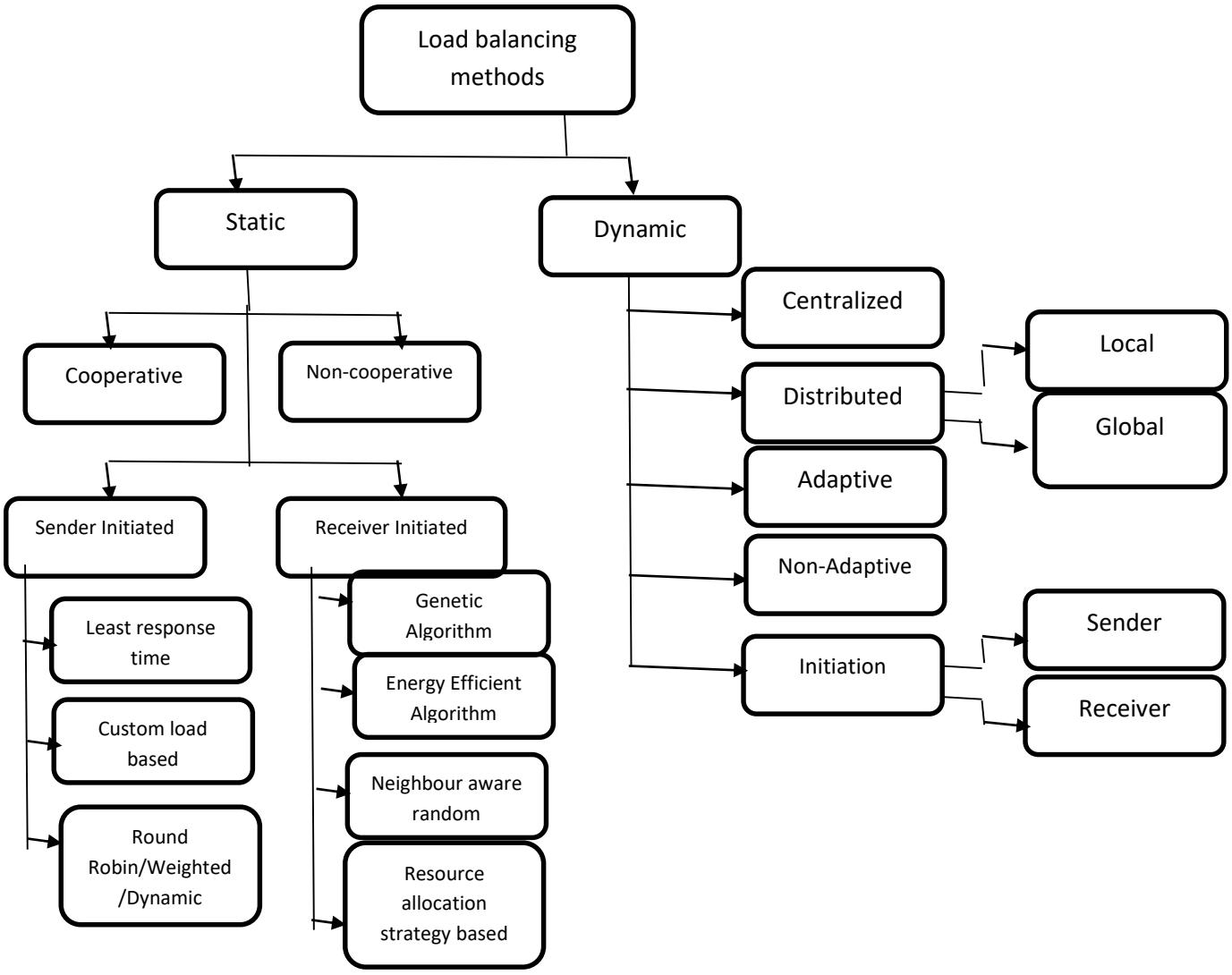



FIGURE 2.2 LOAD BALANCING CLASSIFICATION

Ordinarily cloud vendors execute programmed load balancing instruments into their administration conveyance framework. This methodology enables the quantities of assets to be balanced with changing the interest levels. In an all-around oversaw cloud framework, load balancing capacities is considered an absolute necessity [7]. The objective of the load balancers is twofold: the essential objective is to advance the accessibility and administration provisioning of the Cloud assets, and the optional

objective is to enhance the execution if necessary and relying upon the substances needs [8],[11].

Throughout the years, there have been significant enhancements in load balancing methods. Load balancers have been creating distinctive needs so as to build the productivity and execution of the appropriated and multithreaded frameworks [14]. Three essential load balancing models are utilized in the greater part of the Cloud 'n' services: Server Availability show, IP Traffic Management model and Priority Queue display [16].

The Server Availability demonstrate works dependent on a component that decides the most readily accessible server that can be used. Other than servers, Server Availability model can be connected to any assets

### 2.2.1 STATIC LOAD BALANCING ALOGORITHM

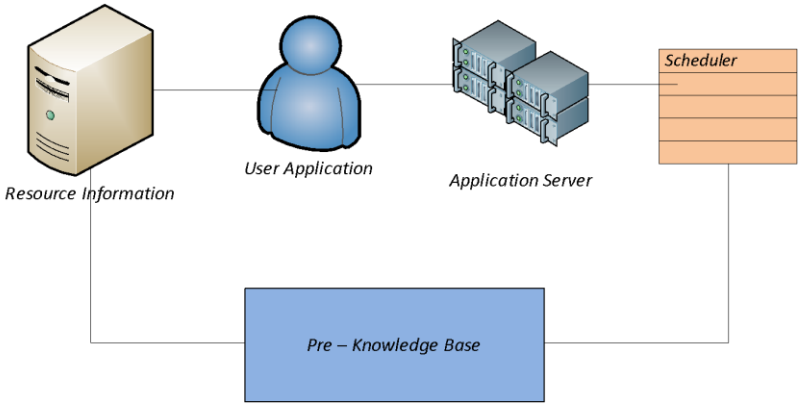



FIGURE 2.3 STATIC LOAD BALANCING

Figure 2.3 shows static load balancing, In this method the performance of the processors is determined at the beginning of execution. Then depending upon their performance the work load is distributed in the start by the master processor.

The slave processors calculate their allocated work and submit their result to the master. A task is always executed on the processor to which it is assigned that is static load balancing methods are non-pre-emptive.

In a static situation, earlier data about hub limit, handling force, memory and execution is required. The insights prerequisites cannot be changed at run-time. Despite the fact that the static condition is a lot simpler for executing the heap adjusting calculations, it is not appropriate for heterogeneous calculation models [19]. A straightforward case of a static calculation is Round-Robin calculation. In this asset booking strategy, the undertaking that starts things out will be served first and

dependent on the time sharing way the asset that is least stacked will be apportioned to finish the errands [20],[21].

The goal of the static load balancing method is to reduce the overall execution time of a concurrent program while minimizing the communication delays with cooperative and non-cooperative approaches. . A general disadvantage of all static schemes is that the final selection of a host for process allocation is made when the process is created and cannot be changed during process execution to make changes in the system load. A few static load balancing techniques are:

- Round robin algorithm - the tasks are passed to processes in a sequential order; when the last process has received a task the schedule continues with the first process (a new round).
- Randomized algorithm: the allocation of tasks to processes is random, to pick slave processors. The slave processors are picked haphazardly following irregular numbers produced in view of a static distribution, the measurement appropriation.
- Simulated annealing or genetic algorithms: mixture allocation procedure including optimization techniques.
- Central Manager Algorithm: Central processor will pick a slave processor to be allocated to work. The picked slave processor is the processor having the minimum load. The central processor can accumulate all slave processors stack data, thereof the picking in view of this calculation are conceivable to be performed.
- Threshold Algorithm: The processes are assigned immediately upon creation to hosts. Hosts for new processes are selected locally without sending remote messages. Each processor keeps a private copy of the system's load. The load of a processor can be characterized by one of the following three levels: Under loaded, medium and overloaded.

Two threshold parameters t_under and t_upper can be used to describe these levels.

Under loaded: load < t_under, Medium: t_under ≤ load ≤ t_upper, Overloaded: load > t_upper.

Even when a good mathematical solution exists, static load balancing still have several flaws:-

- It is very difficult to estimate a-priori [in an accurate way] the execution time of various parts of a program
- Sometimes there are communication delays that vary in an uncontrollable way

- For some problems the number of steps to reach a solution is not known in advance

### 2.2.2 DYNAMIC LOAD BALANCING ALGORITHM

Dynamic load balancing differs from static algorithms because of the work load is distributed among the processors at runtime. The master assigns new processes to the slaves based on the new information collected. Unlike static algorithms, dynamic algorithms allocate processes dynamically when one of the processors becomes under loaded. Instead, they are buffered in the queue on the main host and allocated dynamically upon requests from remote hosts.

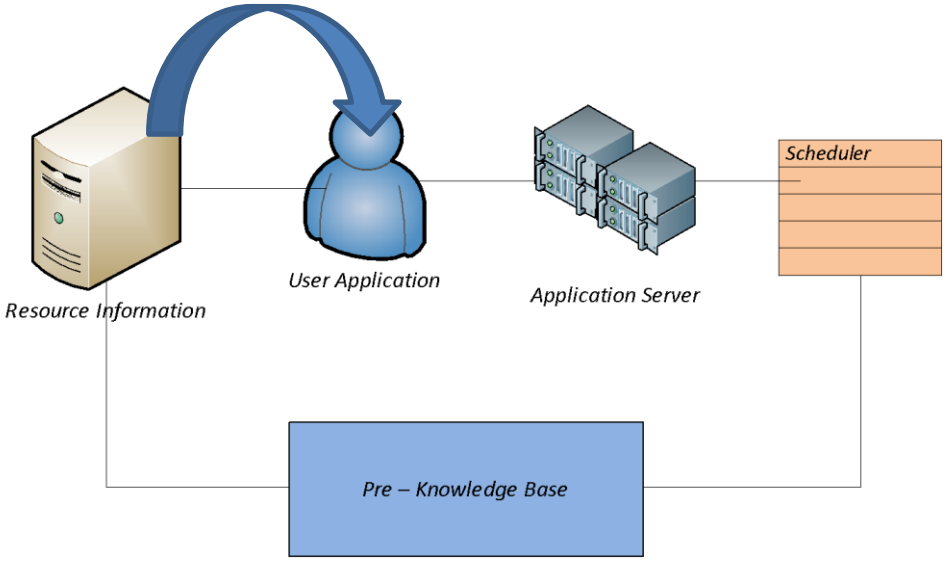



FIGURE 2.4 DYNAMIC LOAD BALANCING

Dynamic load adjusting is a condition in spite of the requirement for earlier, similar to static condition, the calculations work as per the run-time insights [23]. These load adjusting calculations are increasingly adaptable to change and they are highly adaptable to cloud environments.

Load balancing problem has been discussed in traditional distributed systems literature for more than two decades. Various algorithms, strategies and policies have been proposed, implemented and classified. Algorithms can be classified into two categories: **static or dynamic.** Its algorithmic existing system is as shown in below Table 2.1

TABLE 2.1 EXISTING SYSTEM SURVEY

| Sr. No. | Name of algorithm | Description i.e. role of load balancer | Performance Metrics |
|---|---|---|---|
| 1. [11] | Round robin | Processes are divided evenly among all processors | Throughput calculated in real time. But in scalability complications may occur. |
| 2. [12] | Randomized | Process can be handled by node 'n' with probability 'p' | Throughput, performance and storage resource calculated fairly. But fail in fault tolerance not calculated. |
| 3. [13] | Central Manager | load balancing judgment from the on hand information on the system load state. | Clearly defined centralized control but fail in resource utilization |
| 4. [14] | Threshold | processes are assigned immediately upon creation to hosts | Traffic flow has no fixed scheduling |
| 5.[15] | Central Queue | Each new activity arriving at the queue manager is inserted into the queue. | Latency problem may occurred during request distribution. |
| 6.[16] | Local Queue | static allocation of all new processes with process migration initiated by a host when its load falls under threshold limit, is a user-defined parameter of the algorithm | It improves overall performance but reduces idle time of nodes. |
| 7[18] | Least connection | dynamic scheduling algorithms; because it needs to count the number of connections for each server dynamically to estimate its load | Incurs no communication overhead at job arrivals. |
| 8[19] | Weighted Least Connection | The servers with a higher weight value will receive a larger percentage of active connections at any one time | Smallest latency |
| 9.[20] | Load balancing strategy for virtual storage | Uses fair share replication strategy to control access | Enhanced flexibility and robustness. This provides large scale data storage for storage as a service. |
| 10[21] | Central LB policy for virtual machines | Global state info for load balance | Does not consider fault tolerance. |
| 11[22] | Lock free multiprocessing | Runs multiple load balancing processes in one load balancer | Fails in scalability and response time. |
| 12[23] | Scheduling strategy on LB of VM | Uses genetic algorithm with historic and current state of system | It solves the problem of bottleneck but it has high migration cost. |
| 13[24] | Join Ideal Queue | Assigns ideal processors | It uses mainly for reducing system load. |
| 14[25] | Honey bee foraging behaviour | Global load balancing through local server | It does not increases throughput as system size increases |
| 15[26] | Biased random sampling | Using random sampling of system | It does not solves the problem of latency and migration |

| | | | |
|---|---|---|---|
| 16[27] | Active clustering | Job optimization | Performs better with high resources. But also degrades when system diversity increases |
| 17[28] | Ant colony complex network | Uses small world and scale free characteristics | It overcomes heterogeneity and very good scalability. |
| 18[29] | Agent based resource monitoring algorithm. Agents are used to do some task. Agents collect information about VMs. | Domain specific information is collected. Individual packet is checked instead of whole cloud. | SLA issues |
| 19[19] | Genetic algorithm is used. Crossover, mutation, chromosome concepts are used. | Cloudsim is used for experiments. Possible solutions are encoded binary strings. Chromosome is optimal solution. Mutation value of each chromosome is checked. | Simple approach. Only same values of crossover and mutation are taken. |
| 20[20] | Particle swarm optimization. | Instead of migrating whole VM we will migrate extra tasks. VM does not loss information. Cloudsim packages and Jswarm packages are used. | This algorithm is applicable to simple task scheduling optimization. |
| 21[21] | Metadata is used and it is searched using hashing functions. | Multiple copies of single data. It is easy to balance load as multiple copies are present. | This method is applicable to simple file systems. It will be have to work with real world file systems. |
| 22[23] | Optimize balance program. Optimize group program, Optimize hops program. | Many cloud management platforms are available for one who wants to develop his own private cloud e.g., OpenStack, Eucalyptus, Open Nebula. A common design aspect of current platforms regards their black-box-like controlling nature, where cloud administrators have few opportunities to influence how resources are actually managed (e.g., virtual machine placement or virtual link path selection). In this article, they introduce a new concept of cloud management platform where Programmability resource management is made flexible by the addition of programmability to the core of Aurora the platform, with a simplified object-oriented API. | Prototype architecture is complex, admin can introduce his own policy thought programs for different Applications. |
| 23[24] | Multiobjective genetic Algorithm | Live VM Migration technique has been proposed to reduce the downtime for migrated overload VMs. | Another type of migration. Reduces time and cost of VM migration. TBSLB is |

| | | |
|---|---|---|
| Task based system load balancing algorithm. | As VMs migration takes much more times and cost in comparison with tasks migration, this study develops a novel approach to confront with the problem of overload VM and achieving system load balancing, by assigning the arrival task to another similar VM in a cloud environment. In addition, we propose a multi-objective optimization model to migrate these tasks to a new VM host applying multi-objective genetic algorithm (MOGA). | used for task migration. |

Based on above survey following is study:-

Resource allocation provides the asset on a request premise. The fundamental point is to monitor the over-burden hub with the goal that no wastage of assets occurs. The wastage suggests the wastage of CPU speed, memory or data transmission. The whole mapping is done in the following two levels.

The principal level is the mapping of the virtual machine to the host. The virtual machine (VM) dwells on the physical servers known as hosts. The VM is mapped to the host and the procedure relies upon the limit and accessibility. Distribution relies upon the on-request necessities.

The second level is application mapping to the virtual machine. For the execution an application requires some power. The VM gives the power as the applications are executed on the virtual machines. Applications are mapped to the VM and this is subject to the accessibility and the design.

**2.3 PROJECTED SYSTEM ARCHITECTURE**

Our system in Figure 2.5 shows that the proposed scheme for load balancing consist of a VM monitor (VMM), CIS (Cloud Info Services) and shared storage. The VM monitor cluster acts as a resource pool on which virtual machines run. The shared storage is mainly used to store virtual machine disk images. The role of VMM is the infrastructure platform for IaaS cloud.

The CIS is responsible for monitoring the resource usage of cloud pool, collecting historical load information of physical hosts and virtual machines, allocating and relocating virtual machines on appropriate hosts based on load-balancing strategies.

The collector on CIS needs to collect the performance data of each VM and host in VMM. The performance data include CPU utilization, memory usage, network and disk I/O throughput, etc. By online statistical analysis of the performance data, we can obtain the load characteristics of the virtual machine, as well as some cyclical variation. These characteristics of the virtual

machine are used to predict the future load changes, and guide the estimation of resource requirement of virtual machines.

A controller on CIS needs to manage the resource of VMM (IaaS cloud), control the virtual machines and the physical hosts in the VMM resource pool. When a virtual machine starts, the CIS needs to allocate resources in the pool for this virtual machine, that means select a physical host in VMM to run this virtual machine. We can get the load information of virtual machines by load forecasting model. Next, the resource requirement for the physical hosts will be calculated based on these loads by using load-resource model.

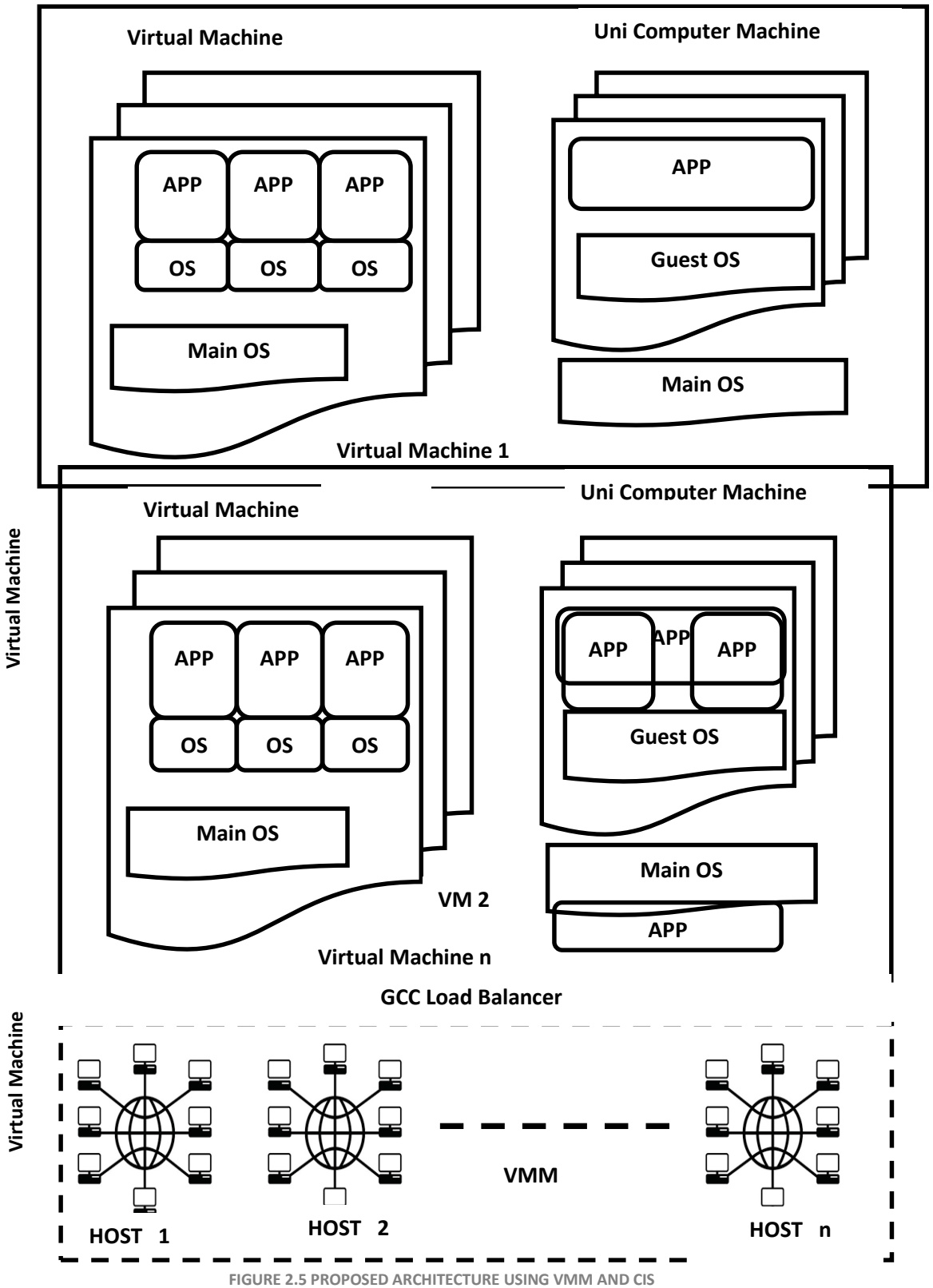



FIGURE 2.5 PROPOSED ARCHITECTURE USING VMM AND CIS

Therefore, we can select an appropriate host to run this VM based on the resource requirement of the virtual machine and the recent load information of physical hosts. The principle for selecting the host is to keep the system load balanced. Furthermore, the controller monitor the resource usage of each physical host, if the resource usage of some host becomes higher than other hosts, some virtual machines

would be relocated to other hosts through VM migration to maintain the entire system load-balancing intact[1, 8].

In cloud computing, IaaS is more important than SaaS and PaaS, which provides VM to users on their demand. Proper deployment of the VM onto available hosts plays crucial role on the load-balancing of the modern data center. At present, many load-balancing methods are based on the load forecasting model that could predict the resource requirements of the virtual machine. However, the resource requirement of the VM in the IaaS Cloud is hard to predict because there will be variety of load types in IaaS cloud and no. of VMs. Moreover, a variety of heterogeneous hardware environments and virtualization technologies make it hard to predict the requirement of virtual machine based on the workloads.

To address the problem, we propose a method to predict and calculate the resource requirement of each virtual machine means forecasting resource requirements and using this model to design a load balancing framework in IaaS Cloud. This includes: (1) A model that forecast the load and estimate the resource requirement of virtual machines in IaaS Cloud; (2) A framework for load balancing [1].

## 2.4 BUSINESS CASE STUDY : LOAD BALANCING PROSPECTS FOR INFRASTRUCTUE AS A SERVICE [ IaaS] USING MODEL BASED METHOD

Infrastructure as a Service provides hardware as a service to an organization, so that it can put anything into the hardware according to its will. IaaS allows the user to “rent” resources as:

- Server space
- Network equipment
- Memory
- CPU cycles
- Storage space

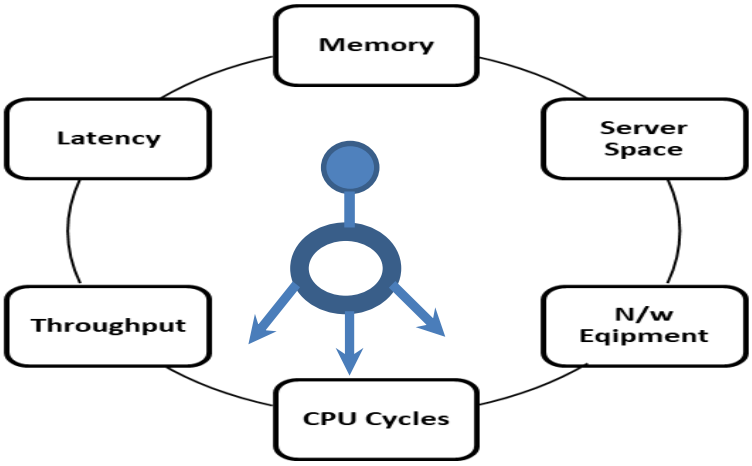



FIGURE 2.6 PARAMETERS NEED FOR IAAS

As shown in Figure 2.6 load balancing is the process of reassigning the total load to the individual nodes of the objective system to make resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are overloaded, while some other are under loaded[3]. A load balancing algorithm which is dynamic in nature does not consider the previous or behavior of the system. That depends on the present behavior of the system. The important thing to consider while developing such an algorithm are: estimation of load, comparison of load, stability of different system, performance of system, interaction between nodes, nature of work to be transformed, etc. This load consideration can be in terms of CPU load, amount of memory used, delay or network load. [7]

Cloud computing has been considered as a solution for solving enterprise application distribution and configuration challenges in the traditional software sales model. IaaS refers to a combination of hosting, hardware provisioning and basic services needed to a cloud. The uses of IaaS are as follows:

- Provides access to shared resources on need basis, without revealing details like location and hardware to clients.
- Provides details like server image on demand, storage queuing, and information about others.
- Offers full control over server infrastructure, not limited specifically to applications, instances and containers

The major issues that are commonly associated with IaaS in cloud systems are resource management, virtualization and multi-tenancy, data management etc. This proposal focuses on resource management due to the following fact that resource management for IaaS in cloud computing offers following benefits: scalability, Quality of Service (QoS), specialized

environment, reduction in overheads and latency, improved throughput, cost effectiveness and simplified interface.

At any time instant, resources are to be allocated to effectively handle workload fluctuations, while providing QoS guarantees to the end users. The computing and network resources are limited and have to be efficiently shared among the users in virtual manner. In order to perform effective resource management, we need to consider the issues such as resource mapping, resource provisioning, resource allocation and resource adaptation [1]. The development of efficient service provisioning policies is among the major issues in cloud research. The issue here is to provide better quality of service in IaaS, by provisioning the resources to the users or applications via load balancing mechanism, high availability mechanism, etc.

To address the problem, we propose a model based method to predict and calculate the resource requirement of each virtual machine and using this model to design a load balancing framework in IaaS Cloud. The contribution of this proposal includes:

- A model that forecasts the load and estimates the resource requirement of virtual machines in IaaS Cloud
- A scalable framework for load-balancing which uses our resource requirement forecasting model.

Our IaaS cloud comprises the following:

- A cloud resources pool composed of virtual machine monitor cluster (VMMC) and shared storage server;
- A collector of virtual machines and hosts performance data;
- A controller, which manages the VMs and physical hosts in VMMC based on our load-balancing strategies.

Forecasting the load of the VM is a key issue in IaaS cloud in order to perform more efficient resource provisioning. To this end, we are developing an on-line load prediction technique based on Exponential Smoothing. Exponential Smoothing (ES) is a simple prediction method based on historical and current data that works very well in practice. ES is a procedure for continually revising a forecast in the light of more recent experience. There are several types of ES's. Here we consider a triple exponential smoothing, which is also named Holt-Winters [1,8]. With this kind of ES the trend and seasonality of data are taken into account for the predictions.

In practice, we found that different types of virtual machines (different virtualization technologies) and heterogeneous physical (Intel or AMD) hosts will impact on the relationship between load of virtual machine and resource utilization of host. Here we use Canonical correlation analysis (CCA) to describe the relationship between the load of virtual machine and the resource utilization of host.

The load-balancing method is for two cases, one for VM starting, and the other for load exceeding or falling below the threshold. VMMC acts as the resource pool to host virtual machines. The master node collects performance data, establishes models and sends control instructions to VMM based on load-balancing strategy [1].

## 2.5 IMPLEMENTATION LOGIC

Load balancing is a technique in which the workload on the resources of a node is shifted to respective resources on the other node in a network without disturbing the running task.

The goals of load balancing are:

- To improve the performance substantially.
- To have a backup plan in case the system fails even partially.
- To maintain the system stability.
- To accommodate future modification in the system.

### 2.5.1 PROPOSED LOAD BALANCING

First, we get the next several hours prediction load of the starting VM. Then, we select 'n' hosts that have lower load in VMM. Then, one suitable host will be selected from these 'n' hosts for the VM running on. The principle for choose this host is that, if this VM running on the host, the load-balancing factor will be the minimum in next several hours.

We define the load-balancing factor as the mean square deviation for these hours:

L=Load

n=no of hosts

S= Load balancing factor

$$S\big(L(Hostx)\big) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(Li(Hostx) - \overline{Li(Hostx)})2} \quad \text{---------- (1)}$$

For the second case which the load of some hosts exceeds or falls below the threshold (here threshold is parameter like CPU utilization, memory or I/O utilization) several hosts with lowest load and highest load will be selected, and some suitable VMs on high load hosts

would be chosen to migration to the hosts with lower load. Similarly, the selection principle for virtual machines and hosts is based on the exponentially smoothed value using triple exponential smoothing technique. In the extreme case, the load of every host is below the threshold, we need migrate the VMs of some hosts and shutdown these hosts. Conversely, if the load of every host is higher than the threshold, new hosts would start to proceed for requests. [1]

Statistical approaches are based on a various mathematical models which expresses the relationship between load and several other input parameters. The predictions are done using following approaches:

- Time Series Forecasting
- Exponential Smoothing
- Neural Networks

From these we will use exponential smoothing for our work.

### 2.5.2 EXPONENTIAL SMOOTHING

The weights are given exponentially which decrease over time to make recent data more influential for forecast. A parameter called the smoothing factor is also introduced to lessen the effect of past data. Exponential smoothing does not totally exclude past data but gives them a weaker influence. [2]

$$St== \alpha * y_t + (1 - \alpha)\, S_{t-1} \qquad \text{---------- (2)}$$

### 2.5.3 DOUBLE EXPONENTIAL SMOOTHING

Double Exponential Smoothing is exponential smoothing of singly exponential smoothing. It is suitable for time series with linear trend. The prediction formula is as follows [2].

$$S_t = \alpha * y_t + (1 - \alpha) * (S_{t-1} + b_{t-1}) \qquad 0 < \alpha < 1 \qquad \text{---------- (3)}$$

$$b_t = \gamma * (S_t - S_{t-1}) + (1 - \gamma) * b_{t-1} \qquad 0 < \gamma < 1 \qquad \text{---------- (4)}$$

$S_t$ =Overall smoothing

$b_t$= Trend Component

Our proposed load balancing algorithm works as follows:

```
Let VMid = the VM which will start
Let Thresbottom = the bottom threshold for the load of VMMC
Let Threstop = the top threshold for the load of VMMC
Let n = the number of hosts that the VM could running on
Input:VMid,Thresbottom,Threstop,n
Output: None
loop
        {Get the t hours load predictions of the starting VM}
        VMPreload<-Get-LoadPrediction(VMid)
        {Select n hosts with lower load, if the load of all host above the
        Threstop,start a new host and return this host}
        Thosts<-Select-LowHosts(n)
        {Get the t hours load predictions of Target hosts}
        For each host in Thosts
                VMs<-Select-AllVMs(each host)
                {Get load prediction of every VM on host}
                PreLoads<-Get-LoadPrediction(VMs)
                {Get resource use by relation model and predict load}
                HRes<-Get_ResFromLoad(VMs,PreLoads,eachhost)
                {Add the resource information in a list}
                AllHres.add(Res)
        end For
        {Select the suitable host for the VM running on}
        For eachres in AllHres
                Pres<-Get_ResFromLoad(Pload.VMId,Pload,eachres.HostId)
                PHres<-ResAdd(eachres,Pres)
                {Compute the mean square deviation of all hosts' resource utilization}
                Fact<-Get-Fact(PHres,AllHres)
                {The minimum Fact means suitable host}
                IF LFact>Fact
                        Lfact<-Fact
                        LHost<-eachres.HostId
                end IF
        end For
        {Start the VM on this host}
        StartVM(VMid,LHost)
end loop
```

### 2.5.4 TRIPLE EXPONENTIAL SMOOTHING

This method is used when the data shows trend and seasonality. To handle seasonality, we have to add a third parameter. There are two main HW models, depending on the type of seasonality:

- Multiplicative Seasonality Model
- Additive Seasonality Model

The multiplicative seasonal model is appropriate for a time series in which the amplitude of the seasonal pattern is proportional to the average level of the series, i.e. a time series displaying multiplicative seasonality [2]:

$$R_t = \alpha*(y_t/S_{t-L}) + (1-\alpha)*(R_{t-1} + G_{t-1}) \quad 0 < \alpha < 1 \quad \text{---------- (5)}$$

$$G_t = \beta*(S_t - S_{t-1}) + (1-\beta)*G_{t-1} \quad 0 < \beta < 1 \quad \text{---------- (6)}$$

$$S_t = \gamma*(y_t/S_t) + (1-\gamma)*S_{t-L} \quad 0 < \gamma < 1 \quad \text{----------(7)}$$

Rt = estimate of the depersonalized level.

Gt= estimate of the trend

α, β, γ = smoothing constants

Exponential smoothing shows better results comparatively with MA and WMA since it uses current as well as previous data. Given data: alpha=0.8, gamma =0.3 , beta =0.2

TABLE 2.2 EXPERIMENTAL VALUES

| Pos | Weight |
|---|---|
| 1 | 1 |
| 2 | 0.75 |
| 3 | 0.5 |
| 4 | 0.25 |

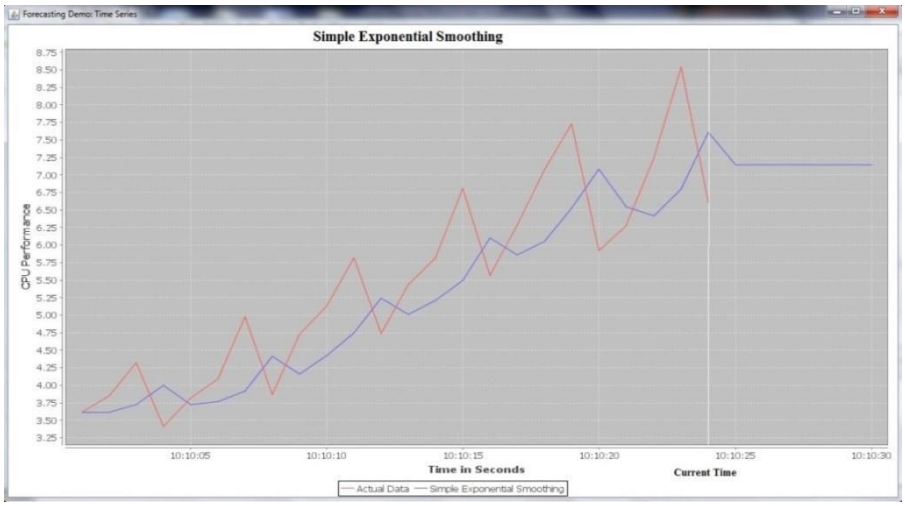



FIGURE 2.7 SIMPLE EXPONENTIAL SMOOTHING

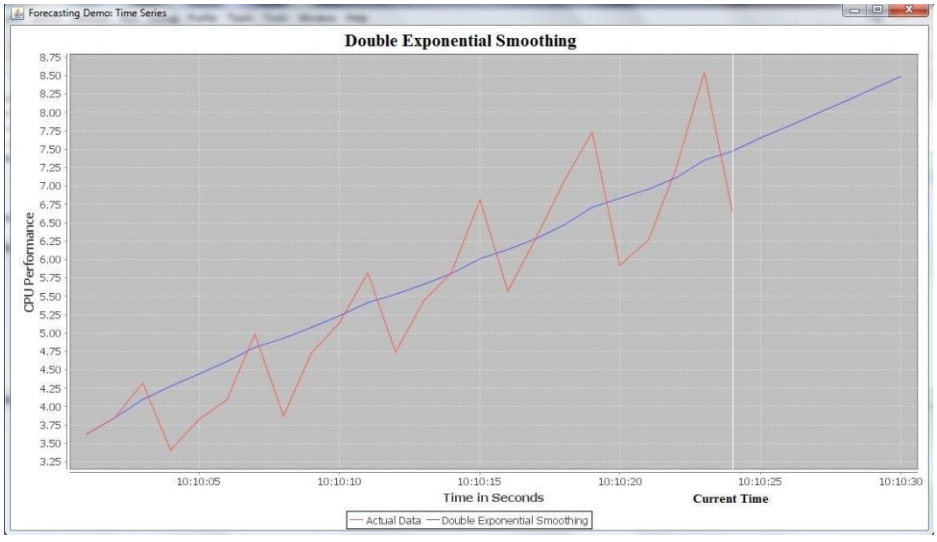



FIGURE 2.8 DOUBLE EXPONENTIAL SMOOTHING

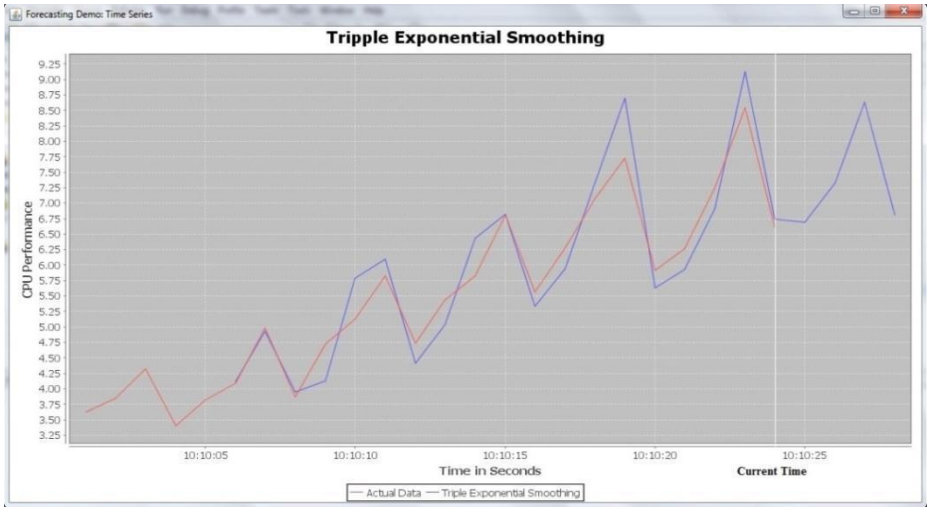



FIGURE 2.9 TRIPLE EXPONENTIAL SMOOTHING

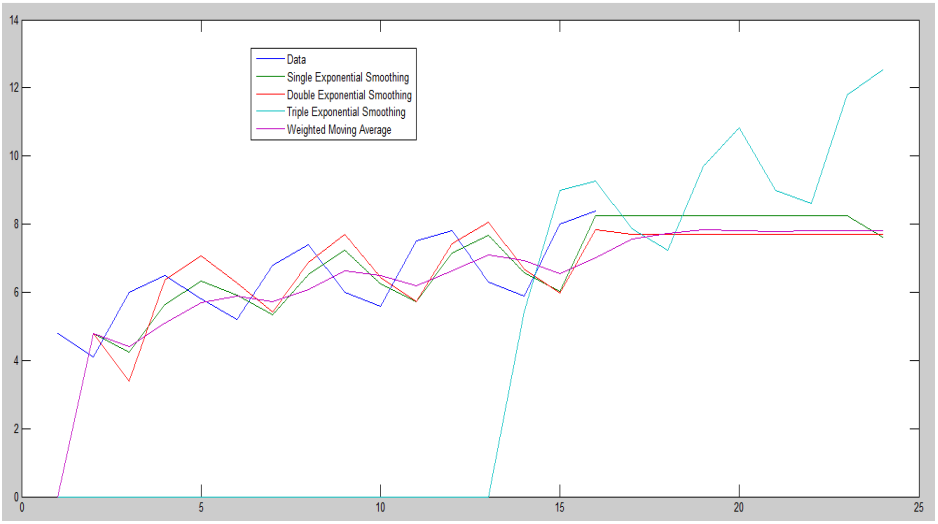



FIGURE 2.10 COMPARISON OF LOAD FORECASTING METHODS

As shown in Figure 2.7 i.e. simple exponential smoothing, graph represent blue color represents actual data as well as red color indicates simple exponential smoothing w.r.t CPU utilization. It also represents actual access data without any smoothing need many variation and need more CPU utilization too.

As comparing the graphs in Figure 2.9 and 2.10, Figure 2.10 itself represents to access actual data after triple exponential green color shows balanced data as associating with blue and yellow color. So applying different exponential methods Load balancing helps in reasonable share of registering asset to accomplish a high client fulfillment and legitimate resource usage .

Resource utilization and proper load balancing helps in limiting asset utilization. It helps in executing bomb over, versatility, and maintaining a strategic distance from blockage.

## 2.6 TEST CASE: - VIDEO ON DEMAND

Video on demand (VOD) is a service, by means of which one can access any video as and when required. In our proposal we are willing to a provide a live VoD service where a user can get a live video feed from a camera in cloud on demand.

Video application requirements depend upon several factors, including:

- Coding algorithms.
- Degree of motion required in the image sequence
- Resolution required in the image.
- QoS requirements for Data

### 2.6.1 TRADITIONAL ARCHITECTURE FOR VIDEO ON DEMAND

In a video service, contents are sent to subscribers through different server elements and different network provisioning aspects. In most video delivery network, the underlying network architecture is already designed as a hierarchical networks The difference lies in the service provider's network dimensioning strategy. Content Delivery Network (CDN), is based on a hierarchy of the servers located at specific places of the network – often located at the core of the network. If one server fails, the same content could be available in the other servers at the upper hierarchical levels, insinuating that there will be additional cost to maintain video availability. When the server capacity is fixed and no storage space is available, the least popular video will be removed from the servers. On the other hand, the increase in video catalogue size, being more frequently than before, will increase the number of VoD service requests, aggravating the interconnection network, especially when most of those videos are streamed from remote servers. Due to this, many recent literatures show that designing a scalable VoD architecture is essential for the viability of VoD services in broadband context [8],[12],[14],[15],17].

Therefore servers need to be redesigned and to employ dynamic management scheme in order to reduce network traffic and delays while minimizing the subscriber's waiting time, especially when substantial interest is being placed for services beyond voice [8].

### 2.6.2 ON-DEMAND CLOUD ARCHITECTURE FOR VIDEO APPLICATION

On-demand videos can be delivered to sub-scribers through different network structures – i.e. the video server location and the network between the video servers to the subscriber. For many cases, proxy server, located closer to the subscribers, is widely used to decrease network

traffic and delays through high speed and robust connection [12, 28]. However, proxy server has finite storage and distribution capacity, and therefore, a popularity scheme is needed to assist in the selection of videos during caching. Video servers, on the other hand, have a finite capacity and can only service limited request at one time. For large content library and the unforeseen spikes in number of active subscribers, to the service calls rejection should be kept to an absolute minimum. Videos can be streamed from any of the virtual servers, irrespective of its capacity, which was align continuously , notably to handle peak loads, to avoid overload and to achieve continuous, high utilization levels of servers while meeting its Service Level Agreements (SLAs). In most cases, performance is not affected as each virtual server behaves as a dedicated server. However, when too many virtual servers reside on the single physical machine, services may be delivered slower [28].

Currently, service operator uses the centralized architecture, with a great deal of redundant bandwidth, as a way of controlling access and limiting hardware cost, for delivering on-demand videos. But this is not a viable solution for large content library, as it may result in performance bottlenecks, long waiting delays and bad experience for subscribers, as they may not find this application interesting. Cloud computing is a well-known solution for rising IT costs, security issues and the increasing complexity of network management – plus its green credentials in reducing emissions, especially when multiplied by millions of users [8].

### 2.6.3 DEPLOYMENT DIAGRAM

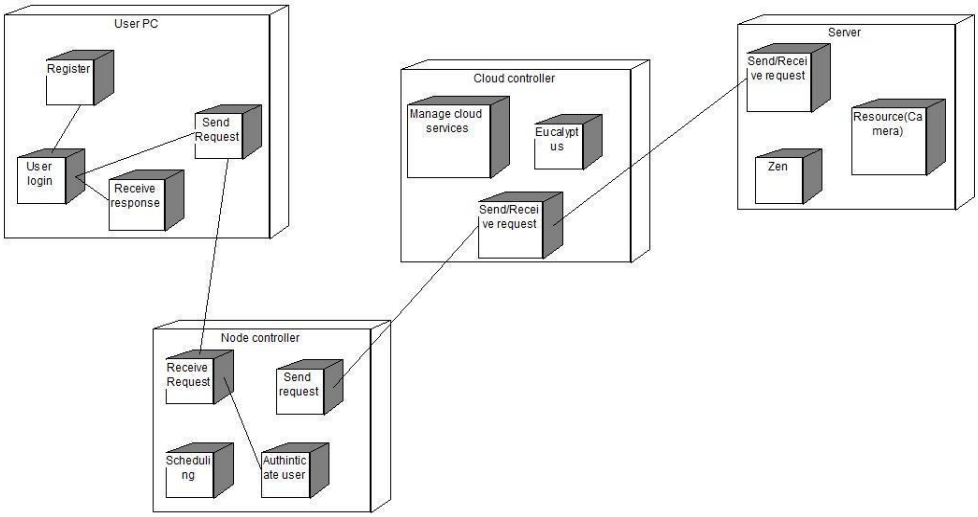



FIGURE 2.11 DEPLOYMENT DIAGRAM

In our proposal, as shown in Figure 2.11 we are using camera on server. The drivers required for the camera are installed on the main server and not on the client server. If the user wants to access the camera then the client request is sent to the server. If the requested

resource is free then it is allocated to the client. If the server has many requests for the resource then the algorithm is applied to select the request. The GUI application will be developed at the client side which has many functions through which the client interacts with the server. The selection of the request is based on various attributes such as the access time, priority and many others. At the end the quality of the developed application is measured in terms of throughput, CPU utilization, etc. There can be many clients. In our proposal the server and the client communicate through local area network.

There are two controllers in this application. The cloud controller handles the overall working of the cloud and the node controller is at the client side. The request is first sent to the node controller then the request is forwarded to the cloud controller and the response from server is directly sent to the client instead of the node controller.

## 2.7 CLOUD BASED VOD

As shown in Figure 2.12 shows the architecture of the cloud based VoD. The Node Controller controls the camera. The User requests through the cloud controller and receives the live feed of the video from the camera through the cloud controller.

On-demand videos can be delivered to sub-scribers through different network structures – i.e. the video server location and the network between the video servers to the subscriber. For many cases, the proxy server, located closer to the subscribers, is widely used to decrease the network traffic and delays through high speed and robust connection

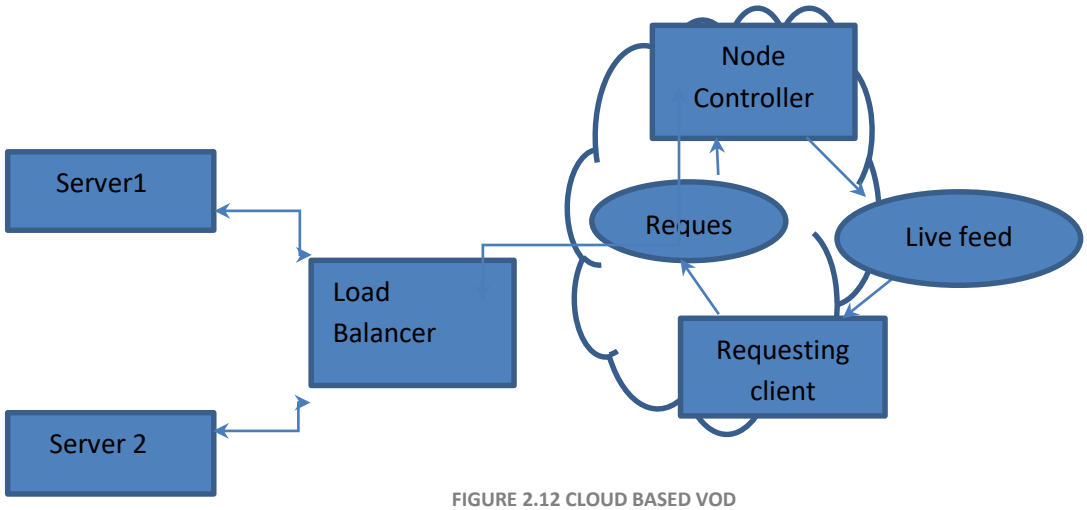



FIGURE 2.12 CLOUD BASED VOD

But the proxy server has a finite storage and distribution capacity, and therefore, a popularity scheme is needed to assist in the selection of videos during caching. Video servers, on the other hand, have a finite capacity and can only service limited request at one time. Cloud-Brokers

represent the behavior of specific IaaS systems, most of the measurements and decisions are made based on their behavior. All measurements are related to the queues of the Cloud-Broker; therefore we summarize their queuing behavior. Cloud-Brokers offer two types of queues: the call queue (Qx, where x identifies the specific Cloud-Broker that handles the queue) and the VM queues (VM Qx,y, where y identifies the specific service – or appliance V Ay – the queued VMs are offering). The members of the call queue represent the service calls that a Cloud-Broker needs to handle in the future (the queue is filled by the meta-broker and emptied by the Cloud-Broker through associating a call with a specific VM). On the other hand, VM queues are handled on a more complex way: they list the currently handled VMs offering a specific service instance. Consequently, the Cloud-Brokers maintain VM queues for all service instances separately. Entries in the VM queues are used to determine the state of the VMs [1]:

*State : VM → { WAITING, INIT, RUNNING.AVAILABLE, RUNNING.ACQUIRED, CANCEL }----(8)*

**Waiting:** the underlying cloud infrastructure does not have resources to fulfil this VM request yet.

**Init:** the VM handler started to create the VM but it has not started up completely yet.

**Running and available:** the VM is available for use the Cloud-Broker can associate calls to these VMs only.

**Running and acquired**: the VM is associated with a call and is processing it currently.

**For cancellation:** the Cloud-Broker decided to remove the VM and stop hosting it in the underlying infrastructure.

Based on these two queues the monitor collects the metrics listed in the following paragraphs [1].

To support decisions for service call rescheduling, the system monitors the call queue for all available Cloud-Brokers for a specific service call s:

$$q(x,s) := \{c \in Qx : (type(c) = s)\}, \qquad \text{--------(9)}$$

Where type(c) defines the kind of the service call c is targeting.

Call throughput measurement of available Cloud-Brokers is also designed to assist call rescheduling:

$$\text{Throughput (x)} \frac{1}{\max c \in Qx\ (\text{waitingtime(c)})} \qquad \text{---------- (10)}$$

Where waiting time(c) expresses the time in sec a service call has been waiting in the specific equation [1].

We define the average waiting time of a service s by

$$\text{awt}(s, Qx) = \frac{\text{Summation } c \in q(x,s) \text{waitingtime}(c)}{|q(x,s)|} \quad \text{---------- (11)}$$

and the average waiting time of a queue by

$$\text{awt}(Qx) = \frac{\text{Summation } c \in Qx \text{waitingtime}(c)}{|Qx| .} \quad \text{---------- (12)}$$

To distinguish the Cloud-Brokers, where VM queue rearrangements could occur, we measure the number of service instances that are offered by a particular infrastructure:

$\text{vms}(x, s) := \{ \text{vm} \in \text{VMQ } x,s :$

State (vm) = RUNNIN..AVAILABLE

$\vee$ State (vm) = RUNNING.ACQUIRED} ---------- (13)

The call/VM ratio for a specific service managed by a specific Cloud-Broker:

$$\text{cvmratio}(x, s) = \frac{|q(x,s)|}{|\text{vms}(x,s)|} \quad \text{---------- (14)}$$

This ratio allows the global autonomous manager to plan VM queue rearrangements and equalize the service call workload on the federated infrastructures. When applied with the local KM system, this ratio allows the system to decide on extending and shrinking the VM queues of particular services and balance the service instances managed by the local Cloud-Broker [1].

The load of the infrastructure managed by a specific Cloud-Broker:

$$\text{Load}(x) := \frac{\text{summation } \forall s |\text{vms}(x, s)|}{\text{Summation } \forall s |\text{VMQx,s}|} \quad \text{---------- (15)}$$

The load analysis is used for VM queue rearrangements in order to reduce the number of waiting VMs in the federation. When applied locally, along with the call/vm ratio the load analysis is utilized to determine when to extend or shrink the VM queues of various services. As a result, Cloud-Brokers could locally reorganize their VM structures that better fit the current call patterns. To support the remaining autonomous actions, the FCM repository and individual service instances are also monitored [1].

First, the system monitors the accumulated storage costs of a virtual appliance in all the repositories (r ∈ R) in the system (expressed in US dollars/day):

$$\text{stcost}(\text{V As}) := \text{summation } (\text{locstcost}(r, \text{V As}) ) , \quad \forall r \quad \text{---------- (16)}$$

Where locstcost(r, V As) signifies the local storage cost at repository r for appliance V As. To better identify the possible appliance storage rearrangements the system also analyzes the usage rate of appliances in the different repositories expressed in the number of times the VMs

based on the appliance have changed status from INIT to RUNNING.AV AILABLE in a single day (deplorer(r, V As)) [1].

Finally, individual services are monitored to support self-instantiated deployment. Here we analyze the service availability (expressed as the % of time that the instance is available for external service calls) of the specific service instance deployed in the same VM where the monitoring system is running [1].

## 2.8 SUMMARY

Figure 2.13 and 2.14 show the web interface for cloud-based VoD. Load balancing is a system that helped systems and assets by giving a most extreme throughput with least reaction time. The most essential thing is choosing the hubs and its additionally incorporate numerous different ones. CPU stack, size of memory required join together to compute the stack of machine. In our day by day life case of load balancing is site as shown in Figure 2.13 web reference.

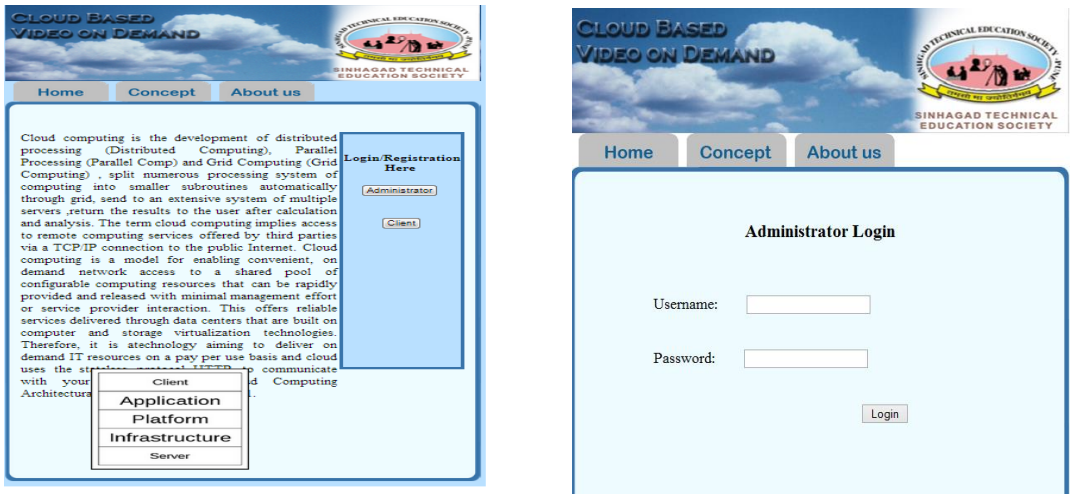



FIGURE 2.13 CLOUD BASED VOD ADMIN LOGIN


FIGURE 2.14 CLIENT USAGE AND LOGIN INFORMATION AUTHENTICATION

In this chapter with the help of case study introduces a new advanced method for controlling the remote camera. It provides new mechanism for video surveillance, which uses load balancing for providing more flexibility, also it is cost effective. By making use of cloud technology it also delivers high throughput.

Load balancing is mainly contingent on virtualization. Virtualization implies which are not exist in genuine, but rather it gives everything like genuine. Virtualization is the programming usage of a machine which will execute diverse projects like a genuine machine.

Through the virtualization client can utilize the assorted applications or administrations of the cloud, so this is the primary piece of the cloud environment. So the role of virtualization is more intricate in next Chapter 3.

**REFERENCES**


[1] R. Buyya, “High Performance Cluster Computing: Architectures and Systems,” Volume 1. Prentice Hall PTR, 1999.

[2] R. Mata-Toledo and P.Gupta “Green data centre: how green can we perform” Journal of technology , research academic and business research institute , vol2,No1,May 2011, pages 1-8

[3] D. Bader and R. Pennington, “Cluster computing: Applications," Georgia Tech College of Computing.

[4] L.Wenzheng and S.Hongyan, “A novel algorithm for load balancing in cluster systems," in 14th International Conference on Computer Supported Cooperative Work in Design, 2010.

[5] S.Sharma, S.Singh, and M.Sharma, “Performance analysis of load balancing algorithms," in World Academy of Science, Engineering and Technology, 2008.

[6] Z. Xu and R. Huang, “Performance study of load balancing algorithms in distributed web server systems," in CS213 Parallel and Distributed Processing Proposal Report.

[7] H. Stone, “Critical load factors in two-processor distributed systems," in IEEE Trans. Software Eng., vol. 4, no. 3, IEEE.

[8] R. Motwani and P. Raghavan, “Randomized algorithms," in ACM Computing Surveys (CSUR), ACM.

[9] L.Rudolph, M.Slivkin-Allalouf, and E.Upfal, “A simple load balancing scheme for task allocation in parallel machines.," in Proceedings of the 3rd ACM Symposium on Parallel Algorithms and Architectures, 1991.

[10] W. Leinberger, G. Karypis, and V. Kumar, “Load balancing across near homogeneous multi- resource servers.," 2000.

[11] Y.Wang and R. Morris, “Load balancing in distributed systems," in IEEE Trans. Computing.C-34, no. 3, IEEE.

[12] C.Youn and I.Chung, “An efficient load balancing algorithm for cluster system," in H. Jin, D. Reed, and W. Jiang (Eds.): NPC 2005, LNCS 3779, IFIP International Federation for Information Processing

[13] P.Werstein, H.Situ, and Z.Huang, “Load balancing in a cluster computer," in Proceedings of the Seventh International Conference on Parallel and Distributed Computing,Applications and Technologies, IEEE

[14] P.S.Pacheco, Parallel Programming with MPI. Morgan Kaufmann Publishers,Inc.

[15] P.Mohammadpour, M.Sharif, and A.Paikan, “A self training algorithm for load balancing," in Fourth International Conference on Networked Computing and Advanced Information Management, 2008

[16] T. C. Chieu, A. Mohindra, A. A. Karve and A. Segal, “Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment”. 2009 IEEE International Conference on e-Business Engineering pp 281

[17] Wei Sun, Xin Zhang, Chang Jie Guo, Pei Sun, Hui Su, Software as a Service: Configuration and Customization Perspectives, 2008, IEEE Congress on Services Part II, pp 18

[18]

[19] J. Namjoshi, A. Gupte, “Service Oriented Architecture for Cloud based Travel Reservation Software as a Service,” 2009 IEEE International Conference on Cloud Computing, pp 147.

[20] H. Mehta, P. Kanungo, and M. Chandwani, “Decentralized content aware load balancing algorithm for distributed computing environments”, Proceedings of the International Conference Workshop on Emerging Trends in Technology (ICWET), February 2011, pages 370-375.

[21] A. M. Nakai, E. Madeira, and L. E. Buzato, “Load Balancing for Internet Distributed Services Using Limited Redirection Rates”, 5th IEEE Latin-American Symposium on Dependable Computing (LADC), 2011, pages 156-165.

[22] Y. Lua, Q. Xiea, G. Kliotb, A. Gellerb, J. R. Larusb, and A. Greenber, “Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services”, An international Journal on Performance evaluation, In Press, Accepted Manuscript, Available online 3 August 2011.

[23] Xi. Liu, Lei. Pan, Chong-Jun. Wang, and Jun-Yuan. Xie, “A Lock-Free Solution for Load Balancing in Multi-Core Environment”, 3rd IEEE International Workshop on Intelligent Systems and Applications (ISA), 2011, pages 1-4.

[24] J. Hu, J. Gu, G. Sun, and T. Zhao, “A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment”, Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), 2010, pages 89-96.

[25] A. Bhadani, and S. Chaudhary, “Performance evaluation of web servers using central load balancing policy over virtual machines on cloud”, Proceedings of the Third Annual ACM Bangalore Conference (COMPUTE), January 2010.

[26] H. Liu, S. Liu, X. Meng, C. Yang, and Y. Zhang, “LBVS: A Load Balancing Strategy for Virtual Storage”, International Conference on Service Sciences (ICSS), IEEE, 2010, pages 257-262.

[27] Y. Fang, F. Wang, and J. Ge, “A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing”, Web Information Systems and Mining, Lecture Notes in Computer Science, Vol. 6318, 2010, pages 271-277.

[28] A. Singh, M. Korupolu, and D. Mohapatra, “Server-storage virtualization: integration and load balancing in data centers”, Proceedings of the ACM/IEEE conference on Supercomputing (SC), November 2008.

[29] http://www.thbs.com/pdfs/EAI-Performance-Analysis-webMethods.pdf

# CHAPTER 3 ROLE OF VIRTUALIZATION

## 3.1 INTRODUCTION

Virtualization is an act of creating indirect version of unbelievable tasks execution. Here we can say sentiment something incredible, when it is not in attendance as reality. Now a days it is an inclination of computing efficiency. In the Virtualization which is responsible of escalating speediness and capabilities today's processor. To prove Green this involved for the level of indirection. The Main advantage of this developed web application would be accessing different operating system from remote place if static IP is provided.

Virtualization or remote access is really important User sitting at remote place need not to have anything except internet connection and major application of this software is user can access or load operating system from browser, at a time user can access multiple operating system in the browser depending on the server side hardware configuration.

Virtualization is one of the fundamental qualities of the cloud computing, which empowers different servers to keep running on a solitary gadget. As it were, this determination enables virtual servers to keep running on equipment that brings adaptability and asset usage [2]. Also, virtualization diminishes the power utilizations, as the quantity of the physical servers will be chopped down. Studies demonstrate that virtualized servers devour 90% less vitality than typical physical servers.

### 3.1.1 ROLE OF VIRTUALIZATION IN GREEN ITINERARY

Cloud computing utilizes virtualization as a base for provisioning administrations to the customer. Numerous working frameworks can be run on the single PC on the base of virtualization so usage of assets is expanded. For upgraded the effectiveness of server the equipment assets are joined. For the best possible asset usage the, PC engineering is utilizes programming called Hypervisor. It is likewise called Virtual Machine Monitor (VMM) for running the various working frameworks on the single host. Hypervisor gives the registering assets (memory, processor and data transfer capacity) to the virtual machine. Extension packages required for compatibility of the Virtual Box according to host OS Versions.

In virtual server conditions, staying aware of the continuing changes occurring on virtual servers necessitates that the heap be circulated crosswise over servers in a productive way. Moreover, it must be disseminated in light of server inertness. As server execution lessens, the load should progressively be dispersed to different servers in a straightforward way. With stack balancers, dealing with this change is programmed. When servers are recognized and designed,

load balancers naturally disperse organize movement to a pool of virtual servers. In the event that the pool limit is uneven because of a few servers sharing their ability with other occupied virtual machines, the heap balancer naturally identifies the uneven inertness and sends the new activity to more responsive servers.

The most developed load balancing provisions incorporate one of a kind innovation that enables VMware servers to give proactive load balancing dimensions. These load balancers utilize the all-around characterized VMware API to question VMware's Virtual Center to accumulate CPU stack, memory use and different status data for virtual servers running under VMware. They join this data with their own particular system execution insights and they utilize it to settle on clever choices about server accessibility and load. For example, you can computerize reactions once a specific system occasion or edge is come to. Or on the other hand you can control up or shut down a VMware server in light of system movement. Having the capacity to control VMware servers in light of system movement permits you save money on influence, cooling and administration cost of running the server farm.

Fundamental load-balancing is required in the server farm to give a faster, denser World Wide Web and Application encounter. Propelled load adjusting is a key part of a vitality productive, progressively provisioned virtualized condition.

### 3.1.2 NEED OF VIRTUAL BOX

Developed of the virtual box on browser it is virtualization software that allows one to run multiple virtual machine instances (guests) on a single physical machine. In this User need not to install the same virtual box on client PC. All users can access server side visitor’s operating System from remote PC. The Virtual Box we can install on existing Operating System as an application, this host application allows additional guest operating systems, each known as a Guest OS, to load and run, each with its own virtual environment. Nearly all operating system is supported in Virtual Box [3].

In Virtual Box it is possible to set up several instances of the same OS, so it is the static data containing the software that the virtual machine will run once started. It is stored on disc. Instance is running the virtual machine it starts from an image and it is capable of running an OS and processes. Performs the computation and & I/O operations [2]. Figure 3.1 Shows rings 0 to 3 of the ring architecture of virtual box. The operating system kernel runs at a higher privilege such as ring 0, and applications at a lower privilege such as ring 3. In software-based virtualization, a host OS has direct access to hardware while the remote machine or guest Oss i.e ring 2 have

limited access to hardware, just like any other application of the host OS. One approach used in x86 software-based virtualization to overcome this limitation is called ring deprivileging, in which involves running the guest OS at a ring higher than the guest user-mode code, running in ring 2, generally runs directly on the host hardware in ring 2 [7].


FIGURE 3.1 ARCHITECTURE DESIGN FOR VIRTUAL BOX [7]

### 3.1.3 BENEFITS OF PROPOSED SYSTEM

When we describe Virtual Box as a virtualization product, we refer to full virtualization, that is, the particular kind of virtualization that allows an unmodified operating system with all of its installed software to run in a special environment, on top of your existing operating system. The physical computer is then, usually called the host, while the virtual machine is often called a guest. Most of the guest code runs unmodified, directly on the host computer, and the guest operating system thinks it is running on a real machine. Users can administer Virtual Box in a headless environment, mirroring the Virtual Box GUI through its web interface.

- Headless Environment: Virtual Box GUI to manage your virtual machines, but a server does not have a desktop environment. Fortunately, Virtual Box comes with a tool called VBoxHeadless that allows you to connect to the virtual machines over a remote desktop connection, so there's no need for the Virtual Box GUI.

- Can access from Mobile and no need to install any application at Client side.
- User can access or load operating System from browser.
- User can access multiple operating systems in the browser depending on the server side hardware configuration.

### 3.1.4 OBJECTIVES OF THE SYSTEM

The following are the objectives of the systems with consideration of networking software:

- User can access it anytime from the remote location even from mobile and PC no needs to install same on client PC.
- Required Internet connection, both local and remote IP specification to access Guest OS.
- Giving end user Software as a service, commutation with multiple Guest operating systems from remote location. It provides user friendly Graphical User Interface features that understood by uses. Graphical User Interface contains three MENU items File Machine Help to reduce computing power, electricity charges, space cost in data canter one of the best solution is virtualization, virtualization is technology that divides combines the resources to the number of functional environments virtualization is used to increase the performance and functionality [2].

In Figure 3.2. Shows virtualization of software the guest operating systems operate in virtual machines within the virtualization application which, in turn, runs on top of the host operating system. The multiple layers of abstraction between the guest operating systems and the underlying host hardware are not conducive to high levels of virtual machine performance. This technique does, however, have the advantage that no changes are necessary to either host or guest operating systems and no special CPU hardware virtualization support is required.

This virtualization application one or more virtual machines are created to run the guest operating systems on the host computer. The virtualization application is responsible for starting, stopping and managing each virtual machine and essentially controlling access to physical hardware resources on behalf of the individual virtual machines.

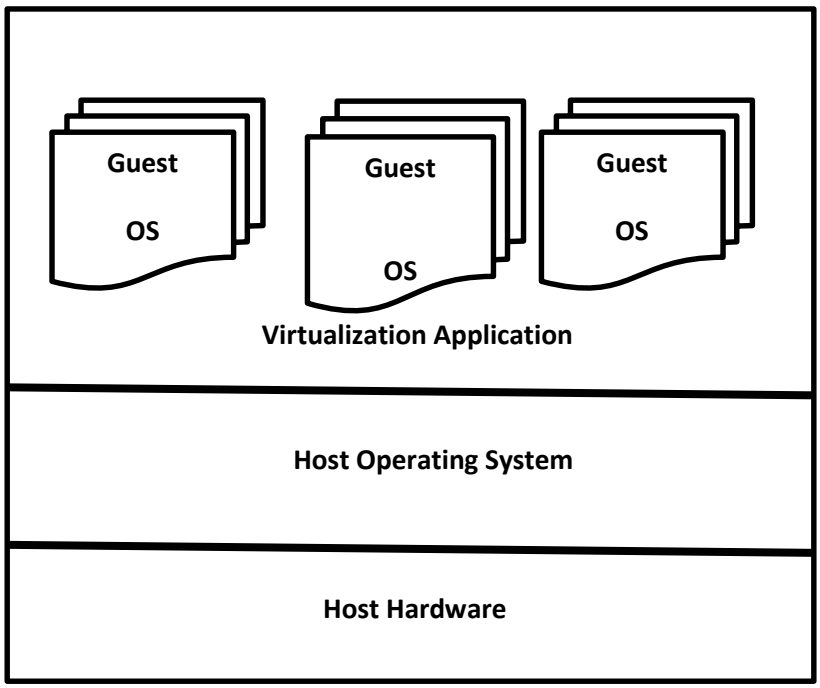



FIGURE 3.2 VIRTUALIZATION

It gives effect of making the guest system think it is running directly on the system hardware, rather than in a virtual machine within an application. Virtualization software allows a single host computer to create and run one or more virtual environments. Virtualization software is most often used to emulate a complete computer system in order to allow a guest operating system to be run, for example allowing Linux to run as a guest on top of a PC that is natively running a Microsoft Windows operating system (or the inverse, running Windows as a guest on Linux).

## 3.2 PROBLEM DEFINITION

Virtualization make it conceivable to see a whole machine as a document or set of records. Virtualization additionally gives an office to move a virtual machine among intensely stacked physical machines with the goal that equalization can be accomplished among them. The primary target of virtual machine relocation is to convey the load in a datacenter or set of datacenters. At that point in what manner can we powerfully disseminate the load while moving the virtual machine to dodge bottlenecks in cloud figuring frameworks. This thesis proposes how Extension packages required for compatibility of the Virtual Box according to host OS Versions.

3.3 CLASSIFICATION OF VIRTUALIZATION

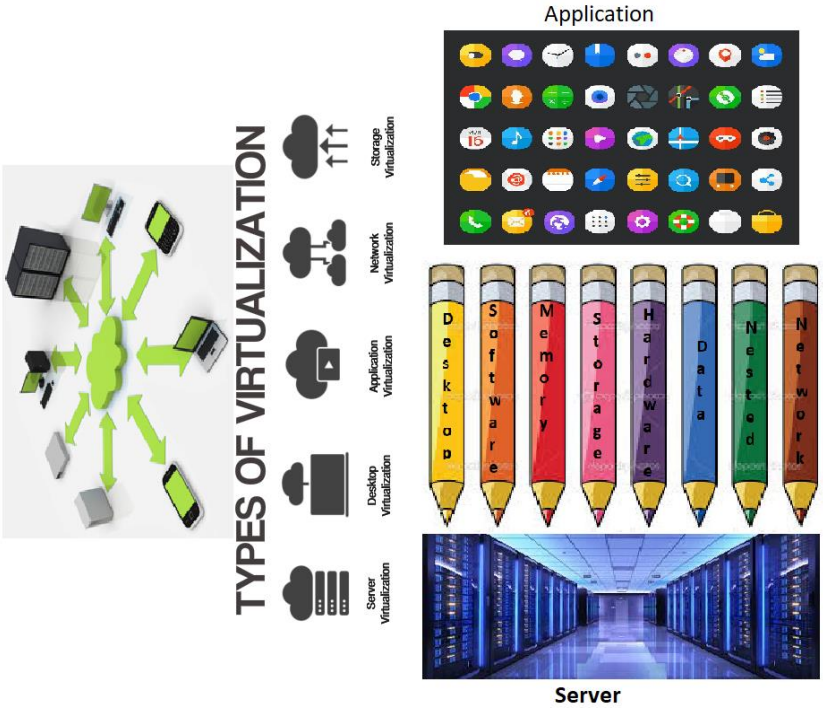



FIGURE 3.4 CLASSIFICATION OF VIRTUALIZATION [9,14]

- Hardware virtualization:

Hardware virtualization or platform virtualization refers to the creation of a virtual machine that acts like a real computer with an operating system. Software executed on these virtual machines is separated from the underlying hardware resources. For example, a computer that is running Microsoft Windows may host a virtual machine that looks like a computer with the Ubuntu Linux operating system; Ubuntu-based software can be run on the virtual machine [3].

In hardware virtualization, the host machine is the actual machine, on which the virtualization takes place, and the guest machine is the virtual machine. The words host and guest are used to distinguish the software that runs on the physical machine from the software that runs on the virtual machine. The software or firmware that creates a virtual machine on the host hardware is called a hypervisor or Virtual Machine Manager.

- Full virtualization– almost complete simulation of the actual hardware to allow software, which typically consists of a guest operating system, to run unmodified.
- Partial virtualization–some but not the entire target environment is simulated. Some guest programs, therefore, may need modifications to run in this virtual environment.

- Para virtualization–a hardware environment is not simulated; however, the guest programs are executed in their own isolated domains, as if they are running on a separate system. Guest programs need to be specifically modified to run in this environment.
- Desktop Virtualization- Desktop virtualization is the concept of separating the logical desktop from the physical machine. One form of desktop virtualization, virtual desktop infrastructure (VDI), can be thought of as a more advanced form of hardware virtualization. Rather than interacting with a host computer directly via a keyboard, mouse, and monitor, the user interacts with the host computer using another desktop computer or a mobile device by means of a network connection, such as a LAN, Wireless LAN or even the Internet. In addition, the host computer in this scenario becomes a server computer capable of hosting multiple virtual machines at the same time for multiple users.

As organizations continue to virtualized and converge their data center environment, client architectures also continue to evolve in order to take advantage of the predictability, continuity, and quality of service delivered by their converged infrastructure. For example, companies like HP and IBM provide a hybrid VDI model with a range of virtualization software and delivery models to improve upon the limitations of computing. Selected client environments move workloads from PCs and other devices to data canter servers, creating well-managed virtual clients, with applications and client operating environments hosted on servers and storage in the data canter. For users, this means they can access their desktop from any location, without being tied to a single client device. Since the resources are centralized, users moving between work locations can still access the same client environment with their applications and data. For IT administrators, this means a more centralized, efficient client environment that is easier to maintain and able to more quickly respond to the changing needs of the user and business.

Another form, session virtualization, allows multiple users to connect and log into a shared but powerful computer over the network and use it simultaneously. Each is given a desktop and a personal folder in which they store their files. With multi seat configuration, session virtualization can be accomplished using a single PC with multiple monitors keyboards or similar connected.

Thin clients, who are seen in desktop virtualization, are simple and/or cheap computers that are primarily designed to connect to the network. They may lack significant hard disk storage space, RAM or even processing power, but many organizations are beginning to look at the cost benefits of eliminating “thick client” desktops that are packed with software (and require software licensing fees) and making more strategic investments. Desktop virtualization simplifies software versioning and patch management, where the new image is simply updated on the server, and the desktop gets the updated version when it reboots. It also enables centralized control over what applications the user is allowed to have access to on the workstation.

- Software Virtualization- Hosting of multiple virtualized at OS level environment within a single OS instance. Application virtualization and workspace virtualization, the hosting of individual applications in an environment separated from the underlying OS. Application virtualization is closely associated with the concept of portable applications. Service virtualization, emulating the behavior of dependent (e.g., third-party, evolving, or not implemented) system components that are needed for application under test (AUT) and development or testing purposes, instead of vitalizing entire components, it virtualizes only specific slices of dependent behavior critical to the execution of development and testing tasks[6].
- Memory Virtualization- Memory virtualization is aggregating random-access memory (RAM) resources from networked systems into a single memory pool. Memories giving an application program the impression that it has contiguous working memory, isolating it from the underlying physical memory implementation.
- Storage Virtualization- virtualization is the process of completely abstracting logical storage from physical storage. Distributed file system, any file system that allows access to files from multiple hosts sharing via a computer network. Virtual, an abstraction layer on top of a more concrete file system, allowing client applications to access different types of concrete file systems in a uniform way Storage hypervisor, the software that manages storage virtualization and combines physical storage resources into one or more flexible pools of logical

storage. Computer program that emulates a disk drive such as a hard disk drive or optical disk drive.

- Data virtualization- Data virtualization is the presentation of data as an abstract layer, independent of underlying database systems, structures and storage. Database virtualization is the decoupling of the database layer, which lies between the storage and application layers.
- Network Virtualization- Network virtualization is creation of a virtualized network addressing space within or across network subnets; network virtualization is the process of combining hardware and software network resources and network functionality into a single, software-based administrative entity, a virtual network. Network virtualization involves platform virtualization, often combined with resource virtualization.
- Nested virtualization- Nested virtualization refers to the ability of running a virtual machine within another, having this general concept extendable to an arbitrary depth. In other words, nested virtualization refers to running one or more hypervisors inside another hypervisor, Nested guest virtual machine need not be homogenous with its host virtual machine; for example, application virtualization can be deployed within a virtual machine created by using hardware virtualization.

Nested virtualization becomes more necessary as widespread operating systems gain built-in hypervisor functionality, which in a virtualized environment can be used only if the surrounding hypervisor supports nested virtualization; for example, Windows 7 is capable of running Windows XP applications inside a built-in virtual machine. Furthermore, moving already existing virtualized environments into a cloud, following the Infrastructure as a Service (IaaS) approach, is much more complicated if the destination IaaS platform does not support nested virtualization[3].

The way nested virtualization can be implemented on particular computer architecture depends on supported hardware-assisted virtualization capabilities. In case a particular architecture does not provide hardware support required for nested virtualization, various software techniques are employed to enable it. Over time, more architectures gain required hardware support; for

example, since the Haswell micro architecture, Intel started to include VMCS shadowing as a technology that accelerates nested virtualization [6].

### 3.4 COMPARISON TO PROVE HOW VMVIRTUAL BOX PREFERABLE WITH VMWARE PLAYER LINUX PERSPECTIVE

TABLE 3.1 COMPARISON

| | VM Virtual Box | | VMware Player |
|---|---|---|---|
| 1 | To make "Shared Folders" work in "Virtual Box", you have to install the "Oracle VM Virtual Box Guest Additions" from inside each virtual machine. It installs from installable virtual CD. | 1 | To make "Shared Folders" work in "VMware Player", you have to install the "VMware Tools" from inside each virtual machine. It installs from a virtual CD. |
| 2 | "Virtual Box" have a virtual "Network Address Translation" router configuration. | 2 | “VMware Player" has a virtual "Network Address Translation" router configuration. |
| 3 | In "Virtual Box", the "Network Address Translation" router blocks file sharing in both directions between the host and virtual machines. This router also blocks file sharing between virtual machines. | 3 | In "VMware Player", you can disconnect the host from the "Network Address Translation" router. When the connection between the host and the "Network Address Translation" router is enabled, this router allows file sharing between the host and virtual machines and it also allows file sharing between virtual machines. |
| 4 | Both "Virtual Box" have a virtual "Bridged" bridge configuration. | 4 | “VMware Player" has a virtual "Bridged" bridge configuration. |
| 5 | "Virtual Box" have a virtual "Host-Only" router configuration. | 5 | "VMware Player" has a virtual "Host-Only" router configuration. |
| 6 | A virtual "Internal Network" (LAN switch) networking configuration is available in "Virtual Box". | 6 | "VMware Player" does not provide a virtual "Internet Network" (LAN switch) Networking configuration directly. |

According to above comparison implementing Virtual Box is feasible as its freeware and available worldwide for further modification and implementation.

There is difference in platform OS support, version, storage communication and important difference is headless mode that in Virtual Box uses to implement without graphical user interface. In headless mode point of view user can access VM on command line.

- They all support Windows 2000, Windows XP, Windows 2003, Windows Vista, Linux and Mac OS X as the host operating systems. But, Virtual Box is the only software that supports Windows 7, Windows 2008 Server, and Solaris, Open Solaris, FreeBSD (in the near future) as the host operating systems.
- All three software support DOS, Windows 3.1, 95, 98, NT, 2000, XP, Vista, Linux as the guest operating system. But again, Virtual Box is the only software that can load Windows 7, Windows Server 2003/2008, OpenBSD and OpenSolaris. VMware does not support OS/2, while Parallels does not support FreeBSD and Solaris as the guest operating system.
- Although, all three support 64-bit versions of guest operating systems, only Virtual Box and VMware support 64-bit host operating systems.
- Both VirtualBox and Parallels support Intel VT-x and AMD-V virtualization extensions, but this support is limited on VMware.
- VirtualBox, VMware and Parallels provide virtual network cards up to 8, 4 and 5, respectively.
- Both Virtual Box and VMware can support IDE or SATA virtual disk controllers, but Parallels will support only IDE. However, Virtual Box is the only software that supports iSCSI (which allows virtual machines to directly access storage servers over iSCSI).
- Although all there software provides Serial ports, only Parallels and VMware provide Parallel ports.
- Only Virtual Box supports CD/DVD writing.
- Furthermore, Virtual Box is the only virtualization software with unrestricted 3D acceleration. In fact, Parallels does not have any 3D acceleration capabilities.
- Out of Virtual Box and Parallels, only Virtual Box supports VMware images.
- Unlike Virtual Box and VMware, Parallels does not support Headless operation.
- VirtualBox is the virtualization software with unrestricted remote virtual machine access (with Integrated RDP server). In fact, Parallels does not have any remote access capabilities. Similarly, only Virtual Box supports remote USB access.

- Only Virtual Box and VMware provide reports on guest power status.
- Only, Virtual Box and VMware come with an API. But only Virtual Box is open source (with few closed source enterprise features).
- Unlike with Parallels and VMware, customizations are possible (upon request) with Virtual Box.
- Finally, Virtual Box is the only free virtualization software out of the three. However, Parallels is considerably cheaper than VMware.

### 3.5 METHODOLOGY

Current trend in the market is all related to virtualization accordingly transforming feasible systems at virtual world and giving the user a feeling as if accessing an actual system. Virtualization at the processor level requires exceptional programming that would actualize it.

Virtualization of the cloud processing enables workloads to be conveyed and scaled-out rapidly through the quick provisioning of virtual machines or physical machines. A cloud computing stage supports excess, self-recouping, very versatile programming models that permit workloads to recuperate from numerous unavoidable equipment/programming disappointments. By methods for virtualization advances, Cloud registering offers to end clients an assortment of administrations covering the whole figuring LOAD, from the equipment to the application level, by charging them on a compensation for every utilization premise. A few merchants, for example, Amazon Web Services and VMWare construct their offering in light of equipment level virtualization and give uncovered figure and capacity assets on request. Application level virtualization by implementing a particular application display that use their vast framework and scale all over on request with following approaches:-

1) Understanding the user requirement.
2) How Remote access of existing system is revolutionary
3) Ease and reliability of the system
4) Reduction in the cost
5) Understanding the types of virtualization
6) VBoxWeb makes use of the following technologies:

### 3.6. APPLIED VIRTUALIZATION TO APPLICATION ZONEMINDER.

The performance of load balancing algorithms is measured by the following parameters.

#### 3.6.1 OVERLOAD REJECTION

If Load Balancing is not possible additional overload rejection measures are needed. When the overload situation ends then first the overload rejection measures are stopped. After a short guard period Load Balancing is also closed down.

#### 3.6.2 FAULT TOLERANT

This parameter gives that algorithm is able to tolerate tortuous faults or not. It enables an algorithm to continue operating properly in the event of some failure. If the performance of algorithm decreases, the decrease is proportional to the seriousness of the failure, even a small failure can cause total failure in load balancing.

#### 3.6.3 FORECASTING ACCURACY

Forecasting is the degree of conformity of calculated results to its actual value that will be generated after execution. The static algorithms provide more accuracy than of dynamic algorithms as in former most assumptions are made during compile time and in later this is done during execution.

#### 3.6.4 STABILITY

Stability can be characterized in terms of the delays in the transfer of information between processors and the gains in the load balancing algorithm by obtaining faster performance by a specified amount of time.

#### 3.6.5 CENTRALIZED OR DECENTRALIZED

Centralized schemes store global information at a designated node. All sender or receiver nodes access the designated node to calculate the amount of load-transfers and also to check that tasks are to be sent to or received from. In a distributed load balancing, every node executes balancing separately. The idle nodes can obtain load during runtime from a shared global queue of processes.

#### 3.6.6 NATURE OF LOAD BALANCING ALGORITHMS

Static load balancing assigns load to nodes probabilistically or deterministically without consideration of runtime events. It is generally impossible to make predictions of arrival times of loads and processing times required for future loads. On the other hand, in a dynamic load balancing the load distribution is made during run-time based on current processing rates and network condition. A DLB policy can use either local or global information.

### 3.6.7 COOPERATIVE

This parameter gives that whether processors share information between them in making the process allocation decision other are not during execution. What this parameter defines is the extent of independence that each processor has in concluding that how should it can use its own resources. In the cooperative situation all processors have the accountability to carry out its own portion of the scheduling task, but all processors work together to achieve a goal of better efficiency. In the non-cooperative individual processors act as independent entities and arrive at decisions about the use of their resources without any effect of their decision on the rest of the system.

### 3.6.8 PROCESS MIGRATION

Process migration parameter provides when does a system decide to export a process. It decides whether to create it locally or create it on a remote processing element. The algorithm is capable to decide that it should make changes of load distribution during execution of process or not.

### 3.6.9 RESOURCE UTILIZATION

Resource utilization include automatic load balancing A distributed system may have unexpected number of processes that demand more processing power. If the algorithm is capable to utilize resources, they can be moved to under loaded processors more efficiently.

## 3.7 LOAD BALANCING W.R.T. ZONEMINDER

The algorithm proposed expects the user to provide input parameters as usage duration, number of cameras, and number of monitors and resolution of video. The usage duration is the amount of time for which the user wants to use the service. The second input parameter is number of cameras the user wants to connect to the system along with the total number of monitors. To use Zoneminder properly you need to define at least one Monitor. Essentially, a monitor is associated with a camera and can continually check it for motion detection and such like. The function of the monitor essentially defines what it is doing. This can be one of the following:

- 'None' – The monitor is currently disabled and no streams can be viewed or events generated.
- 'Monitor' – The monitor will only stream feeds but no image analysis is done and so no alarms or events will be generated,

- 'Modect' – or MOtionDEteCTtion. All captured images will be analyzed and events generated where motion is detected.
- ‘Record’ – In this case continuous events of a fixed length are generated regardless of motion which is analogous to a convention time-lapse video recorder. No motion detection takes place in this mode.
- ‘Mocord’ – This is a hybrid of Modect and Record and results in both fixed length events being recorded and also any motion being highlighted within those events.
- 'Nodect' – or No DEteCTtion. This is a special mode designed to be used with external triggers. In Nodect no motion detection takes place but events are recorded if external triggers require it.

So one camera can have maximum five number of different monitors performing different functions. The resolution of the input video has been restricted to 320 X 240 or 640 X 480 of which the user can select anyone. Once the input has been accepted the user request proceeds to the load balancer where the algorithm would execute as follows:

The algorithm can be divided into three sections, which are explained in the following subsections.

### 3.7.1 SCALING

Scaling in reference to our system is the ability of dynamically invoking and suspending the virtual resources as and when required. So the following routine will execute at regular intervals of time to determine if any of the virtual resource should be suspended.

Check for each VM and if there is VM with 0 processes running in it, shutdown the particular VM.

```
For each instance i in N_VirtualResources
If (NUsers_i ==0)
        Shut down the instance
        Decrement N_VirtualResources
End for
```

3.7.2 SCHEDULING

Scheduling is basically to allocate the most appropriate virtual resource to the user request. To do this we are calculating the weightage of the user request with respect to the number of monitors the user wants to create.

Request_Weightage =

$$\frac{0.4*(M_u/M_{MAX}) + 0.3*M_u*CPU_{monitor} + 0.3*M_u*RAM_{monitor}}{\text{Sum } k=1, N_{VirtualResources}\ 0.4*(M_k/M_{MAX}) + 0.3*CPU_k + 0.3*RAM_k}$$

where

$M_u$: Number of monitors requested by user

$M_{MAX}$: Maximum number of monitors per instance

$N_{VirtualResources}$: Number of virtual resources available

$CPU_{monitor}$: CPU usage for one monitor.

$RAM_{monitor}$: RAM usage for one monitor

$M_k$: Number of active monitors on instance k

$CPU_k$: Current CPU usage of $k^{th}$ instance

$RAM_k$: Current RAM usage of $k^{th}$ instance

So this value of Request_Weightage gives us a fraction the load this user request is going to put on the system out of the current load on the system.

Then we calculate, for each Virtual machine, its maximum available capacity and minimum available capacity. The maximum available capacity is calculated considering the currently active monitors on that machine and current CPU and RAM usage of that machine.

$$Maximum\_available\_capacity_i = \frac{0.4(1-M_i/M_{MAX}) + 0.3(1-CPU_i) + 0.3(1-RAM_i)}{\text{Sum } k=1, N_{VirtualResources}\ 0.4(1-M_k/M_{MAX}) + 0.3(1-CPU_k) + 0.3(1-RAM_k)} \quad \ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Whereas for the minimum available capacity we consider the maximum number of monitors that the current users can activate on that machine in future i.e. the possible number of monitors. This value is calculated because there may be x number of cameras being handled by that particular instance with some y number of monitors per camera. But we have restricted the number of monitors per camera to 5. So if there are x cameras then there could be x*5 number of possible monitors that users can add up to. If the users increase the number of monitors then load will be increased so we must consider this future condition.

Thus we find out the minimum available capacity of the instance by considering possible number of monitors on $i^{th}$ instance as:

$$Possible\_M_i = NCameras_i * 5$$

$$Minimum\_available\_capacity_i = \frac{0.4(1-Possible\_M_i/ M_{MAX}) + 0.3(1-CPU_i) + 0.3(1- RAM_i)}{\text{Sum } k=1, N_{VirtualResources}\ 0.4(1-Possible\_M_k/ M_{MAX}) + 0.3(1-CPU_k) +0.3(1- RAM_k)} \quad \text{.....(3)}$$

With these two above values we calculate the average capacity of the instance. So we have now established one of the main conditions for selecting the instance that is - the Request Weightage should be less than the average capacity of the instance. After that we calculate the time for which this instance will surely be running. So if the duration for which the current user wants to use the system is less than the time for which instance is surely up and running we can allocate the user request to that instance provided it obeys the first condition. Otherwise we keep this instance as an option. The reason to do this is that we will be able to shut down the instance after it has served all its current requests.

To calculate the sure up time for the instance, we find maximum Start Time + Usage Duration, for each user currently active on this instance. So Current Time – MAX of (Start Time + Usage Duration) is the time for which this instance will surely be up means running.

Thus the second condition to select the instance is that the sure up time of that instance should be greater than the duration for which the user wants to use the system. If at all we don't get any instance that satisfies the second condition we select the one instance having maximum sure up time from the instances we have kept as an option.

As shown in Figure 3.5 the general architecture to balance the load on virtual resources consists of a load balancer which executes the load balancing algorithm, a number of virtual instances which altogether provide the virtualized environment. The section of the load balancing algorithm and the number of virtual instances running depends on the number of users requesting the services. All the virtual instances fetch and access the centralized MySQL database for the users' data.

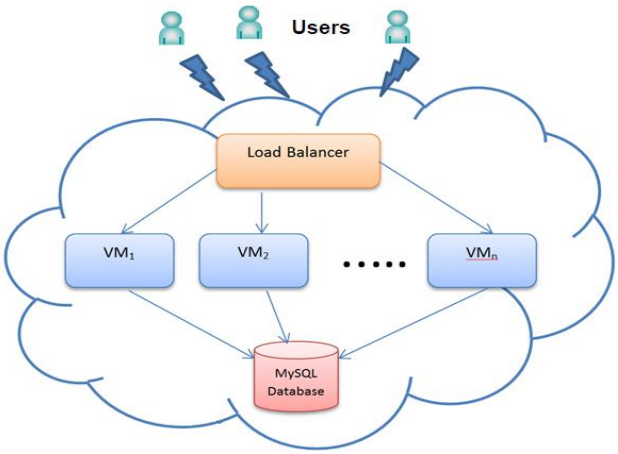



FIGURE 3.5 ARCHITECTURE TO BALANCE THE LOAD ON VIRTUAL RESOURCES

## 3.8 VIRTUALIZED LOAD BALANCING PROPOSAL

When the user request will be allocated to the most appropriate instance the load on the virtual resources will be shifted hence this load has to be balanced again. The next part of the algorithm does the same.

### 3.8.1 PSEUDO CODE FOR LOAD BALANCING ALGORITHM

*Calculate for every machine*
*$Needed_i$ = Request_Weightage – $Current\_Load_i$*

*Compare $Needed_i$ with every user's load on $i^{th}$ Machine*
*If ($Needed_i$ = User_Load)*
*Select user, migrate it to other machine using scheduling algorithm*
*Put the current user on this machine*
*Return*
*Else*
*Find the pair of users who can be migrated to other machine and put current request on that machine*
*If no such pair is found*
*Start new instance*
*If no instance available to start*
*Ask user to wait*
*Where,*
*$Needed_i$ = the amount of capacity needed to serve the current request on $i^{th}$ virtual machine*
*$Current\_Load_i$= the current lad on $i^{th}$ virtual machine*
*User_Load = the load that particular user putting on the system*

It finds out the needed capacity on each virtual machine in order to serve the request. It then finds out if any of the current users is putting the same load on that machine as the needed capacity. If such user exists he is migrated to another instance using the scheduling algorithm and the current user is allocated that machine. Otherwise it finds a pair of users who can be

migrated to other machine and puts current request on that machine. If it doesn't find even such pair users then a new machine is started.

### 3.8.2 PSEUDO CODE FOR THE SCHEDULING ALOGORITHM

*Accept $NCameras_u$, No of monitors ($M_u$) and Usage_Duration from the user.*

*Request_Weightage =*

$$0.4*(M_u/M_{MAX}) + 0.3*M_u*CPU_{monitor} + 0.3*M_u*RAM_{monitor}$$

$$\text{Sum } k=1, N_{VirtualResources}\ 0.4*(M_k/M_{MAX}) + 0.3*CPU_k + 0.3*RAM_k$$

*For each VM from i to $N_{VirtualResources}$*

*$Maximum\_available\_capacity_i$ =*

$$0.4(1-M_i/M_{MAX}) + 0.3(1-CPU_i) + 0.3(1-RAM_i)$$

$$\text{Sum } k=1, N_{VirtualResources}\ 0.4(1-M_k/M_{MAX}) + 0.3(1-CPU_k) + 0.3(1-RAM_k)$$

*$Possible\_M_i = NCameras_i * 5$*

*$Minimum\_available\_capacity_i$ =*

$$0.4(1-Possible\_M_i/M_{MAX}) + 0.3(1-CPU_i) + 0.3(1-RAM_i)$$

$$\text{Sum } k=1, N_{VirtualResources}\ 0.4(1-Possible\_M_k/M_{MAX}) + 0.3(1-CPU_k) + 0.3(1-RAM_k)$$

*$Sure\_Up\_Time_i$ = Current_Time – MAX (Start_Time + Usage_Duration)$_i$*

*//Update the Capacity_Matrix*

*Capacity_Matrix[i][Max_Capacity]= $Maximum\_available\_capacity_i$*

*Capacity_Matrix [i] [Min_Capacity] = $Minimum\_available\_capacity_i$*

*Capacity_Matrix [i] [Sure_Up_Time] = $Sure\_Up\_Time_i$*

*Now we have the VMi's minimum, maximum available capacity with sure up time:*

*$Avg\_Capacity_i$ =*

*(Capacity_Matrix[i][Min_Capacity]+Capacity_Matrix[i][Max_Capacity]) /2*

*If (Request_Weightage < $Avg\_Capacity_i$)*

*If (Capacity_Matrix [i] [Sure_Up_Time] > Usage_Duaration)*

*Select instance*

*Return Else*

*Candidate[k] = i*

*Increment k*

*End If*

*End for all VM*

*Select instance from Candidate having highest Sure_Up_Time*

### 3.8.3 CALCULATING THE CONSTANTS

In order to implement this algorithm we had to carry out some tests to find the values for the maximum number of monitors a virtual machine can handle at a time ($M_{MAX}$), CPU and RAM usage for one monitor etc.

To find out the CPU and RAM usage for one monitor we used the 'top' command in Ubuntu to conclude that every new monitor of resolution 640 x 480 consumes 6% of the CPU and 1.5% of the RAM. So those are the values for the $CPU_{monitor}$ and $RAM_{monitor}$ respectively.

To find the value for maximum number of monitors on a virtual machine ($M_{MAX}$) we conducted tests to get the data shown in table 3.1. We continuously added the monitors and after regular intervals calculated the corresponding CPU usage and Memory usage.

TABLE 3.2 CPU USAGE VS RAM USAGE

| CPU usage | RAM usage | No Of Monitors Running |
|---|---|---|
| 7.5 | 786292 | 0 |
| 14.4 | 879328 | 5 |
| 24.2 | 1104028 | 10 |
| 21.5 | 1322084 | 15 |
| 27.4 | 1708896 | 20 |
| 30 | 1847428 | 25 |
| 34.9 | 1861952 | 30 |
| 42.5 | 1877572 | 35 |
| 47.6 | 1882800 | 40 |
| 50.2 | 1880248 | 45 |

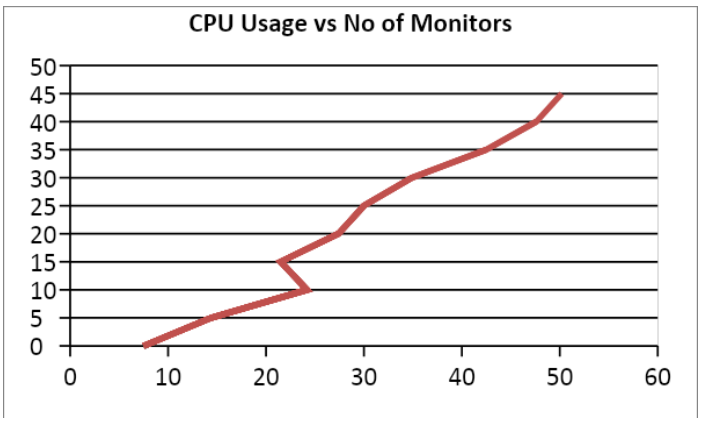



FIGURE 3.6: RELATION BETWEEN CURRENT CPU USAGE OF THE VIRTUAL MACHINE AND NUMBER OF ACTIVE MONITORS ON THAT MACHINE

From these values we concluded that when there are 45 active monitors on a virtual machine it consumes about 50% of the CPU. So we decided on value of $M_{MAX}$ as 90. The main contribution towards the proposal of load balancing algorithm for proper scheduling of the user requests in the virtualized environment. The load balancer executes this algorithm on a regular

basis in order to distribute the computational load evenly among the virtual machines. The decision to execute which section of the algorithm is taken by the load balancer considering the current scenario of the whole system. We believe that this algorithm will provide an efficient and reliable service to the users willing to use Zoneminder as a Service.

1. ***Input***: Multiple Requests from distributers.

2. ***Output*:** order completed successfully.

Model Using Probability Theory:

Probability is needed at the load balancing phase. Here

P(gi) = Probability of order get generated.

P(di) =Probability of order get delivered on or before end date.

P(ci)= probability of order get completed.

P(gi) is depend on quantity of stock , end date , start date.

P(di) is depend on available transpoters and capacity of vehicle.

By using Bayes Theorem,

P(ci)=p[(di)/(gi)]

= {p[(gi)/(di)] * p(di)}/ p(gi)

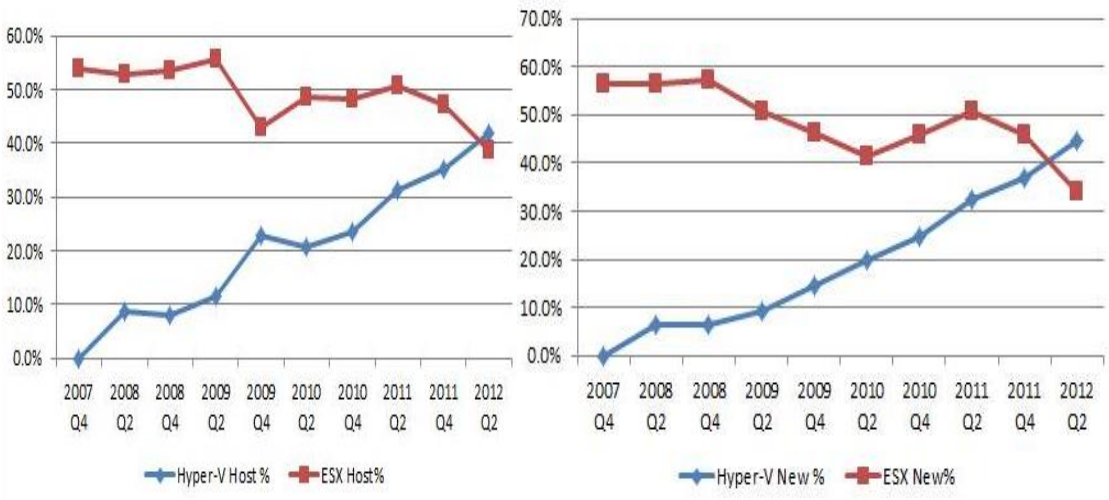



FIGURE 3.7 APPLIED CLOUD WITH VIRTUAL BOX BEFORE AND AFTER

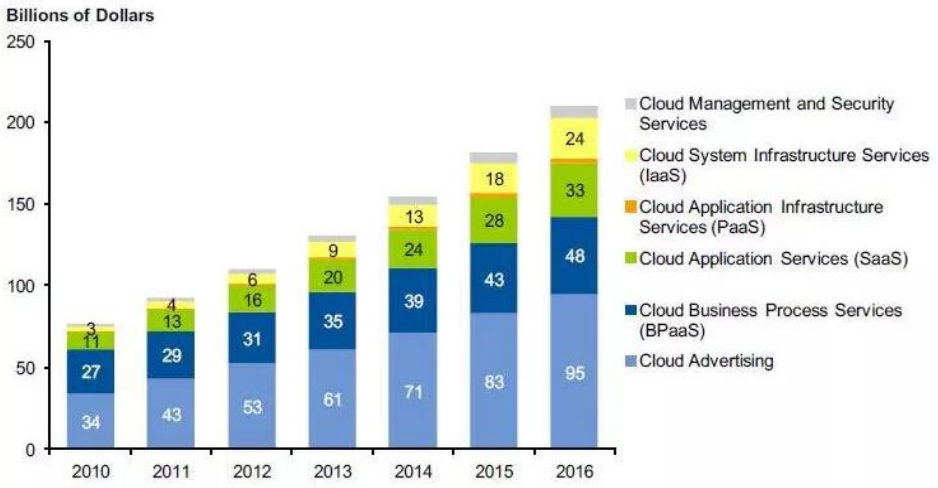



FIGURE 3.8 USAGE OF VIRTUALIZATION WORLDWIDE

## 3.9 SUMMARY

VMs and holders are two essential virtualization advancements for conveying visitors in server farms. Despite the fact that VMs should have high overheads, this proposal demonstrates that the hole between the VMs and compartments isn't as high as implied. We assess two basic components for a cloud supplier, inactivity and thickness, as for VMs and holders. The startup time for VMs is 2 requests of greatness higher than that of compartments however the start-up time can be diminished by a request of size utilizing checkpoint-reestablish. The impact of pressing more visitors on CPU bound outstanding task at hand is the same on compartments just as VMs, and the outright execution of VMs and holders is equal for CPU-bound outstanding task at hand. VMs languish over I/O bound outstanding tasks at hand more than Containers on account of continuous VM exits which have been unraveled by research works like [16,18]. Virtualization is a foundational innovation for conveying cloud-based framework that enables a solitary physical server to run different working framework pictures simultaneously. As an empowering influence of combination, server virtualization diminishes the aggregate physical server impression, which has inalienable green advantages.

From an asset proficiency point of view, less gear is expected to run workloads, which proactively decreases server farm space and the possible e-squander impression. From a vitality proficiency viewpoint, with less physical hardware connected to, a server farm will expend less power.

- Developing Virtual Box VMs on the server remotely and conveniently.
- The system is working and is capable of accessing from remote place on browser.
- The System will help user to load different operating system on server side system and accessing it from Remote location through internet.

- Each and every phase in this proposal is developed keeping the goals in mind as far as concerned.

In cloud processing, there are numerous services that are required to be executed by accessible assets while accomplishing best execution, negligible aggregate time for finishing, most limited reaction time, and usage of resources and so on. To accomplish these goals we have to configuration, create and propose a scheduling calculation, which will be further elaborated in Chapter 4.

**REFERENCES**


[1] Domingues, P. ; Araujo, F. ; Silva, LEvaluating the performance And Intrusiveness of virtual machine for desktop grid computing. Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on DOI: 10.1109/IPDPS.2009.5161134 Publication Year: 2009, Page(s): 1- 8

[2] Semnanian, A.A. ; Pham, J. ; Englert, B. ;Virtualization technology and its impact on computer hardware architecture Xiaolong Wu Information Technology: New Generations (ITNG), 2011 Eighth International Conference on DOI: 10.1109/ITNG.2011.127 Publication Year: 2011 , Page(s): 719- 724

[3] JAIN, R. ; Paul, S. Network virtualization and software defined networking for cloud computing Volume:51, Issue: 11 DOI: 10.1109/MCOM.2013.6658648 Publication Year: 2013

[4] Wen Huang, Xiaoyi Wei, Yixia Zhao;Efficient virtualization on real time activitiesService-Oriented Computing and Applications (SOCA), 2011 IEEE International Conference on DOI: 10.1109/SOCA.2011.6166248 Publication Year: 2011, Page(s): 1- 4

[5] Ashino, Y. ; Nakae, M.Virtual machine migration method between different hypervisor implementation and its evaluation ;Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on DOI: 10.1109/WAINA.2012.172 Publication Year: 2012 , Page(s): 1089- 1094

[6] Borislav, D. ; Timcenko, V. ; Verica, V. ; Branka, A ; Performance comparison of three virtualization software used in windows 7. Telecommunications Forum (TELFOR), 2013 21st DOI: 10.1109/TELFOR.2013.6716405 Publication Year: 2013 , Page(s): 1019- 1022

[7] Liuyang WANG, Yangxin YU; Design Intelligence for multi agent for virtualization resource in cloud computing Computer Science & Service System (CSSS), 2012 International Conference on Date of Conference:11-13 Aug. 2012 Page(s):729 - Publisher: IEEE

[8] http://www.cubrid.org/blog/dev-platform/x86-server-virtualization-technology/

[9] http://www.virtuatopia.com/index.php/An_Overview_of_Virtualization_Techniques

[10] http://www.informit.com/articles/article.aspx?p=1627061

[11] http://www.c-sharpcorner.com/uploadfile/nipuntomar/system-Development-life-cycle-part-2/

[12] The complete Reference HTML and CSS fifth edition Author: Thomas A. Powell Publication McGraw-Hill

[13] The complete Reference JavaScript third edition Author: Thomas A. Powell and Fritz Schneider Publication McGraw-Hill

[14] The complete Reference JavaScript Ajax Author: Thomas A. Powell and Fritz Schneider Publication McGraw-Hill

[15] The complete Reference PHP cover 5.2 Author: Steven Holzner Publication McGraw-Hill

[16] Software Engineering A Practitioner's Approach Author: Roger Pressman Publication McGraw-Hill

[17] https://wiki.archlinux.org/index.php/VirtualBox

# CHAPTER 4 RESOURCE MANAGEMENT

Cloud computing is an on demand service in which shared resources, information, software and other devices are provided according to the clients requirement at a specific time. Cloud computing is internet-based computing in which large groups of remote servers are networks to allow scheduling for auxiliary processing. As mentioned in Chapter 3 virtualization will not be effective without scheduling. In continuation in this chapter how scheduling can be fruitful with optimized way is proposed.

Software as a Service is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it. The client typically earnings on a per-use basis In this service model, cloud providers offer resources to users/machines that include computers as virtual machines, raw (block) storage, firewalls, load balancers, and network devices. One of the most persistent issue in cloud computing is the resource management of an effective load balancing algorithm [4].

As shown in Figure 4.1 On demand Services (Any SaaS, PaaS or IaaS) is key in Cloud Computing, which provides through virtual machines (VMs) to users. Proper deployment of the virtual machine onto available hosts plays crucial role on the load-balancing of modern data Centre. At present, many load-balancing methods are based on the load forecasting model that could predict the resource requirements of the virtual machine.

Virtualization is a structure or strategy of partitioning the resources into different execution situations, by applying at least one idea or advances, for example, segregating of hardware and software, time-sharing, simulation, emulation etc. It gives a scope of advantages to processing frameworks, for example, enhanced resource utilization and administration, application disconnection, convenience, live migration, transportability and framework unwavering quality. Virtualization innovation not just empowers IT work force to react to business needs more quickly yet additionally changing the IT conveyance model to give on-request self-benefit access to a common pool of figuring assets through expansive system.

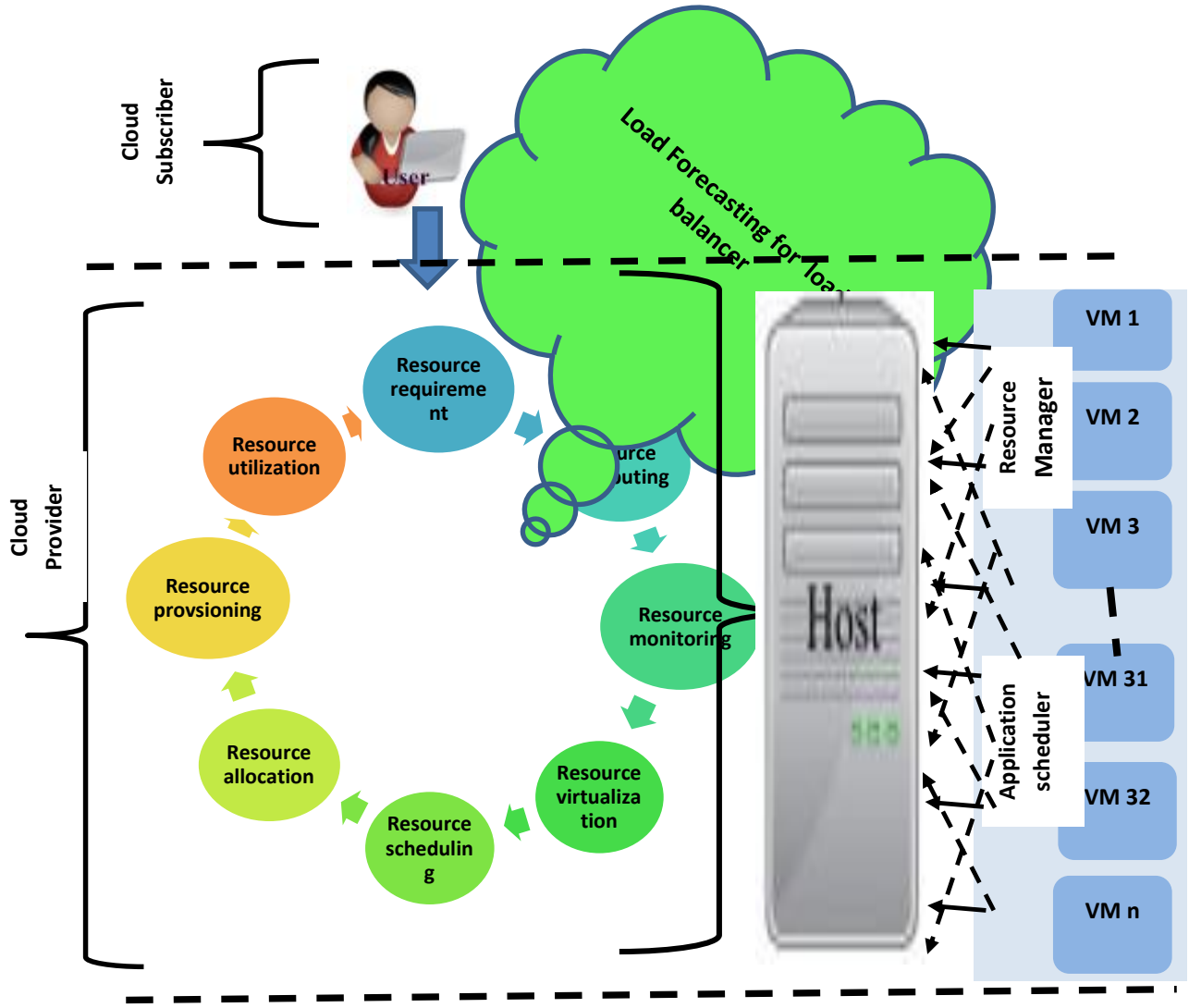



FIGURE 4.1 RESOURCE MANAGEMENT NEED VIRTUALIZATION

As shown Figure 4.1 there following are the steps for resource management for server side load balancing :-

### 4.1 RESOURCE REQUIREMENT

However, the resource requirement of virtual machine in Cloud is hard to predict because there will be variety of load types in cloud and no of VMs. Moreover, a variety of heterogeneous hardware environments and virtualization technologies make it hard to predict the requirement of virtual machine based on the workloads. To address the problem, we propose a model based method to predict and calculate the resource requirement of each virtual machine means forecasting resource requirements and using this model to design a load balancing framework in a cloud.

Customary resources administration strategies are not satisfactory for cloud computing as they are based on virtualization innovation with dispersed nature. Cloud computing presents

new difficulties for asset administration because of heterogeneity in equipment capacities, on-request benefit show, pay per utilize model and certification to meet QoS.

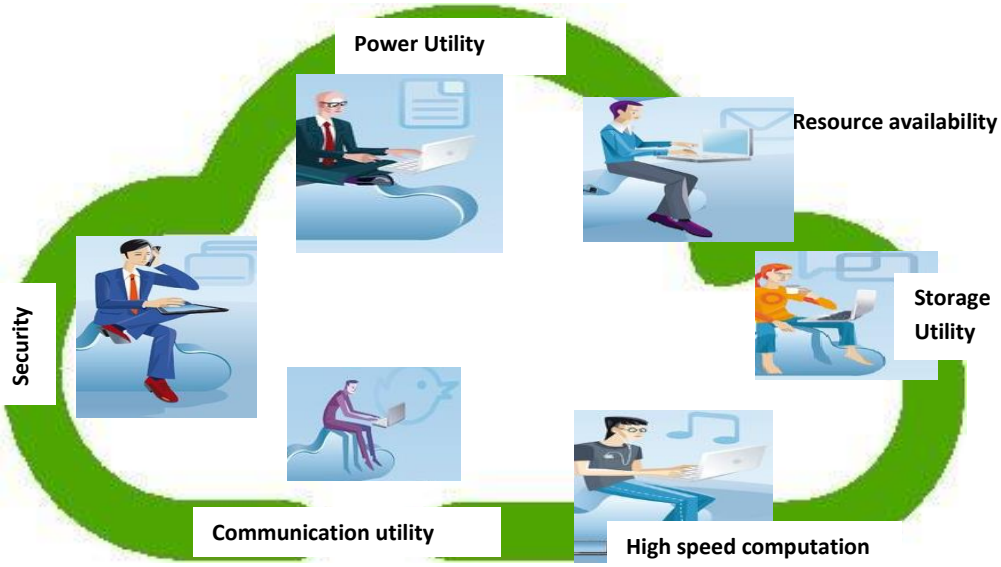



FIGURE 4.2 RESOURCE CLASSIFICATIONS

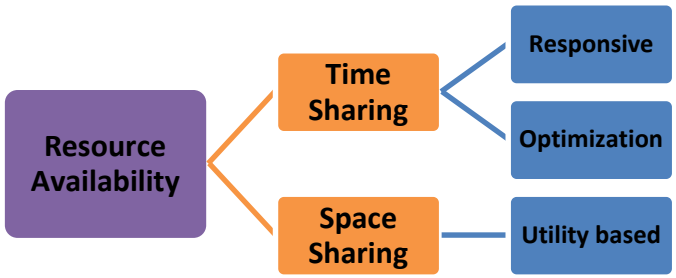



FIGURE 4.3 RESOURCE AVAILABILITY

As shown in Figure 4.3 resources will be available based on time sharing and space sharing based utility. Resource availability is the procedure in all working frameworks in which especially framework resource (e.g. Central Preparing Unit (CPU), random access memory, secondary storage device, and so forth.) is allocated to especially procedures, strings and applications. This is more often than not done to accomplish high throughput, nature of administration, decency what's more, adjust between all processes[1],[2],[3] .

To play out this overseeing level errand, we require planning calculations to guarantee that every one of the procedures share the system resources similarly as per need [3]. This service level agreement is the essential necessity for those frameworks that are performed multitasking and multiplexing [4],[5].

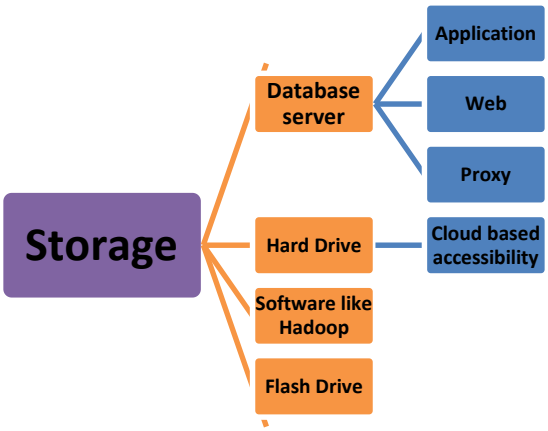



FIGURE 4.4 STORAGE FOR RESOURCE AVAILABILITY

Relatively putting away information at local storing gadget, we store them at capacity gadget which is situated at remote place. Capacity utility comprises of thousands of hard drives, streak drives, database servers and so forth. As PC frameworks will undoubtedly bomb over the time of time information excess is required here. Due to cloud's opportunity variation benefit show stockpiling utility needs to give highlights like cloud flexibility [12, 14]. Through capacity utility in cloud computing gives Storage as a Service (StaaS) as shown in Figure 4.4 .

Figure 4.5 precise likewise be called as Network Utility or Network as a Service (NaaS). Quick calculation utility and capacity utility cannot be thought without correspondence utility. Correspondence utility comprises of physical (intermediate devices, sensors, physical correspondence interface) and legitimate (data transmission, delay, conventions, virtual correspondence connect) assets. In distributed computing every last administration is given through rapid Internet. So data transmission and deferral are most critical from organize perspective [12, 14].

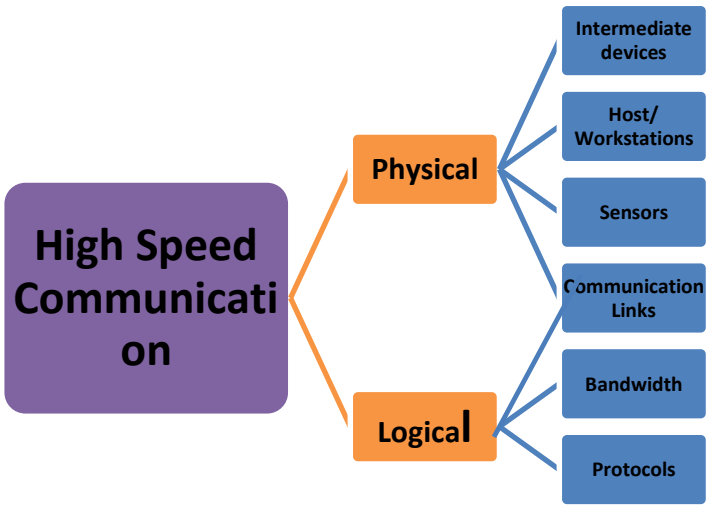



FIGURE 4.5 HIGH SPEED COMMUNICATION FOR RESOURCE AVAILABILITY

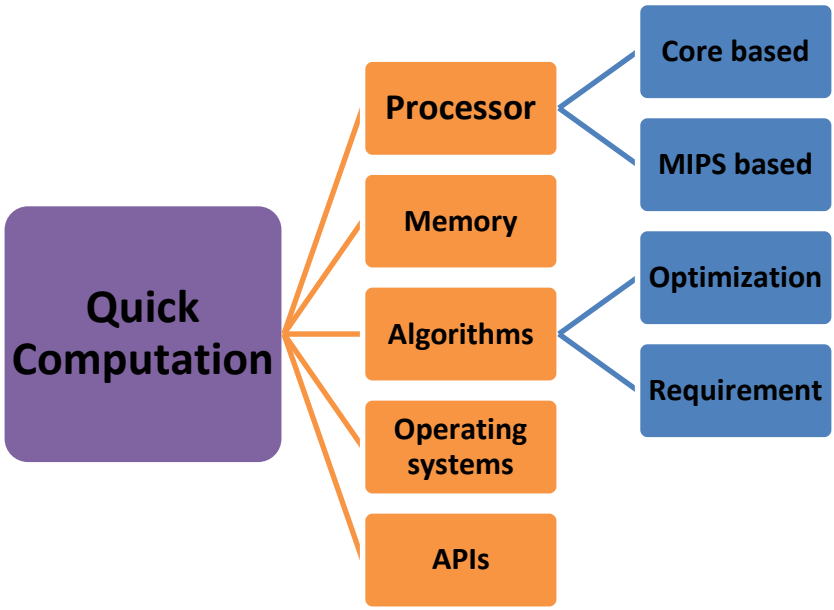



FIGURE 4.6 QUICK COMPUTATION FOR RESOURCE AVAILABILITY

As shown in Figure4.6 the sort of resources gives quick computational utility in the cloud computing condition. Through quick computation utility computing gives Computation as a Service (CaaS). Quick computation utility incorporates handling capacity, estimate of memory, effective calculations and so forth.

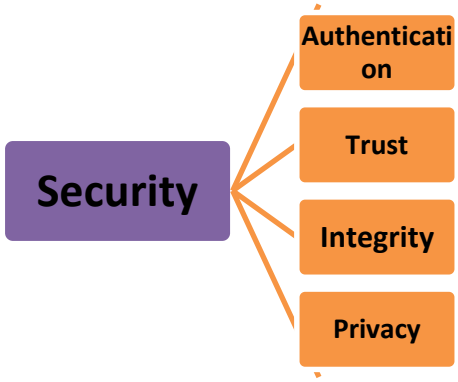



FIGURE 4.7 SECURITY FOR RESOURCE AVAILABILITY

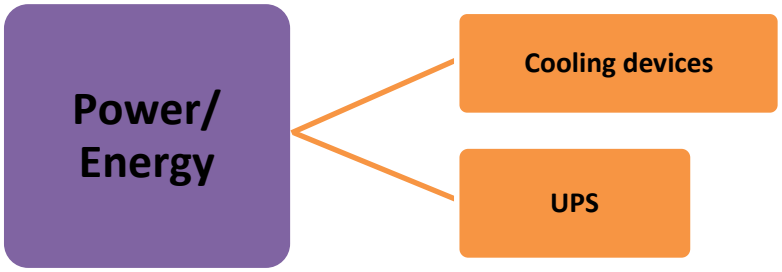



FIGURE 4.8 POWER ENERGY FOR RESOURCE AVAILABILITY

As shown in Figure 4.8 Power/Energy Utility: Now a days specialists are completing a great deal of research deal with vitality proficient strategies in distributed computing. Vitality

cost can significantly be diminished by utilizing power mindful methods. Because of thousands of information servers control utilization is high in cloud processing. Cooling gadgets and UPS are at the central point of these sort of resorces. They can likewise be considered as optional resorces.[11, 12, 14].

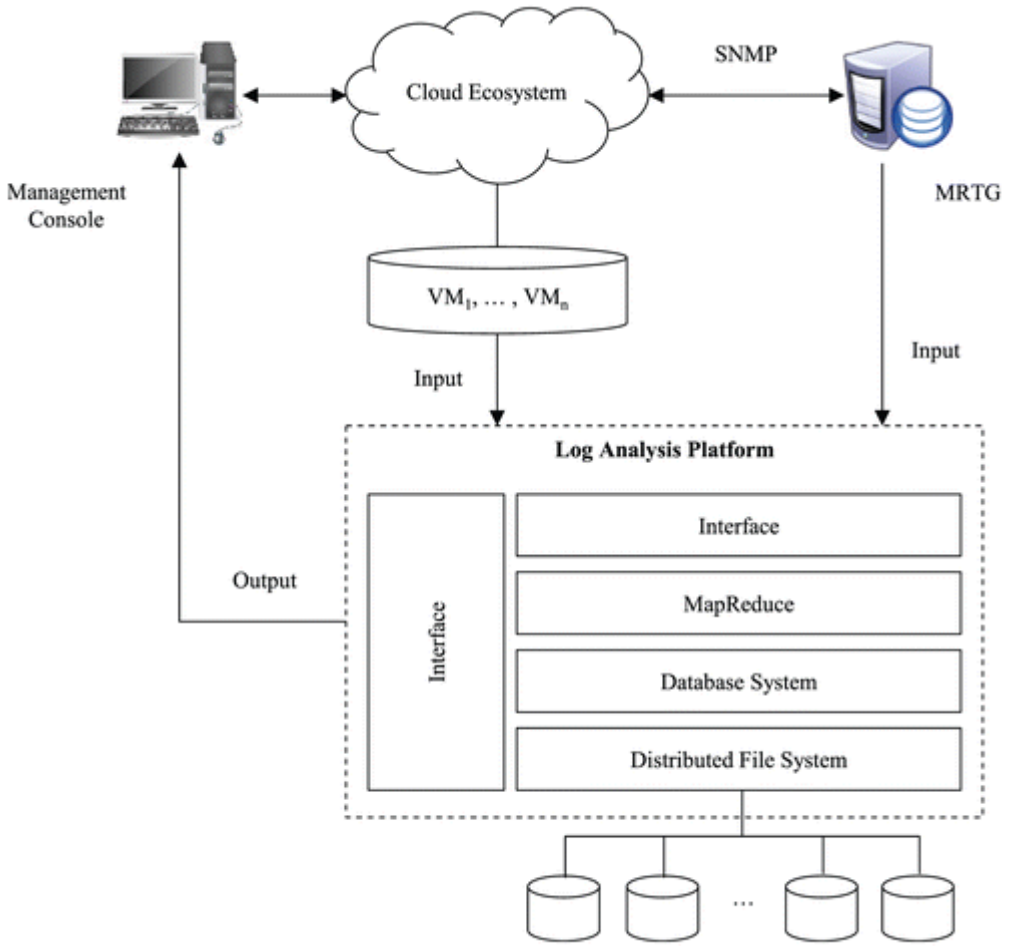



FIGURE 4.9 PROCESS OF REASSIGNING THE TOTAL LOAD TO THE INDIVIDUAL NODES

Load balancing is the process as shown in Figure 4.9 of reassigning the total load to the individual nodes of the objective system to make resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are overloaded while some other are under loaded. A load balancing algorithm which is dynamic in nature does not consider the previous state or behavior of the system, that is, it depends on the present behavior of the system is also depend on resource management done by Server. . The important thing to consider while developing such an algorithm are: estimation of load, comparison of load, stability of different system, performance of system, interaction between

nodes, nature of work to be transformed, etc. This load consideration can be in terms of CPU load, amount of memory used, delay or network load [7].

#### 4.1.1 STRATEGY AND MODEL BASED APPROACH

Cloud computing has been considered as a solution for solving enterprise application distribution and configuration challenges in the traditional software sales model. IaaS refers to a combination of hosting, hardware provisioning and basic services needed to a cloud. The uses of IaaS are as follows:

- Provides access to shared resources on need basis, without revealing details like location and hardware to clients.
- Provides details like server image on demand, storage queuing, and other information.
- Offers full control over server infrastructure, not limited specifically to applications, instances and containers.

The major issues that are commonly associated with SaaS in cloud systems are resource management, virtualization and multi-tenancy, data management etc. This proposal focuses on resource management due to the following fact that resource management for IaaS in cloud computing offers following benefits: scalability, Quality of Service (QoS), specialized environment, reduction in overheads and latency, improved throughput, cost effectiveness and simplified interface. At any time instant, resources are to be allocated to effectively handle workload fluctuations, while providing QoS guarantees to the end users. The computing and network resources are limited and have to be efficiently shared among the users in virtual manner. In order to perform effective resource management, we need to consider the issues such as resource mapping, resource provisioning, resource allocation and resource adaptation. The development of efficient service provisioning policies is among the major issues in cloud research. The issue here is to provide better quality of service in IaaS by provisioning the resources to the users or applications via load balancing mechanism, high availability mechanism, etc. To address the problem, we propose a model based method to predict and calculate the resource requirement of each virtual machine and using this model to design a load balancing framework in IaaS cloud. The contribution of this proposed task includes:

- A model that forecast the load and estimate the resource requirement of virtual machines in IaaS Cloud.

- A scalable framework for load-balancing which uses our resource requirement forecasting model.

Our IaaS cloud is composed mainly by three aspects:

- A cloud resources pool composed of virtual machine monitor cluster (VMMC) and shared storage server.
- A collector of virtual machines and hosts performance data.
- A controller which manages the virtual machines and physical hosts in VMMC based on our load-balancing strategies.

Forecasting the load of the virtual machines is a key issue in IaaS cloud in order to perform more efficient resource provisioning. To this end, we are developing an on-line load prediction technique based on Exponential Smoothing. Exponential Smoothing (ES) is a simple prediction method based on historical and current data that works very well in practice. ES is a procedure for continually revising a forecast in the light of more recent experience. There are several types of ES's. In this work a triple exponential smoothing is used, which is also named Holt-Winters. With this kind of ES the trend and seasonality of data are taken into account for the predictions.

In practice, we found that different types of virtual machines (different virtualization technologies) and heterogeneous physical (Inter or AMD) hosts will impact on the relationship between load of virtual machine and resource utilization of host. In this paper we use Canonical Correlation Analysis (CCA) to describe the relationship between the load of virtual machine and the resource utilization of host.

The load-balancing method is for two cases, one for VM starting, and the other for load exceeding or falling below the threshold. VMMC acts as the resource pool to host virtual machines. The master node collects performance data, establishes models and sends control instructions to VMM based on load-balancing strategy.

#### 4.1.2. SEED IDEA

The seed idea of proposal comes from the following IEEE papers referred:

In [1], a survey expresses the following concepts about resource management for SaaS:-

- One of the most pressing issues in cloud computing for IaaS is the resource management. Resource management problems include allocation, provisioning, requirement mapping, adaptation, discovery, brokering, estimation, and modelling.
- Resource management for SaaS in cloud computing offers following benefits: scalability, quality of service, optimal utility, reduced overheads, improved throughput, reduced latency, specialized environment, cost effectiveness and simplified interface.

- The development of efficient service provisioning policies is among the major issues in cloud research. The issue here is to provide better quality of service in IaaS by provisioning the resources to the users or applications via load balancing mechanism, high availability mechanism, etc.

In [2], the authors proposed a model based load-balancing method in SaaS cloud with the following concepts:

- Proper deployment of the virtual machine onto available hosts plays an important role on the load-balancing of modern Data Centre. At present, many load-balancing methods are based on the load forecasting model that could predict the resource requirements of the virtual machine.
- The resource requirement of virtual machine in SaaS Cloud is hard to predict because there will be variety of load types in IaaS cloud. Moreover, a variety of heterogeneous hardware environments and virtualization technologies make it hard to predict the requirement of virtual machine based on the workloads.
- To address the problem, we propose a model based method to predict and calculate the resource requirement of each virtual machine and using this model to design a load balancing framework in IaaS Cloud.

In [3], an HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud was proposed with the following concepts:-

- HTV dynamic load balancing model uses all of the above parameters.
- As it is dynamic in nature it overcomes disadvantages of round robin and throttled model.
- It gives maximum accuracy.

In [4], a “Trust and Reliability based Load Balancing Algorithm for Cloud SaaS” was proposed with the following concept:-

- Trust based model uses time parameters only.
- According to that trust values are calculated. So that VM can be categorized into trusted and un trusted VMs.
- It shows high efficiency when compared with other non-trust algorithms. In [5], the proposed Load balancing model uses static and dynamic parameters i.e. CPU, Memory respectively.

It initializes parameters and checks load degree for public cloud

TABLE 4.1 PARAMETERS USED IN DIFFERENT LOAD BALANCING MODELS

| Parameters | A Model Based Load-Balancing Method in IaaS Cloud | HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud | Trust and Reliability based Load Balancing Algorithm for Cloud IaaS | Load Balancing In Public Cloud |
|---|---|---|---|---|
| **CPU utilization** | Yes | Yes | No | Yes |
| **Memory utilization** | Yes | Yes | No | Yes |
| **Network I/O** | Yes | Yes | No | Yes |
| **Disk I/O** | Yes | Yes | No | No |
| **Initialization time** | No | No | Yes | No |
| **Machine instruction per second** | No | Yes | Yes | No |
| **Fault rate** | No | Yes | Yes | No |
| **Overload rejection** | No | Yes | No | No |

4.1.3 STATE OF THE ART

For resource management in cloud data center, previous work has focused on the problem of placing and replacing VMs in servers, in order to optimize resource management for different criteria, including performance, power and cost. There are some products and research focuses on virtualized resource management, such as Open QRM, VMware DRS. They dynamically allocate the CPU, memory and I/O resources to partitions virtual machines according to customer's requirement. A service-oriented priority-based resource scheduling scheme for virtualized utility computing propose a resource scheduling scheme to improve resource utilization and guarantee QoS of service. They Qos guarantee under the premise of improving resource utilization cloud environment. The QoS issue is also consider in work. Some work use the similar method used in the software to manage the virtual machines, minimized the number of physical machines using dynamic adaptation technique based on off-line analysis of application performance as a function of machine utilization.

The load forecasting techniques are widely used in the management of cloud resources, such as load balancing and resource scheduling. In a model-predictive controller is proposed to minimize the total power consumption of the servers in an enclosure subject to a given set of QoS constraints. A green scheduling algorithm integrated with neural network has also been

proposed for optimizing resources in cloud. Neural network prediction estimates the dynamic incoming load, and serves as input to a green scheduling algorithm for turning servers on and off. These decisions work to minimize the number of currently running servers. We should choose a suitable load forecasting method for the VMs in IaaS cloud environment.

### 4.1.4 NEED OF VIRTUALIZATION AND RESOURCE MANAGEMENT FOR LOAD BALANCING

The virtualization technology is the basis of the IaaS cloud. The existence of VM will impact the resource allocation because of the overhead for CPU, memory and I/O virtualization. Some research found the complex relationship between virtual machine and physical machine, but they mainly focus on the migration of the virtual machine, such as the work in Application performance in a virtualized environment. About virtual machine performance modelling, presented the VM performance modelling challenges and highlighted that visible and invisible resource interference cause a significant performance degradation and need to be considered when modelling VM performance. In their models, they considered not only CPU, Memory and I/O, but also shared cache, memory bandwidth and virtual machine monitor overhead. A model for describing the relationship between workload of VM and the resource of physical host can be created with the following specifications:

- System will be designed to fulfill requirements of load balancing in IaaS. IaaS is part of cloud in which lot of work has to be done in the field of load balancing.
- Load is considered in the form of CPU utilization, Memory and I/O usage in IaaS.
- System has ability of load forecasting & VM migration which helps in efficient load balancing.
- User Classes and characteristics: There are two classes of user of this system. First the user who will be any person wanted to use IaaS load balancing facility for efficient resource mapping. Second provider lease resources to customers. They are interested in maximizing profit and ensuring QoS for customers to enhance their reputation in marketplace.
- Design and Implementation Constraints: As the idea is based on cloud and we have to install applications on cloud, we have to use our own cloud where we can implement all proposed algorithms. We will implement our system on actual cloud after we will get successful results on cloudsim which is used to simulate cloud computing environment. that utilizes our proposed algorithms for resource mapping.

- Assumptions and Dependencies: The following events must occur for the proposal to be successful:
    - User can make request of resources at any time of day.
    - User has to make requests and rest will be handled by system.
    - Cloud provider can share their resources over the internet during resource scarcity.

#### 4.1.5 SYSTEM FEATURES

Functional Requirements:

- System should have valid process to validate users.
- System should have efficient load forecasting model.
- Forecasting must be accurate to state what will be load on particular server.
- System must have ability to take requests from user to fulfill it.
- System must have ability to allocate task to available servers so that none will remain ideal.
- Load balancer must have some threshold value which will help to decide what is capacity of specific server.
- System must have ability to migrate load to another server if threshold value of current server exceeds.
- System must have ability of resource mapping.
- System will map resources as per request from user.
- System must have to update information on runtime.
- System must provide efficient communication between datacenters and cloudlet.
- Must available resources to users in less time.

### 4.2 PROPOSED SYSTEM DESIGN

Our system consists of a virtual machine monitor (VMM) , CIS ( Cloud Info Services) and shared storage. Virtual machine monitor cluster acts as a resource pool on which virtual machines run. Shared storage is mainly used to store virtual machine disk images. The role of VMM is the infrastructure platform for SaaS cloud. The CIS is responsible for monitoring the resource usage of cloud pool, collecting historical load information of physical hosts and virtual machines, allocating and relocating virtual machines on appropriate hosts based on load-balancing strategies.

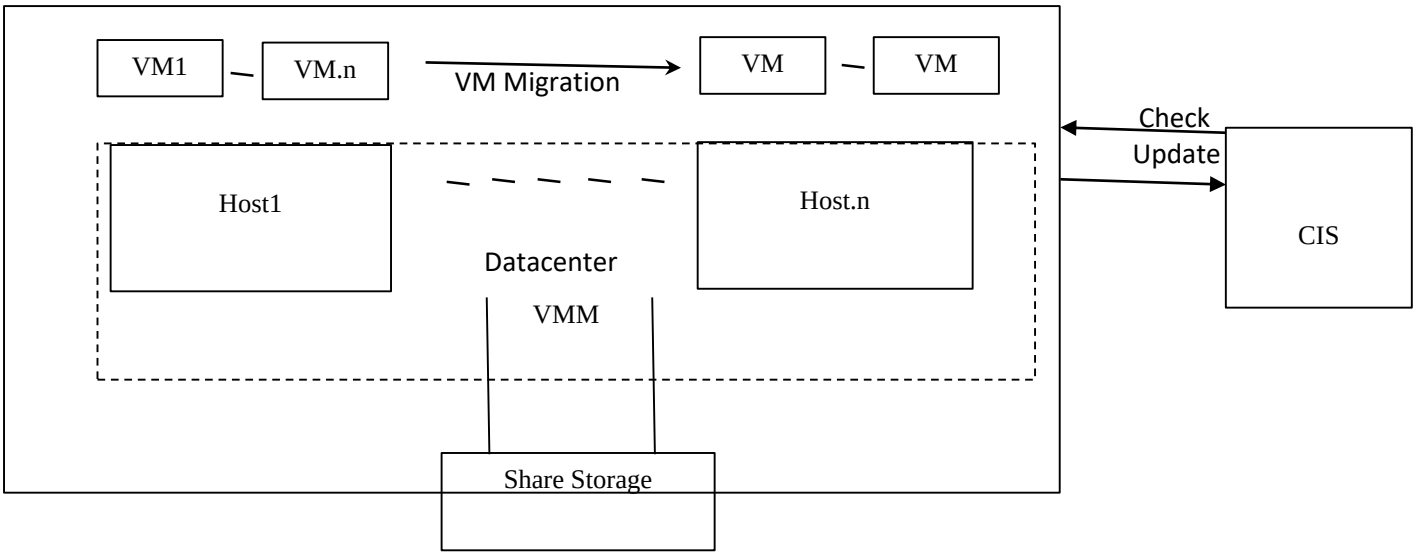



FIGURE 4.10 SYSTEM ARCHITECTURE

The resource usage of the pool is monitored by a CIS, which collects the performance data of the hosts and virtual machines. These data are used to analyze and predict the true resource requirement of each virtual machine, and to provide references for the load-balancing strategy. Virtual machines can run on any host in the resource pool. The CIS allocates the VMs on appropriate hosts, and migrate them between hosts based on load-balancing strategies.

The collector on CIS needs to collect the performance data of each virtual machine and host in VMM. The performance data include CPU utilization, memory usage, network and disk I/O throughput, etc. By online statistical analysis of the performance data, we can obtain the load characteristics of the virtual machine, as well as some cyclical variation. These characteristics of the virtual machine are used to predict the future load changes, and guide the estimation of resource requirement of virtual machines.

A controller on CIS needs to manage the resource of VMM (IaaS cloud), control virtual machines and physical hosts in the VMM resource pool. When a virtual machine starts, the CIS needs to allocate resources in the pool for this virtual machine, that means select a physical host in VMM to run this virtual machine. We can get the load information of virtual machines by load forecasting model. Next, the resource requirement for the physical hosts will be calculated based on these loads by using load-resource model. Therefore, we can select an appropriate host to run this virtual machine based on the resource requirement of the virtual machine and the recent load information of physical hosts. The principle for selecting the host is to make the whole system load balancing. Moreover, the controller monitor the resource usage of each physical host, if the resource usage of some host raise higher than other hosts, some virtual

machines would be relocated to other hosts through VM migration to maintain the entire system load-balancing.[1][8]

## 4.2 UML DIAGRAMS

4.3.1 USE CASR DIAGRAM

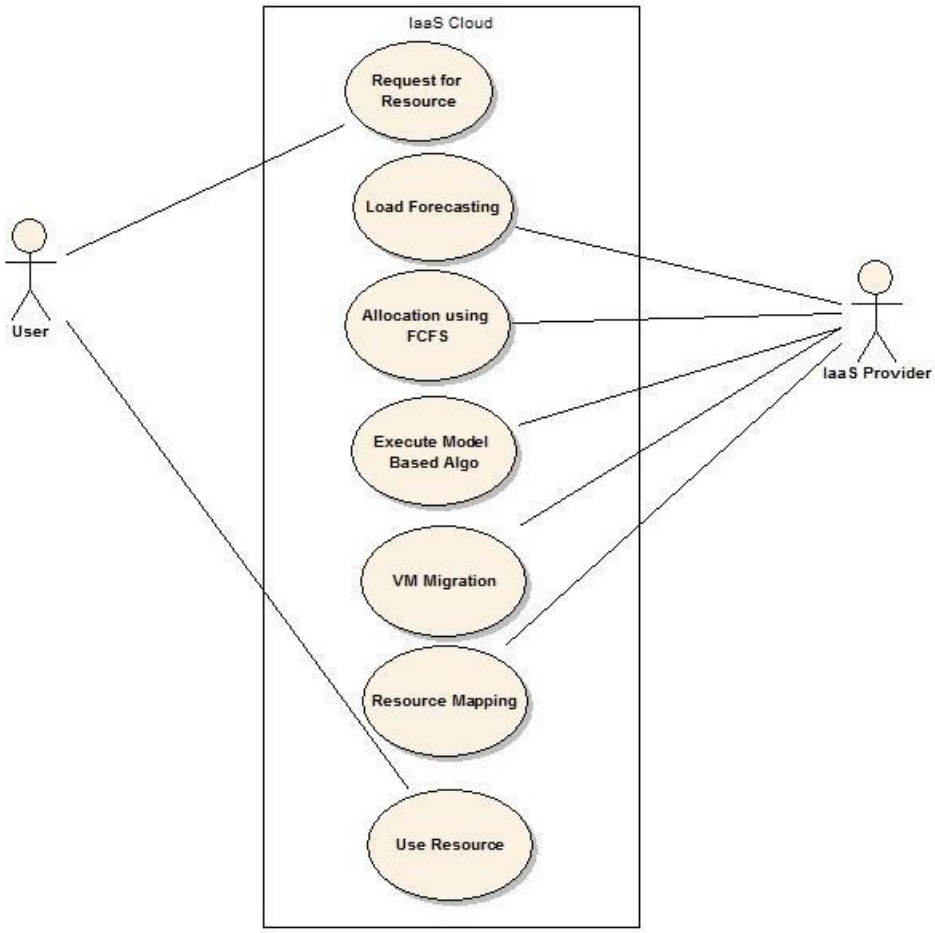



FIGURE 4.11 USE CASE DIAGRAM

4.3.2 ACTIVITY DIAGRAM

Recently, cloud computing emerged as the leading technology for delivering reliable, secure, fault-tolerant, sustainable, and scalable computational services, which are presented as Software, Infrastructure, or Platform as services (SaaS, IaaS, PaaS). Moreover, these services may be offered in private data centers (private clouds), may be commercially offered for clients (public clouds), or yet it is possible that both public and private clouds are combined in hybrid clouds. A suitable alternative is the utilization of simulations tools, which open the possibility of evaluating the hypothesis prior to software development in an environment where one can reproduce tests.

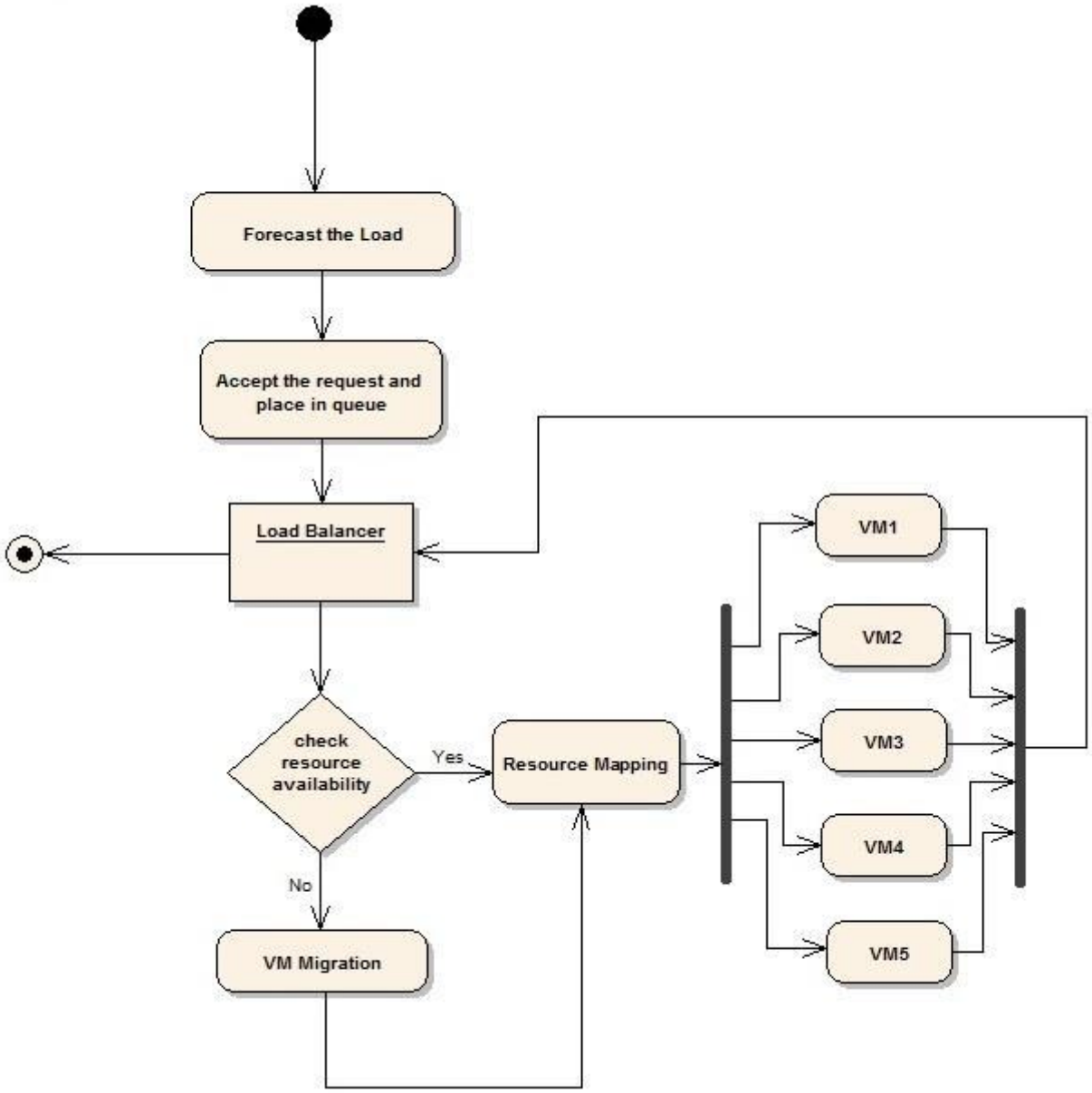



FIGURE 4.12 ACTIVITY DIAGRAM

Specifically, in the case of cloud computing, where access to the infrastructure incurs payments in real currency, simulation-based approaches offer significant benefits, as it allows Cloud customers to test their services in repeatable and controllable environment free of cost, and to tune the performance bottlenecks before deploying on real Clouds. At the provider side, simulation environments allow evaluation of different kinds of resource leasing scenarios under varying load and pricing distributions [9].

### 4.3.3 SEQUENCE DIAGRAM

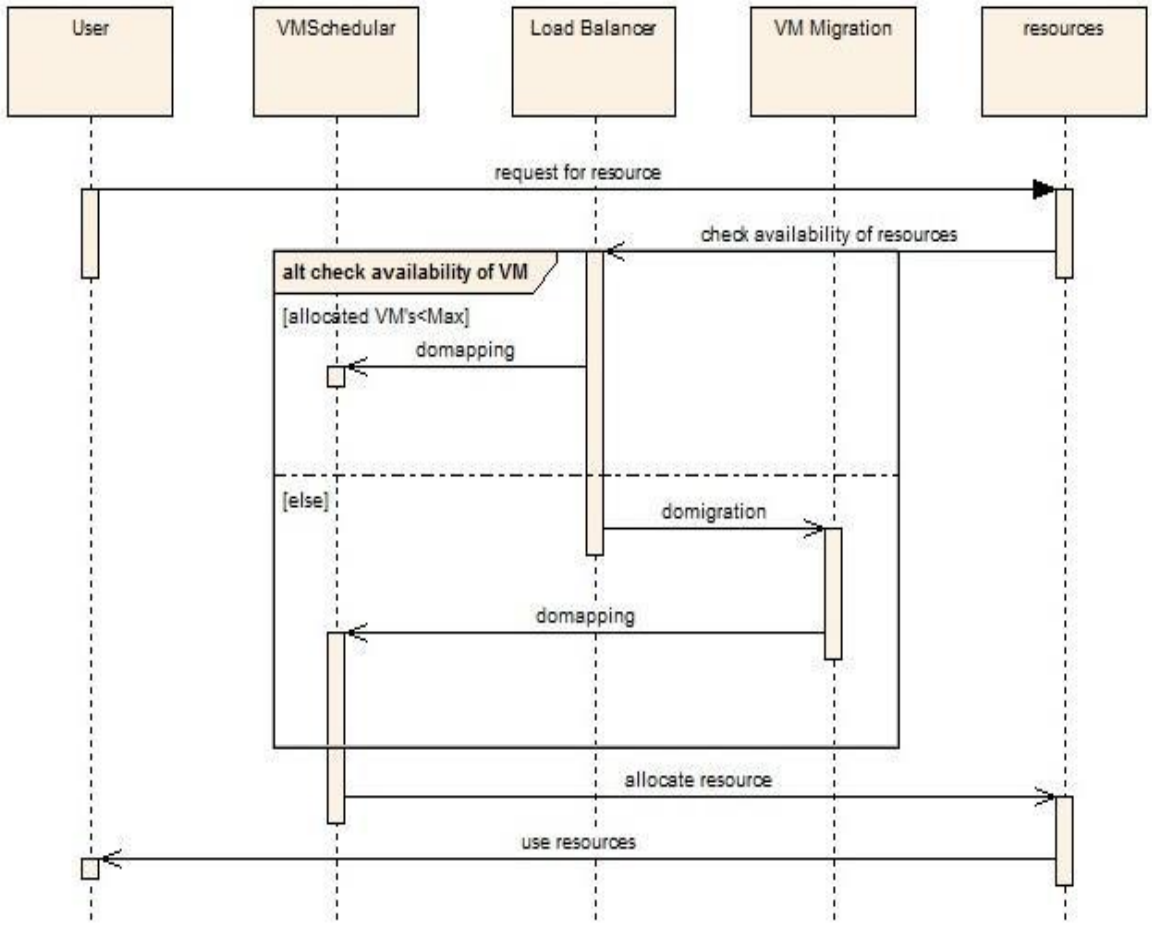



FIGURE 4.13 SEQUENCE DIAGRAM

### 4.3.4 COMPONENT DIAGRAM

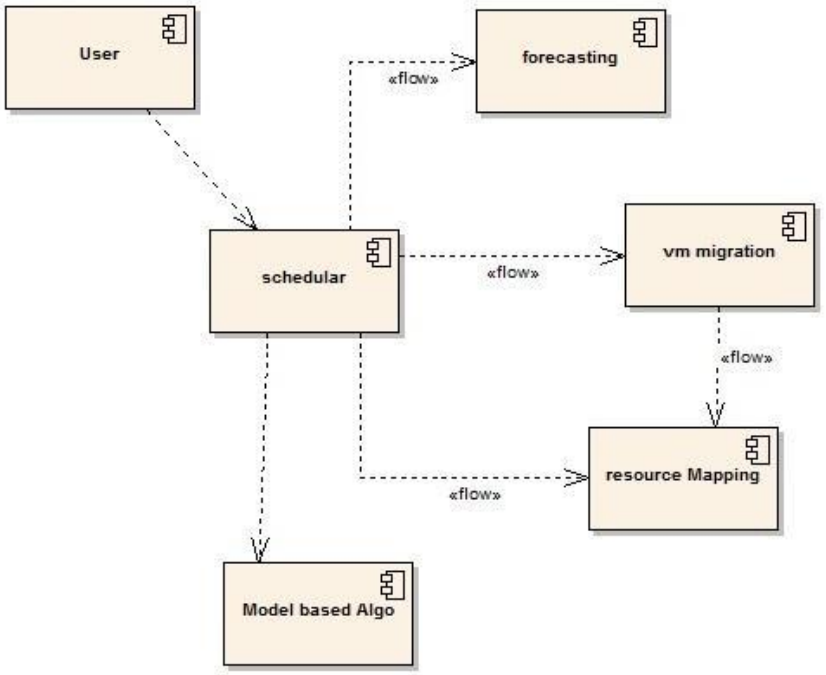



FIGURE 4.14 COMPONENT DIAGRAM

## 4.4. EVALUATION

The first test that we present here is aimed to analysis the efficiency of Cloudsim. Following Figure 4.15 represents the amount of time required to execute the cloudlets on VM. The VM were configured as a cloudletschedularSpaceShared and hosts are configured as a vmschedularTimeshared.

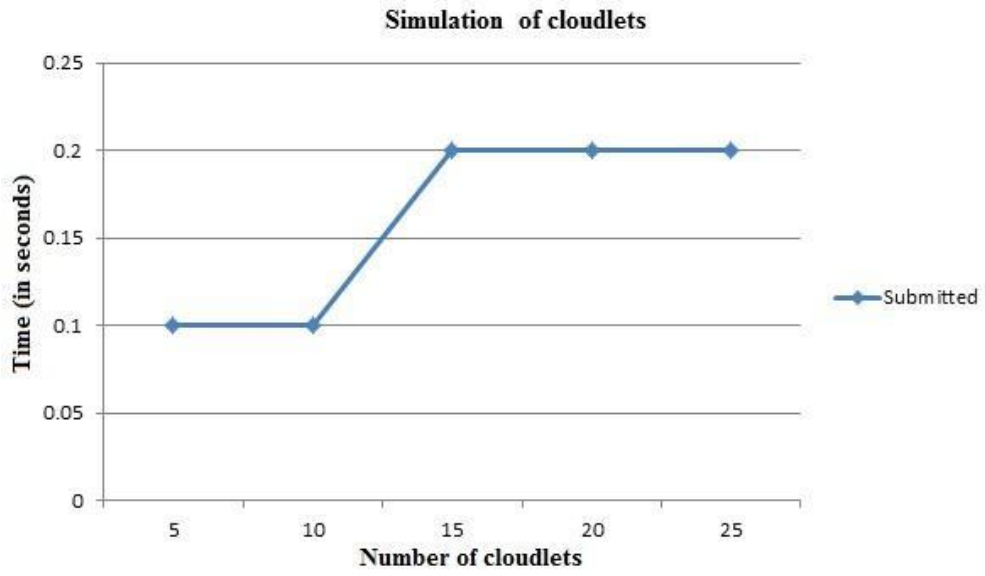



FIGURE 4.15 SIMULATION OF SPACE SHARED POLICY FOR CLOUDLETS

### 4.4.1 ADVANTAGES

- Large scale cloud computing at data centers.
- Virtualized server hosts with Cloudsim able policies.
- Support for modeling and simulation of virtualized server, hosts with Cloudsim able policies for provisioning host resources to VM's.
- Support for simulation of energy –aware computational resources.
- Support for modeling and simulation of data center network topologies and message passing applications.
- Support for dynamic insertion of simulation elements, and support for stopping and resuming simulation.
- Support for user defined policies to allot hosts to VMs, as well as allotting host resources to VMs.

### 4.4.2 DISADVANTAGES

Major Limitation is GUI, but instead of this it is used in universities and industry for implementation of cloud based algorithms.

## 4.5 SUMMARY

Resources are dynamic in nature so the stack of assets shifts with change in Configuration of cloud so the Load Balancing of the assignments in a cloud situation can fundamentally impact cloud's execution. A poor planning approach may leave numerous processors inert while a sharp one may devour an unduly extensive segment of the aggregate CPU cycles. In the current methodology we confront overhead issue of conveyed dispatching of undertaking to asset. In our proposed framework our fundamental objective of load adjusting is to give a dispersed, minimal effort, conspire that adjusts the load over every one of the processors.

To enhance the worldwide throughput of cloud assets, powerful and effective load adjusting calculations are on a very basic level vital. Different systems, calculations and strategies have been proposed, actualized also, characterized for executing Load adjusting in Cloud figuring condition.

## REFERENCES


[1] Domingues, P. ; Araujo, F. ; Silva, LEvaluating the performance And Intrusiveness of virtual machine for desktop grid computing. Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on DOI: 10.1109/IPDPS.2009.5161134 Publication Year: 2009, Page(s): 1- 8

[2] Zhenzhong Zhang, Limin Xiao, Yuan Tao, Ji Tian, Shouxin Wang, Hua Liu"A Model Based Load Balancing Method In IaaS Cloud ",2013 42ndinternational conference on parallel processing

[3] Jitendra Bhatia, Tirth Patel, Harshal Trivedi & Vishrut Majmudar2012, 'HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud', International Symposium on Cloud and Services Computing, pp.15-20.

[4] Punit Gupta, Mayank Kumar Goyal, Prakash Kumar " Trust and Reliability based Load Balancing Algorithm for Cloud IaaS " 2013 3rd IEEE International Advance Computing Conference (IACC)

[5] Shrikant M. Lanjewar, Susmit S. Surwade, Sachin P. Patil, Pratik S. Ghumatkar, Prof Y.B. Gurav "Load Balancing In Public Cloud" IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727Volume 16, Issue 1, Ver. VI (Feb. 2014)

[6] Borislav, D. ; Timcenko, V. ; Verica, V. ; Branka, A ; Performance comparison of three virtualization software used in windows 7. Telecommunications Forum (TELFOR), 2013 21st DOI: 10.1109/TELFOR.2013.6716405 Publication Year: 2013 , Page(s): 1019- 1022

[7] Liuyang WANG, Yangxin YU; Design Intelligence for multi agent for virtualization resource in cloud computing Computer Science & Service System (CSSS), 2012 International Conference on Date of Conference:11-13 Aug. 2012 Page(s):729 - Publisher: IEEE

[8] http://www.cubrid.org/blog/dev-platform/x86-server-virtualization-technology/

[9] http://www.virtuatopia.com/index.php/An_Overview_of_Virtualization_Techniques

[10] http://www.informit.com/articles/article.aspx?p=1627061

[11] http://www.c-sharpcorner.com/uploadfile/nipuntomar/system-Development-life-cycle-part-2/

[12] The complete Reference HTML and CSS fifth edition Author: Thomas A. Powell Publication McGraw-Hill

[13] The complete Reference JavaScript third edition Author: Thomas A. Powell and Fritz Schneider Publication McGraw-Hill

[14] The complete Reference JavaScript Ajax Author: Thomas A. Powell and Fritz Schneider Publation McGraw-Hill

[15] M. Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia , "A view of cloud computing", Communications of the ACM vol. 53 , 2010 , pp. 50-58, http://cacm.acm.org/magazines/2010/4/81493-a-view-of-cloud-computing/fulltext

[16] Suneel K S, Ravichandra A J , H S Guruprasad, "Enhanced Load Balancing Algorithm in Three-Tier Cloud Computing" , 2014, pp. 296-301. [3] D. Durkee, Why Cloud Computing Will Never Be Free , Volume 8 ,Issue 4, April 2010 , pp. 20-29. [4] Borko Furht , Armando Escalante Handbook of Cloud Computing, ISBN 978-1-4419-6523-3, Springer, 2010. [5] Ruixia Tong ,Xiongfeng Zhu, A Load Balancing Strategy Based on the Combination of Static and Dynamic, Database Technology and Applications (DBTA)2010, pp. 1-4.

# CHAPTER 5 ALGORITHMS WITH OPTIMIZATION

Considering the principle challenges, load balancing is featured as a noteworthy normal for Cloud computing which is still viewed as an open zone for look into. As included in Cloud scientific classification, load balancing can assume an imperative part in broad daylight, private, and cross breed Cloud. Along these lines planning an advanced load balancer either in framework or stage level can advance the Cloud execution or give clients.

Cloud computing has created as a strong space in the field of systems administration basically as a result of the limit of running an application or program in the meantime on various hubs those are related through a system. It incorporates asset sharing or computational information among the hubs. Scheduling is required for the proficient working of the cloud condition. To increase the capability of the work heap of cloud computing, scheduling is one of the errands performed to get most outrageous advantage. Cloud Computing utilized as a part of various sorts of mists with improved nature of administration

Today success of internet is largely to the vast amount of contents available at no cost to users. Measuring an internet traffic is the dominant service in today's internet. However, practically today's networking protocols and devices do not meet the need of the content related services. Recent services available on internet are very limited to those in which a connection is established based on key term i.e. IP addresses of the individual machine[1][7]. Many routing protocols like OSPF or BGP are capable of content aware routing packets based on IP addresses. But these protocols have no knowledge of which server (IP address is suitable for particular content. One portal site has more than one content server managed through a DNS ( Domain Name Server). DNS implementation returns IP addresses of multiple servers with same domain name in round robin scheme. DNS has no knowledge if these servers have different processing capacity or load. The network can routes reserve resources without the user or application level signaling.

### 5.1 CONTENT AWARE SERVICE MODEL

R. Metz suggested [11] in a wired magazine a content service model in a content driven internet consist of:-

- Content Aware Customer [CAC] :- A content request from client .
- Content Aware Service Provider [CASP]:- Is a mediator which offers different services by which content customer is able to access the content.

- Content Aware Provider [ CAP] . Actual creator or owner of the content.

Although cloud computing is a new technology that should have gained a lot of experience from the traditional computer technology, with content aware computing, will be a fragile system at its early beginnings.

**5.2 SERVICES OF CONTENT AWARE SERVICE PROVIDER**

Some common services of a CASP include :-

- Locating a content.
- Searching static contents to build a dynamic content.
- Charging information.
- Content negotiation.

Cloud computers face many security issues in content aware services is as follows :-

- Viruses: Every content aware service provider must use a strong antivirus and update it frequently, to avoid the collapse of the system and use a very restricted policy that each costumer's data must be checked before uploading it to the server.
- Hacking: The content service provider have to upgrade his system's security level to protect it from getting hacked. So, the users have to find a reliable content aware service provider like Yahoo, America On Line Info seek , to insure the safety and security of their data.

5.2.1 PRIVACY

Privacy is guaranteed when using cloud computing. As key loggers and phishing attacks cannot penetrate the cloud computing system because the user's computer does not have any available memory to store them. On the other side, some people may suggest that the cloud computing system may collect data about the users, the programs, the use and the sites they surf. Then the content aware service provider would sell this information to a third party company, which can use for advertisements and other purposes. This is not likely to happen because these service providers depend on their reputation as customers will not subscribe to bad reputational or unreliable Services Providers. Therefore, the company loses everything when it loses its reputation.

5.5.2 LEVEL OF CONTROL AND USAGE

The traditional computer has its own resources (hardware and software) so the user can control his own computer, i.e. he can open many of applications at the same time according to his needs. So, the users' level of usage is unlimited [12].

As for cloud computers, the level of control should be limited as the main concept of cloud computing is that all programs are found on one source (the cloud). Every program used from the cloud affects the whole system, overloading on a specific program may occur. Overloading may cause the failure of the system, so the level of usage should be limited by making the user predetermine the programs which will use on the cloud.

#### 5.2.3 CONNECTIVITY ISSUES

The user would have no access to the cloud, if the internet connectivity is lost. The user's data and applications would be inaccessible. The solution to this problem is very simple, internet outlets will be like electricity plugs in the near future as internet will be totally reliable. So, connectivity issues will not be a problem in the near future.

#### 5.2.4 SUBSCRIPTION ISSUES

- Some people may say that the service provider will have the right to delete the user's data, if he did not pay his subscription for six months. This way, the user would lose all what he had saved on the cloud. This problem can be solved by using the drink model where the user only pays for what he really uses, so the subscription prices will not be high [6].
- In fact, the cloud computing subscription will be cheaper than a telephone bill. In addition to that, if the user will not be able to pay his subscription for a couple of months but does not want to lose his data, he can suspend his subscription for a while until he is able to reactivate his account. This way the user's data will not be lost. (Hartman, March 2009)
- All of the previously mentioned problems are solved by using a transition period where the user can shift between the cloud computer and the traditional computer [11].

### 5.3 THE TRANSITION STATE

When computer designers shift from traditional computers to cloud computers, they have to make a smooth shift. This smooth shift will be done by making a transition state which will depend on a hybrid mode, where both the traditional and cloud computing systems will be installed the user's computer. When the user turns on his computer, it will allow him to select between the two systems, similar to having two

operating systems, like Windows and Linux, on one computer, where the user can choose between them.

### 5.3.1 PROBLEM FACE NEAR FUTURE

In a cloud computing content aware based routing and content aware based security are two efficient event delivery issue in load balancing. As we aware in a cloud computing communication is client server type. So issues related with client server with content based are an important paradigm for asynchronous communication between entities in a distributed network [3].

### 5.3.2 ISSUES TO BE CONSIDERED IN CONTENT BASED ROUTING

- What should the interconnection topology apply for routing for requests look like'?
- How should events user used correctly and efficiently routed through the network to the interested subscribers'?

### 5.3.3 ISSUES TO BE CONSIDERED IN CONTENT BASED SECURITY

- Which algorithm will apply for request and reply?
- What are the different secured strategies apply for server load balance?

## 5.4 CONTENT BASED ROUTING

Existing event routing solutions can be largely categorized into two classes: the, filter-based approach and the multicast-based approach [4], [5], [6], [8],[9]. In the filter based approach; routing decisions are made via successive content-based filtering at all nodes from source to destination. The way matches the event with remote subscriptions from other servers; and then forwards it only to fixed directions that lead to matching subscriptions.

This approach can achieve high network efficiency; but at the cost of expensive subscription information management and high processing load at each servers.

In the multicast-based approach; a limited number of multicast group are computed before event transmission begins. For each event, the routing decision is made only once at the publisher; mapping the event into the single appropriate group. The event is then multicast to that group. Assuming IP multicast or application-level multicast [9], [10] support. Because only a limited number of multicast groups can be built; sensors into different interests may be clustered into same group. And events may be sent to uninterested servers as well.

## 5.5 PROPOPSED CONTENT AWARE MODELS

### 5.5.1 CONTENT AWARE CENTRALIZED MODEL

This model distributes the incoming requests to the backend servers in a round-robin way. An incoming content aware request can be received by any server on cloud. Each server will determine, which server is suitable in terms of load and content type. This requires those all servers communicate with each other through a protocol to learn the status of other servers. The advantage of this scheme is that there is no bottleneck front end as every server share the incoming load.

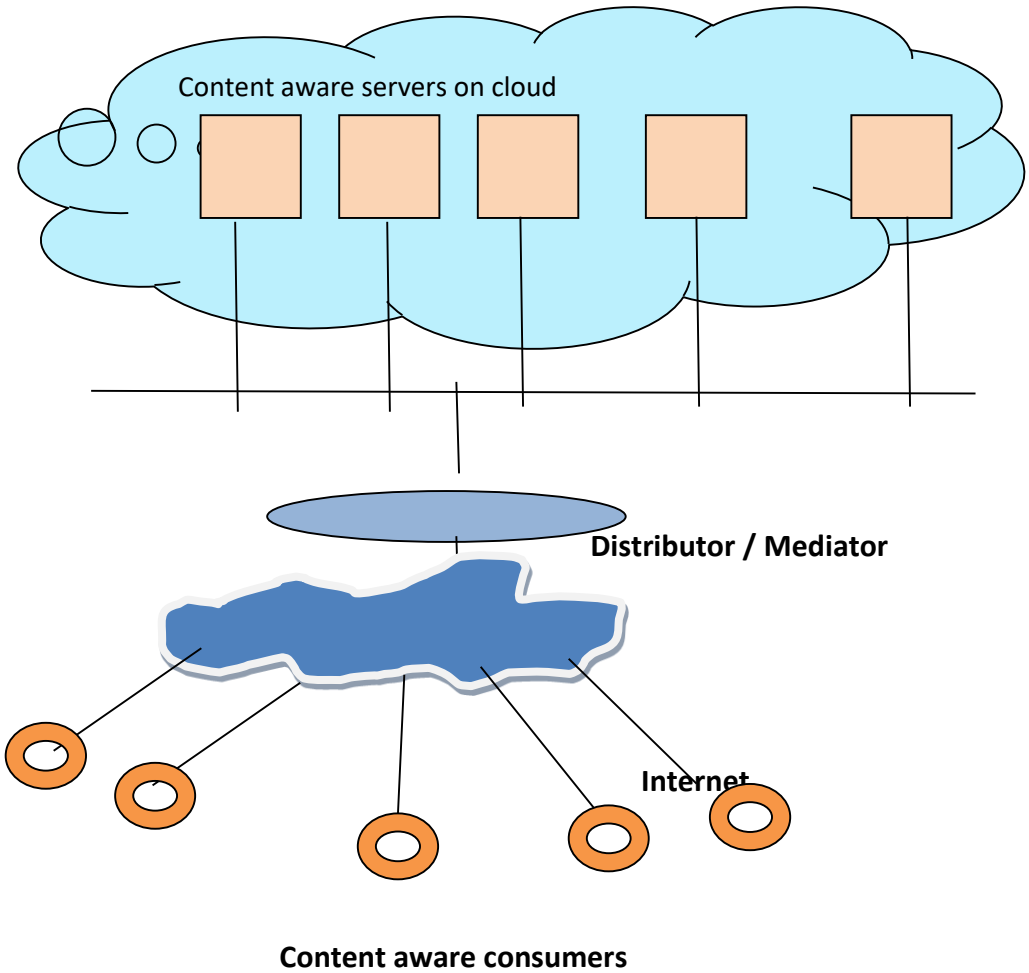



FIGURE 5.1 CONTENT AWARE CENTRALIZED MODEL

### 5.5.2 CONTENT AWARE DISTRIBUTION MODEL

A new protocol is required between servers, which might increase traffic load. Servers should be arranged based on content type such as audio and video server to be a true content aware networking. Content distribution refers to distributing content from one server to

multiple servers (mirror sites) located at different locations. It is becoming a common practice that, content is pushed or stored close to the customers, which means that content servers are distributed at the edges of network. One problem is how to serve a static content to a group of users whose requests are received at different times at a content server.

- Time slot multicast: All the users whose requests, for the same content, are received in the same time slot would be referred to as simultaneous users.
- Once a group of users is identified then the server can create a multicast group and instruct them to join a new multicast group.
- The advantage of this method is that the servers do not worry about packet order and clients can start the play sequence immediately since they always receive the content from the beginning.

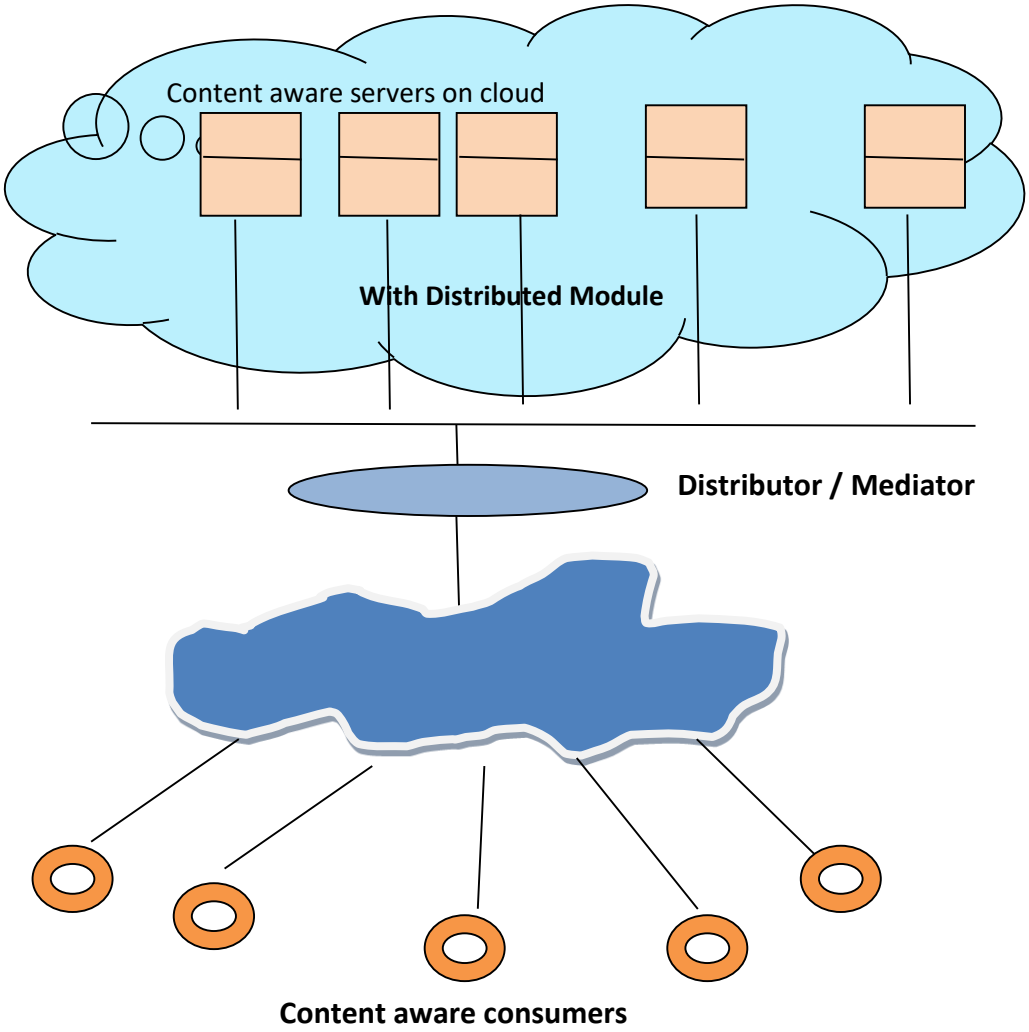



FIGURE 5.2 CONTENT AWARE DISTRIBUTION MODEL

By the definition of time slot, it is clear that longer time slots result in longer wait time for the users and shorter time slots take less advantage of the multicasting mechanism by having fewer requests in shorter time slots. Therefore, an optimum time

slot size must be selected based on these two non-orthogonal parameters, user wait time and multicast efficiency. The multicast delivery mechanism would be beneficial if the minimum expected number of requests exceed a certain threshold in a time slot. It is easy to see that the use of multicast becomes more likely as the requested content becomes more popular. Recently, reliable multicast has been proposed to distribute contents especially between content servers with in the realm of a web portal.

As the content servers are distributed and moved closer to the edges of a network in order to reduce the latency, we see that reliable multicast is being used to push the content from one server to all other servers. As the content servers are distributed and moved closer to the edges of a network in order to reduce the latency, we see that reliable multicast is being used to push the content from one server to all other servers.

### 5.5.3 LOCATION AWARE CONTENT SERVICES

Content routing pertains to directing the request to the most appropriate server with intelligence closer to the client.

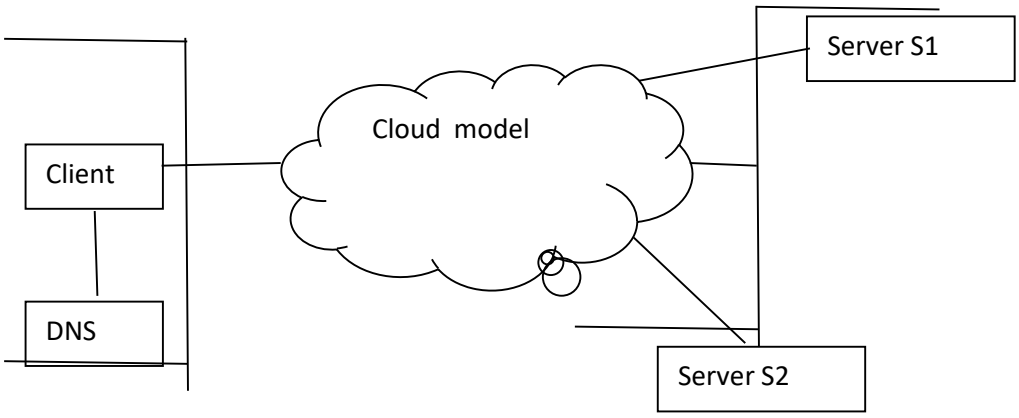



FIGURE 5.3 GENERAL CLOUD CONNECTION MODEL

The server distribution may be non-overlapping or it may be complete duplication (often referred to as mirroring). It should be noted that s1 and s2 are machines with two different IP addresses and are located at two different geographical locations. Current implementation of DNS points to a group of IP addresses for a single fully qualified hostname, and a name resolution request from the clients to the DNS is returned with the IP address of one the servers in a round-robin fashion.

This scheme is obviously inefficient for two reasons. Firstly, the requested content may not lie on the server whose IP address is returned by the DNS to the client for non-duplicated content. Secondly, the connection to the server, whose IP address is returned by the DNS, might be much slower than some other server in case of fully duplicated content.

The use of clients location is also helpful for content routing, such that the interface to a content aware DNS respond with the IP address of one of the distributed servers which is closest to the client. In fact, location aware services are more useful when a CASP want to push contents to customers. This allows CASP to provide up to date information about a customer's environment such as a sale in a nearby shopping Mall or traffic report as the user enters a traffic zone. One of the problems with providing location aware content services is due to lack of association of IP addresses with the physical location.

By making use of GPS or some other device and this location information is transmitted to the CASP. For non-dialup connections, the location of default gateway is usually fixed and can be transmitted as approximate location of the client, which is assumed to be close to the default gateway. This certainly requires quite a bit of cooperation among already deployed protocols

CSP then processes the location information along with the requested content and contacts the content aware provider (CAP) to retrieve the desired content and finally delivers it to the client, or redirects the client to the desired content.

## 5.6 KEY ISSUES

Cloud control over content aware traffic has the following issues.

- Disaster recovery:
    - Provide an extra level of High Availability to important applications.
    - Direct requests based on availability or health of the application.
- Active RTT:
    - Send client connections to the fastest responding data center.
    - Based on ping or DNS response time.
- Geo-location:
    - Send client connections to the "closest" physical data center.

- Send client connections to servers offering a specific language.

- Session capacity:
  - Send client connections to the data center with the most available remaining capacity.
- Weighted values:
  - Send client connections to the data center that has the highest combined score.
  - Send client connections to the data center with the most available active servers.
- Bandwidth Cost:
  - Query the bandwidth utilization of each site.
  - Select the site(s) whose bandwidth utilization has not exceeded a configured threshold during the most recent query interval.

## 5.7 CONTENT AWARE LOAD BALANCING PARAMETERS

### 5.7.1 CLUSTER PARAMETERS

- **IP address** :- This parameter specifies the cluster's primary IP address in standard Internet dotted notation (for example, w.x.y.z). The address is a virtual IP address and must be set identically for all hosts in the cluster. This IP address is used to address the cluster as a whole, and it should be the IP address that maps to the full Internet name that you specify for the cluster. This should be the primary IP address and subnet mask for the cluster. If you want to add multiple IP addresses to the cluster, you enter the additional IP addresses in the TCP/IP properties dialog box or in the Cluster IP Addresses dialog box in Network Load Balancing Manager.
- **Subnet Mask :-** This parameter denotes the subnet mask for the IP address specified. The mask is entered in standard Internet dotted notation (for example, 255.255.255.0).
- **Network address:**- This parameter specifies the network address (Medium Access Control [MAC] address) for the network adapter to be used for handling client-to-cluster traffic.

If multicast support is disabled (causing the host to revert to unicast mode), Network Load Balancing automatically instructs the driver belonging to the cluster adapter to override the adapter's unique, built-in network address and to change its MAC address to the cluster's MAC address. This is the address used on all cluster hosts. You do not need to manually configure the network adapter to recognize this address.

#### 5.7.2 CLUSTER OPERATION MODEL

These parameters specify whether or not a multicast MAC address should be used for cluster operations. If multicast is enabled, Network Load Balancing converts the cluster MAC address belonging to the cluster adapter into a multicast address. It also ensures that the cluster's primary IP address resolves to this multicast address as part of the ARP protocol. At the same time, the adapter can now use its original, built-in MAC address that, in unicast mode, was disabled.

## 5.8 HOST PARAMETERS

#### 5.8.1 INTERFACE

This parameter only appears when using Network Load Balancing Manager and is configured when you add the host to the cluster. The parameter specifies the host's network adapter that will use network load balancing in context of the current cluster.

#### 5.8.2 IP ADDRESS

This parameter specifies this host's unique IP address used for network traffic not associated with the cluster (for example, Telnet access to a specific host within the cluster). It should be entered in standard Internet dotted notation (for example, w.x.y.z). This IP address is used to individually address each host in the cluster and hence should be unique for each host. The dedicated IP address should always be entered first in TCP/IP properties.

Network Load Balancing references the dedicated IP address only when a single network adapter is used to handle both client-to-cluster traffic and other network traffic that must go specifically to the dedicated IP address. Network Load Balancing ensures that all traffic to the dedicated IP address is unaffected by the Network Load Balancing current configuration, including:

- When this host is running as part of the cluster.

- When Network Load Balancing is disabled due to parameter errors in the registry.

5.8.3 PORT RULES

To maximize control of various types of TCP/IP traffic, you can set up port rules to control how each port's cluster network traffic is handled. The method by which a port's network traffic is handled is called its filtering mode. There are three possible filtering modes: Multiple hosts, Single host, and Disabled.

We can also specify that a filtering mode apply to a numerical range of ports. Each rule consists of the following configuration parameters:

- The virtual IP address that the rule should be applied.
- The TCP or UDP port range for which this rule should be applied.
- The protocols for which this rule should apply, including TCP, UDP, or both.
- The filtering mode that specifies how the cluster handles traffic described by the port range and protocols.

**5.9 EXISTING CLOUD AND ACCESS TECHNOLOGIES**

Identity access management of cloud has following questions :-

- Who has access to what?
  - What can they do with that access?
  - What can they do with the information they obtained?
  - And, what did they do?

In order to answer these questions, you need to be able to effectively manage and control three areas, as follows:

• Control identities Manage user identities and their roles, provision users for access to resources, maintain compliance with identity and access policies, and monitor user and compliance activity.

• Control access Enforce policies relating to access to systems, web applications, and information. Provide management of privileged users to avoid improper administrator actions. Provide flexible, strong, and risk-based authentication capabilities so as to help identify and prevent fraudulent activities.

• Control information Discover and classify sensitive corporate information, and enforce usage policies relating to this information.

## 5.10 PROPOSED CLOUD LAYERS AND ACCESS TECHNOLOGY

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cloud provides a dynamically scalable, abstracted computing and storage infrastructure that is typically based on a virtualized, distributed, fault tolerant, parallel computing architecture. Cloud computing refers to computation, software, data access, and storage services that do not require any end-user knowledge of the physical location and configuration of the system that delivers the services. Depending on the type of resources provided by the Cloud, distinct layers can be defined in below Figure 5.4. The bottom-most layer provides basic infrastructure components such as CPUs, memory, storage, henceforth it often denoted as Infrastructure-as a-Service (IaaS). Amazon's Elastic Compute Cloud (EC2) is a prominent example for an IaaS offer.

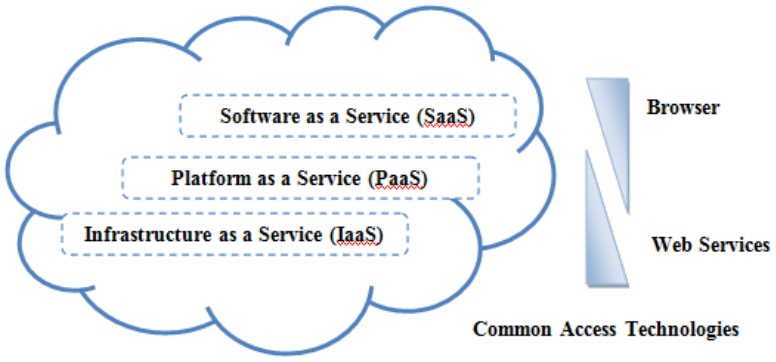



FIGURE 5.4 CLOUD DISTINCT LAYERS

On the top of IaaS, more platform-oriented services are present which allow the usage of hosting environments tailored as per specific need. Google App Engine is an example for a Web platform as a service (PaaS) which enables to deploy and dynamically scale Java based Web applications. Finally, the top-most layer provides its users with ready to use applications also known as Software- as-a-Service (SaaS). To access these Cloud services, two main technologies can be currently identified. Web Services are commonly used to provide access to IaaS services and Web browsers are used to access SaaS applications. In PaaS environments both approaches can be found.

## 5.11 EXISTING FORMULA FOR WEB SWITCH BASED ON CLASSICAL POLICIES [14]

With reference to above papers it was reflected that policies are based on QoS parameters like mainly Web traffic. They also mentioned the first step to construct a dispatching algorithm is to decide when to monitor the system. It is possible to do it each time a request arrives to the front end of the web system and then to decide the target server that will attend that request as it is done in [8]. It is also possible to define the distribution of the load among servers at fixed times by using static time slot scheduling.

The decision of using a content-aware load balancer depends on whether or not the content of the requests that are going to be balanced matters. For instance, if an e-commerce application is included in the web system, there will be some requests like the "buy confirmation" where bank account data is included that should considered a priority request and then be safely served. Hence, to design a complete general purpose content-aware load balancer it is necessary to include service differentiation.

In the existing service they divided the set of 'm' types of service previously defined in 's ' types of requests that can be dynamic and static. Then m can be equal to or greater than, depending on the nature of the applications running on the web servers. In the case each type of service is either static or dynamic, m would be equal to s. But in the case some service types can be both of them (static and dynamic) then the system will at most consider **m = 2s** different service types. Making this differentiation the scheme is completely customizable for all cases.

## 5.12 PROPOSED FORMULA BASED ON FUNCTIONING

In a given paper m= 2s where s types of request can be dynamic and static which does not contain following parameters like :-

$N = E \times U \times T < (1-B)$ ……(1)

Where

N – the number of servers to be virtualized on a single new physical server

B – embodied packet ratio

E – efficiency factor (energy consumption of a single new server with capacity equivalent to the original N servers divided by energy consumption of N original servers, assuming the same technology and utilization, for the projected life)

T – technology factor (energy consumption of new servers per unit CPU capacity divided by energy consumption of old servers per unit CPU capacity)

U = utilization factor (utilization of old servers divided by utilization of new server ) To pay back the cost of embodied energy and realize a net gain, from U1,U2,U3....Un

Out of that U contains different accesses included f1,f2,.... fg.

i.e. $\mu$ i (f1,f2.... fg) ……(2)

Where f1= First parameter like node speed

f2= Security parameter based on access technology etc.

where . $\mu$ i is a factors of request time 't' and no of user 'X'

i.e. $\mu$ i = λ(t,X) ……(3)

Where t = 0..... k-1

X = 1.....n users

To focus more λ(t,X1,X2.....Xn)

## 5.13 EXISTING ALOGORITHM WITH BASIC SECURITY GOALS

Content based routing are messages in a network is an essential component of Internet communication, as each packet in the Internet must be passed quickly through each network (or autonomous system) that it must traverse to go from its source to its destination. It should come as no surprise, then, that most methods currently deployed in the Internet for routing in a network are designed to forward packets along shortest paths. Indeed, current interior routing protocols, such as OSPF, RIP, and IEGP, are based on this premise, as are many exterior routing protocols, such as BGP and EGP .

The algorithms that form the basis of these protocols are not secure, however, and have even been compromised by routers that did not follow the respective protocols correctly. Fortunately, all network malfunctions resulting from faulty routers have to date been shown to be the result of misconfigured routers, not malicious attacks. Nevertheless, these failures show the feasibility of malicious router attacks, for they demonstrate that compromising a single router can undermine the performance of an entire network.

Basic security goals:

- Fault detection: The algorithm should run correctly and, in addition, should detect any computational steps that would compromise the correctness of the algorithm.
- Damage containment**:** The algorithm should contain the damage caused by an incorrect router to as small an area of the network as possible.

5.14 PROPOSED ALGORITHM FOR ADDITIONAL SECURITY GOALS

- Authentication: The algorithm should confirm that each message is sent from the host or router that the message identifies as its source.
- Data integrity: The algorithm should confirm that the contents of received messages are the same as when they are sent, and that all components of a message are as intended by the algorithm (even those message portions added by routers other than the original sender).
- Timeliness**:** The algorithm should confirm that all messages interacting to perform this algorithm are current up-to-date messages, thereby preventing replay attacks.

**5.15 SUMMARY**

Load balance is an important factor in cloud communication. As any overloaded server or network link may degrade the total by system performance. We mainly focus on balancing node between communication where stress on each servers are difficult . Because link stress is affected by network resource dimensioning and provisioning strategies, it left as future work.

**REFERENCES**


[1] Sarwar, B.: Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering. In: 5th International Conference on Computer and Information Technology (2002)

[2] Kużelewska, U.: Advantages of Information Granulation in Clustering Algorithms. In: Filipe, J., Fred, A. (eds.) ICAART 2011. CCIS, vol. 271, pp. 131–145. Springer, Heidelberg (2013)

[3] Pitsilis, G., Zhang, X., Wang, W.: Clustering Recommenders in Collaborative Filtering Using Explicit Trust Information. In: Wakeman, I., Gudes, E., Jensen, C.D., Crampton, J. (eds.) Trust Management V. IFIP AICT, vol. 358, pp. 82–97. Springer, Heidelberg (2011)

[4] L. Fiege. G. Miihl. F. Girtner. "A Modular Approach to Building Event-Based Svstems". In ACM Symposium on Applied Computing, 2002

[5] Y. Wang, L. Qiu: D.. Achlioptas. G. DR. P: Larson. and .H. Wang. "Subscription Partitioning and Routine in Content-based Publish Subscnibe Networks." In 16th International Svmpoaiuni on Distributed Committing. 2002

[6] J. Pereira. F. Fabret. F. Llirbat and D. Shasha. "Efficient ‘matching for web-based subscribe systems, ' In Proc. of the FIfih IFCI! 2000.

[7] O. Kharif. Why web publishers fear a little sharing. BusinessWeek Online. October 19 2004 http://www.businessweek.com/bwdaily/dnflash/oct2004/ nf20041019_9800_db016.htm.

[8] B. Krishnamurthy and C. E. Wills. Improving Web Performance by Client Characterization Driven Server Adaptation. In Proceedings of the WWW, May 2002.

[10] R. Metz. We don’t need no stinkin’ login. WIRED Magazine, July 2004 http://www.wired.com/news/print/0,1294,64270,00.html.

[11] www.communication.howstuffworks.com/cloud-computing.htm

[12] www.scs.org/magazines/2010-10/index_file/Files/Fujimoto.pdf

[13] Katja Gilly,Nigel Thoomas,Carlos Juiz, ”Scalables Qos content aware load balancing algorithm for web switch based on classical polocies” in 22nd International Conference on Advanced Information Networking and Applications,2008 IEEE

# CHAPTER 6 CLOUD SECURITY

## 6.1 INTRODUCTION

In cloud computing it is sometimes necessary for users to share the power to use a cryptosystem. The system secret is divided up into shares and securely stored by the entities forming the distributed cryptosystem. The main advantage of a distributed cryptosystem is that the secret is never computed, reconstructed, or stored in a single location, making the secret more difficult to compromise. In many applications, a threshold (t) or more share holders are required to cooperatively generate a digital signature, in contrast to the conventional single signer. This may also be seen as a distribution of trust since the shareholders must collaborate and contribute equally to produce a valid multiparty signature.

Secure cloud computing schemes combine the properties of threshold group-oriented signature schemes and multisignature schemes. Threshold multisignature schemes are also referred to as threshold signature schemes with traceability. The combined properties guarantee the signature verifier that at least 't' members participated in the generation of the group-oriented signature and that the identities of the signers can be easily established. The majority of the existing secure cloud computing scheme belong to variants of the single signatory, generalized ElGamal signatures extended to a group/multiparty setting.

Secure cloud computing schemes can be differentiated from threshold group signatures by the fact that by definition, the individual signers remain anonymous since it is computationally hard to derive the identities from the group signature, with the exception of the group managers. In contrast, by the above-defined trace ability property of secure cloud computing schemes, the individual signers are publicly traceable and do not enjoy anonymity. Consequently, the traceability property of secure cloud computing schemes allows the individual signers to be held accountable in the public domain and renders the unlink ability property of threshold group signature schemes, as defined in, inapplicable.

## 6.2 PURPOSE

Cloud computing has been envisioned as the next-generation architecture of IT enterprise, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk. As a disruptive technology with profound Implications, cloud computing is transforming the very nature of how businesses use information

technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced into the Cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely into the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances. While these advantages of using clouds are unarguable, due to the opaqueness of the Cloud—as separate administrative entities, the internal operation details of cloud service providers (CSP) may not be known by cloud users—data outsourcing is also relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outrages and security breaches of noteworthy cloud services appear from time to time. Secondly, for the benefits of their own, there do exist various motivations for cloud service providers to behave unfaithfully towards the cloud users regarding the status of their outsourced data. Examples include cloud service providers, for monetary reasons, reclaiming storage by discarding data that has not been or is rarely accessed, or even hiding data loss incidents so as to maintain a reputation. In short, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, it does not offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the successful deployment of the cloud architecture.

**6.3 SCOPE**

As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. Thus, how to efficiently verify the correctness of outsourced cloud data without the local copy of data files becomes a big challenge for data storage security in Cloud Computing. Note that simply downloading the data for its integrity verification is not a practical solution due to the expensiveness in I/O cost and transmitting the file across the network. Besides, it is often insufficient to detect the data corruption when accessing the data, as it might be too late for recovering the data loss or damage. Considering the large size of the outsourced data and the user's constrained resource capability, the ability to audit the correctness of the data in a cloud environment can be formidable and expensive for the cloud users. Therefore, to fully ensure the data security and save the cloud users' computation resources, it is of critical importance to

enable public auditability for cloud data storage so that the users may resort to a third party auditor (TPA), who has expertise and capabilities that the users do not, to audit the outsourced data when needed. Based on the audit result, TPA could release an audit report, which would not only help users to evaluate the risk of their subscribed cloud data services, but also be beneficial for the cloud service provider to improve their cloud based service platform. In a word, enabling public risk auditing protocols will play an important role for this nascent cloud economy to become fully established, where users will need ways to assess risk and gain trust in Cloud.

Recently, the notion of public auditability has been proposed in the context of ensuring remotely stored data integrity under different systems and security models. Public auditability allows an external party, in addition to the user himself, to verify the correctness of remotely stored data. However, most of these schemes do not support the privacy protection of users' data against external auditors, i.e., they may potentially reveal user data information to the auditors. This drawback greatly affects the security of these protocols in Cloud Computing. From the perspective of protecting data privacy, the users, who own the data and rely on TPA just for the storage security of their data, do not want this auditing process introducing new vulnerabilities of unauthorized information leakage towards their data security. Moreover, there are legal regulations, such as the US Health Insurance Portability and Accountability Act (HIPAA), further demanding the outsourced data not to be leaked to external parties.

Exploiting data encryption before outsourcing is one way to mitigate this privacy concern, but it is only complementary to the privacy-preserving public auditing scheme to be proposed in this paper. Without a properly designed auditing protocol, encryption itself cannot prevent data from "flowing away" towards external parties during the auditing process.

Thus, it does not completely solve the problem of protecting data privacy but just reduces it to the one of managing the encryption keys. Unauthorized data leakage still remains a problem due to the potential exposure of encryption keys. Therefore, how to enable a privacy-preserving third-party auditing protocol, independent to data encryption, is the problem which has been tackled in the paper we have referred to. The paper considered is among the first few ones to support privacy-preserving public auditing in Cloud Computing, with a focus on data storage. Besides, with the prevalence of Cloud Computing, a foreseeable increase of auditing tasks from different users may be delegated to TPA. As the individual auditing of these growing tasks can be tedious and cumbersome, a natural demand is then how to enable TPA to efficiently perform the multiple auditing tasks in a batch manner, i.e., simultaneously.

To address these problems, this work utilizes the technique of public key based homomorphic authenticator, which enables TPA to perform the auditing without demanding the local copy of data and thus drastically reduces the communication and computation overhead as compared to the straightforward data auditing approaches. By integrating the homomorphic authenticator with random mask technique, the protocol guarantees that TPA could not learn any knowledge about the data content stored in the cloud server during the efficient auditing process.

In the ongoing occasions, Cloud Storage has turned into an essential development propensity as to Information Innovation. All things considered, information security has turned a vital issue that hinders it regarding business applications like information honesty, accessibility and secrecy. Redistributing of capacity is one rising pattern that happens to provoke a few interesting security issues, various which that have as of now been explored broadly amid the past.

### 6.4 MOTIVATION

To solve the problem of data integrity verification ,many schemes have been proposed under various models and systems. all the schemes of verification mainly fall into two categories private auditing and public auditing[10].The private auditing scheme have achieved efficient auditing as data is private and restrict access to particular limited users. However in public auditing not only the client/owner of data but also any one can challenge the integrity state of data. This scheme introduces an entity i.e. Third Party Auditor (TPA) ,which relaxes the user.

The aggregation and algebraic properties of the authenticator further benefit the design for the batch auditing. Specifically, contribution in this work can be summarized as the following three aspects:

- Motivate the public auditing system of data storage security in Cloud Computing and provide a privacy-preserving auditing protocol, i.e., the scheme supports an external auditor to audit user's outsourced data in the cloud without learning knowledge on the data content.
- Apparently, this scheme is the first to support scalable and efficient public auditing in the Cloud Computing. In particular, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.

- To prove the security and justify the performance of proposed schemes through concrete experiments and comparisons with the state-of-the-art. from performing timely audit. As users may have busy hectic schedule, so he may not get time to perform timely audit, the TPA will perform audit ,on behalf of user and notify user if any one tries to steal their data or make changes to their data[10] .
- Another challenge in cloud security is supporting data dynamics i.e. when a user access remote data, he/she may wish to update their data through operations like block insertion, block modifications, deletions etc unfortunately till now this aspect is not been achieved in previous work, which mainly focuses on static data.it is very important to support data dynamics in order to achieve proper outsourcing of data.

In short this scheme proposes:

Public auditing of data storage in cloud computing security and propose a scheme to enable data dynamism which was absent in previous schemes. Batch auditing , where TPA perform various auditing tasks simultaneously. It also support dynamic operations on data to make sure that changes made by such operations should be reflected to the data stored on cloud properly to maintain data integrity of data stored on cloud.

**6.5 RELATED WORK**

Prior to this scheme, a lot of work has been done for remotely stored data integrity verification. In existing work the first to consider public auditing in their scheme 'Provable Data Possession' (PDP) for safeguarding the possession of files stored remotely. In their work they have applied the concept of RSA-based homomorphic tags for auditing remotely stored data. But in [1] author did not consider dynamic data support. Later in their subsequent work [2], the authors extended the concept of PDP to propose a dynamic form of PDP. This scheme provides efficient and secure provable data possession with less computation overhead. It supports secure and limited dynamic operations on the data. The drawback of this scheme is that it supports only a prefixed number of verifications.

Further, it supports only a limited number of basic block level dynamic operations and does not support block insertion operation. In [3], considered the dynamic data support in a distributed environment. The authors applied Erasure correcting codes, to support the dynamic operations for the data update, delete and append. This scheme is

efficient against Byzantine failure, malicious attacks etc. Though this scheme supports dynamic data operations, it does not support insertion of data blocks in between the data. In [4], proposed a ”Proof of Retrievability” (PoR) scheme. The authors applied the concept of spot checking and error correcting codes, to guarantee both the ’Possession’ and the ’Retrievability’ of the files stored remotely. The authors used special purpose precomputed encrypted blocks known as ’sentinels’, which were included in a file randomly for detection purposes. However, the use of sentinels stops the detection of data dynamics updates. This scheme does not support public auditing properly [11]. In [5], proposed an improved PoR Scheme. The authors used the concept of homomorphic authenticators constructed based on BLS-signatures, to aggregate a security proof into a small authenticator value, and to achieve public verifiability. This scheme works only on static files. In [6], there were the first to consider ’Dynamic Provable Data Possession ’(DPDP), which is an extended version of the PDP scheme. They applied the concept of Authentication Directories with ranking schemes. To provision block insertion, the authors removed the index information in the ’tag’ calculation of the existing PDP scheme. This scheme enables efficient proof of the entire file system by providing verification at different levels for different users without downloading the whole data. Though this scheme focuses on both the audit and the dynamic operations of data, its efficiency is unclear. Though various schemes have been proposed for remote data integrity verification, the issue of supporting both public auditing and data dynamics has not been fully addressed. In [7], have proposed a scheme, which supports both public auditing and data dynamics.

They combined the concept of PDP and POR schemes. An external TPA for the verification of the user’s data was introduced to the scheme. This scheme is beneficial as it reduces the tasks of the user. The authors have used the concept of homomorphic authenticators and Merkle hash tree, to verify that the values and positions of the data blocks are unmodified. Our scheme uses the same concept of TPA but using zero knowledge proof and CRC-64 techniques rather than Elliptical Curve Cryptographic (ECC) techniques. Our scheme efficiently reduces the time required to perform the verification of the remotely stored data. Cloud computing storage depends on redistributing the registering activities to some outsider. This includes hazard dangers as to uprightness, accessibility and secrecy of data and administration. It is a fundamental issue persuading cloud clients about their information being kept up unblemished particularly as they

typically don't use to store that data locally. For typical cases, when customer information is put away in a few cloud servers by them, honesty testing and conveyed stockpiling become basic. Henceforth, contingent upon dispersed registering, it is workable for us to examine conveyed remote data respectability testing model and outfit the related solid convention relating to different cloud capacity.

### 6.6 EXISTING SOLUTION

Most existing schemes have proposed various solutions. Assume the outsourced data file F consists of a finite ordered set of blocks m1,m2,…mn. One straightforward way to ensure that the data integrity is to precomputed MACs for the entire data file. Specifically, before data outsourcing, the data owner precomputed MACs of F with a set of secret keys and stores them locally. During the auditing process, the data owner each time reveals a secret key to the cloud server and asks for a fresh keyed MAC for verification.

This approach provides deterministic data integrity assurance, We treat the leaf nodes as the left-to-right sequence. Straightforward as the verification covers all the data blocks. However, the number of verifications allowed to be performed in this solution is limited by the number of secret keys. Once the keys are exhausted, the data owner has to retrieve the entire file of F from the server in order to compute new MACs, which is usually impractical due to the huge communication overhead. Moreover, public auditability is not supported as the private keys are required for verification. Another basic solution is to use signatures instead of MACs to obtain public auditability.

The data owner precomputed the signature of each block M1,M2,…..Mi and sends both F and the signatures to the cloud server for storage. To verify the correctness of F, the data owner can adopt a spot-checking approach, i.e., requesting a number of randomly selected blocks and their corresponding signatures to be returned. This basic solution can provide probabilistic assurance of the data correctness and support public auditability. However, it also severely suffers from the fact that a considerable number of original data blocks should be retrieved to ensure a reasonable detection probability, which again could result in a large communication overhead and greatly affects system efficiency. Notice that the above solutions can only support the case of static data, and none of them can deal with dynamic data updates.

## 6.7 CLOUD INTEGRITY VERIFICATION PROBLEM

Cloud integrity, in terms of cloud storage security, is the assurance that information can only be accessed or modified by those authorized to do so. When users store their data in the cloud, the user hands over the data to the Cloud Service Provider(CSP). The CSP may have data centers at geographically different places. The benefit of the cloud is that the users can access their data in cloud anytime, and anywhere with a single Internet connection. But the drawback is, that the user has no knowledge or control about where the data is being stored. Data integrity at untrusted servers is an important issue. Data integrity is an act of making sure that the user's data stored in the cloud is correct and consistent. It is an action, which makes sure that data can only be accessed and altered by those who have been authorized to do so. Sometimes to efficiently manage the storage space and to save expenses, the service provider might be a bit negligent to rarely accessed ordinary users data. Hence, it becomes very essential to be able to check the status of the data stored in the cloud.

## 6.8 PUBLIC AUDITING AND DATA DYNAMICS PROBLEM

In order to perform data integrity verification, different schemes have been proposed till now. Fundamentally, all the prior schemes can be distinguish between two categories, public auditing and private auditing. Private auditing schemes are more proficient than public auditing schemes. Public auditing schemes allow for not just the user but anyone to challenge the cloud server for the accuracy of the data. This scheme takes the public auditing into consideration. It is not possible for a user to carry out an integrity check on a timely basis.

We propose an external Third Party Auditor (TPA), whose task is to verify the user's data stored in the cloud, on behalf of the user. Public auditing is nothing but a verification of the user's data performed by an external TPA, on behalf of the user. For security purposes, the TPA will perform stateless verification, i.e. verification will be performed without retrieving the user's data stored in the cloud. This scheme also addresses the issue of dynamic data operations. When the user stores his data in the cloud, the user also makes some modifications to the data, whenever necessary. Data dynamics is an ability to support block level operations, such as block insertion, block deletion, block modification on the data stored.

It is necessary to check whether the changes made to the data by the user are replicated properly to the data stored in the cloud. This scheme proposes public auditing and dynamic data support for ensuring cloud data storage integrity. Further, this scheme aim to assess the performance of the TPA by enabling to calculate the time required to accomplish each verification request by the TPA.

To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met:

- TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user.
- The third party auditing process should bring in no new vulnerabilities towards user data privacy.

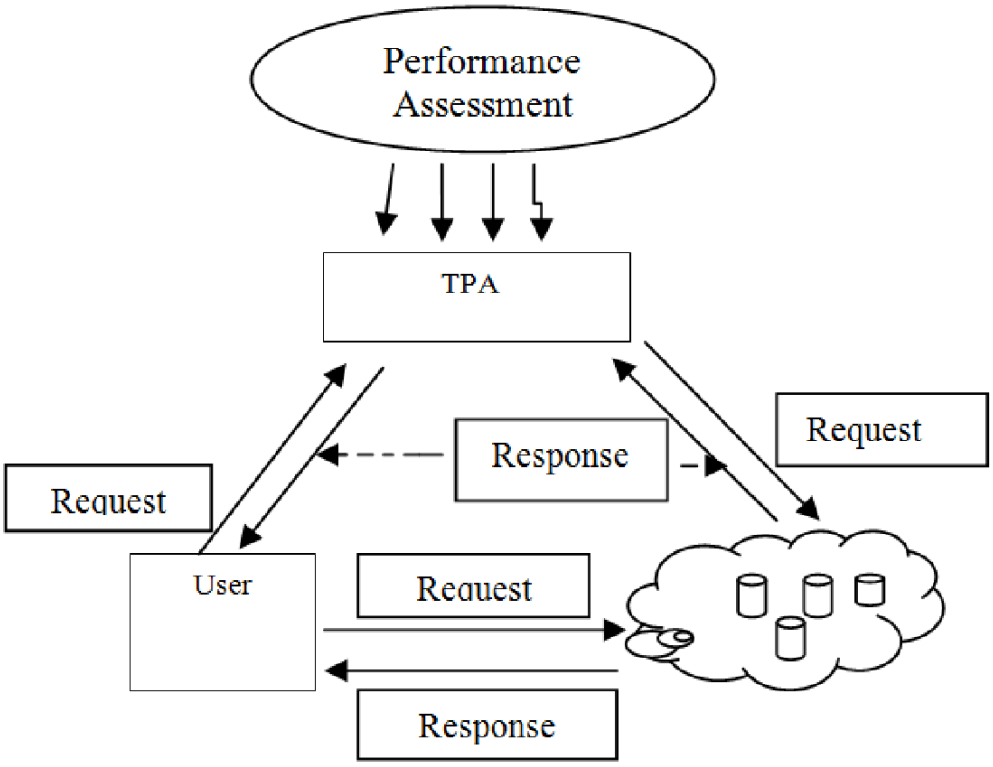



FIGURE 6.1 PROPOSED SYSTEM ARCHITECTURE

In short, “To encourage the public auditing and dynamic data operation’s support for ensuring cloud data storage integrity with the provision to Assess the performance of Third Party Auditor (TPA), this scheme calculates the time required to accomplish each verification request by TPA”.

In new proposal, the Public Auditing and Data Dynamics for integrity verification of cloud storage is achieved by using Zero Knowledge Proof and CRC-64.Which enhances

the Performance of Third Party Auditor (TPA) by reducing the time taken by TPA to perform each verification

The System Architecture is shown in Figure 6.1. the following blocks can be defined:

- User**:** The user is an entity, who has some data to be stored in the cloud. The user stores his data to the cloud, and depending on the cloud for the computation and the maintenance of the data. The user can be an individual person or an organization. In this paper, we will use terms 'user' and 'client' interchangeably. Whenever the user wishes to perform a verification of his data, he sends a challenge request to the server via the TPA.
- Cloud Storage Server (CSS) or Server: This is an entity who possesses massive amount of storage space and resources needed to manage the user's data. It allows the users to store their data in the cloud storage and to manage the user's data.When the server receives a challenge request from a client, a proof of correctness gets generated and returned to the client.
- **TPA:** This is an external entity whose task is to perform the verification of the user's data, on behalf of the user. The TPA performs the verification by challenging the server, to verify that the user's data is unaltered .

  Though the TPA is going to perform the verification on behalf of the user, it should not retrieve the user's data i.e. the TPA should perform 'block less verification'. The TPA verifies the user's data without retrieving it. In the cloud model, by keeping larger files to the remote servers, the user can be relaxed from the load of storage and maintenance.

  As data stored in the cloud can be stored on any geographical distant server, the user does not possess any data locally. Therefore, it is essential for the user to ensure that the data are accurately stored and maintained. The user should be able to perform check data accuracy periodically without having any local copies of the data. Practically, it is not possible for the user to take audits on a periodical basis. The user can hand over this task to the TPA. Any TPA with the public key can become a verifier. This scheme assumes that the user can communicate with the cloud server via the CSP, to access, to modify or to retrieve their data. The user or the TPA can challenge the server on a timely basis to check the accuracy of the data. In response to this challenge the server returns proof of correctness.

  A cloud data storage service involving three different entities, the cloud user (U), who has large amount of data files to be stored in the cloud; the cloud server (CS), which

is managed by cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service security on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance.

They may also dynamically interact with the CS to access and update the stored data for various application purposes. The users may resort to TPA for ensuring the storage security of their outsourced data, while hoping to keep their data private from TPA. Considering the existence of a semi-trusted CS in the sense that in most of time it behaves properly and does not deviate from the prescribed protocol execution. While providing the cloud data storage based services, for benefits the CS might neglect to keep or deliberately delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks or Byzantine failures to maintain reputation. It is assumed that the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process.

TPA should be able to efficiently audit the cloud data storage without local copy of data and without bringing in additional on-line burden to cloud users. However, any possible leakage of user's outsourced data towards TPA through the auditing protocol should be prohibited. Note that to achieve the audit delegation and authorize CS to respond to TPA's audits, the user can sign a certificate granting audit rights to the TPA's public key, and all audits from the TPA are authenticated against such a certificate. These authentication handshakes are omitted in the following presentation.

To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, protocol design should achieve the following security and performance guarantee:

Basically cloud computing is an internet based service model, which provide on demand network access to a shared pool of configurable computing resources (e.g. networks, servers storage, applications, data etc.) can be rapidly provided with a minimal management efforts, or service provider interaction. In short, cloud computing abstracts infrastructure complexities of servers, applications, data and heterogeneous platforms Cloud Computing allows user to access software's, applications without purchasing

licensed copy of them; to use hardware and networks, platforms without establishing infrastructures; it provide metered usage of services i.e. pay as you go.

- Data storage integrity

  Integrity in cloud computing implies data integrity. Integrity means data has to be stored correctly on cloud servers and in case of failures or incorrect computing, problems have to be detected.

- Public Auditing

  By storing data on cloud user may relaxed from the burdon of storage problem, but without leaving a copy in their local computers. Sometimes the data stored in the cloud is so important that the clients must ensure it is not lost or corrupted. While it is easy to check data integrity after completely downloading the data to be checked, downloading large amounts of data just for checking data integrity is a waste of communication bandwidth. Many schemes have been proposed for Integrity Verification but those schemes were used to support static data only. Hence, integrity protocol which allow data integrity to be checked without completely downloading the data is proposed which is known as public auditing.

- Data Dynamics: The term Data Dynamics refer to the support of dynamic operations on the data stored on cloud such as block insertion, block modification, block deletion etc. Data dynamics make sure that the changes made on the data stored on cloud should be reflected properly, to maintain its integrity.

- Performance Assessment on TPA: TPA is an entity which is going to perform auditing of user's data stored on cloud on behalf of user. TPA can perform multiple auditing tasks simultaneously, it is necessary to assess the performance of TPA for better results.

- Public auditability**:** to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional on-line burden to the cloud users;
- Storage correctness**:** to ensure that there exists no cheating cloud server that can pass the audit from TPA without indeed storing users' data intact;
- Privacy-preserving: to ensure that there exists no way for TPA to derive users' data content from the information collected during the auditing process;

- Batch auditing**:** to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously;
- Lightweight**:** to allow TPA to perform auditing with minimum communication and computation overhead.

## 6.9 PROPOSED SCHEME TO MOVE TOWARDS JOURNEY OF GREEN SECURE CLOUD

In Multisignature Scheme, two pairs of prime numbers are generated. Different pairs of keys are generated for different stages as opposed to RSA where the same key is used for encryption in various stages. Decryption requires the sets of pair of keys to be used in the reverse order increasing security. This makes group validation also possible unlike RSA. To overcome the problem of Path Identification the data and keys generated are sent to a Trusted Third Party Auditor who sends the ciphertext and keys in the required order to the receiver. Hence the data is safe and prevents hacking.

During the verification phase the following steps are carried out:

Let *chalmsg* be the challenge message. Message *chalmsg* specifies the blocks to be challenged .The TPA sends a *chalmsg* message to the server and generate the random set -*Chalmsg*{i,Bi } where i ∈ I .

When the server receives the request *chalmsg*={i,Bi}, the server computes the following:

121

$$\psi = \sum_{i=s_1}^{s_c} \leq B_i \, m_i \in Z_p \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

where ψ is data set. Ψcompute

$$\phi = \prod_i^{s_c} \leq \phi_i^{V_i} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

where $\phi$ is a signature set. .

Here, both the data blocks and the corresponding signature block are combined into a single block. Additionally, the provider will also provide the verifier with a small amount of auxiliary information $\boldsymbol{\omega}$. This is the information about the nodes that reside on the path between the root and the requested node i.e. information about neighborhood nodes.

The Prover responds to the verifier with a proof

P= { ψ,**ϕ**,{H(mi), $\boldsymbol{\omega}$ }s1<i<sc, Sigsk(H(R))} …….. (3)

When the client receives the response proof from the server, it verifies the proof by computing the root R using

{H(mi), $\boldsymbol{\omega}$ }

where i ∈ I and $\omega$ is auxiliary information; then the root would be verified by using sigsk(H(R))else output false if fails.

Finally, it verifies the data block $m_i$

$$m_i,\{m_i\}\text{i.e.}(\phi.g)=e(\phi \prod_i^{s_c} \leq H\ (m_i)^{B_i}\ u^{\psi}\ ,v). \quad \ldots\ldots\ldots\ldots\ (4)$$

To ensure that changes made to the data by user have been reflected properly, this scheme provides support on data operations such as block insertion, block deletion, block modification, dynamically. To support the dynamic operations on the data, the following steps are carried out.

- Data modification:

To modify block $m_i$ to $m_i{}'$ generate a new signature for that corresponding block.

$$\phi' = (H(m_i{}'\ u^{\ m_i{}'})^{\alpha} \quad \ldots\ldots\ldots\ldots\ (5)$$

Upon getting this modification request, the server updates the file F and computes R' and returns

Pupdate = (ω,H($m_i$)),sigsk(H(R).$R'$) ..... (6)

Upon receiving Pupdate, the client computes R using

{H($m_i$),$\omega$ )

Then it performs the verification Sigsk(H(R)) and the output is 'False' if the verification fails.

The following operations Are then performed Compute Rnew using

{ $\omega$,H($m_i'$)}

i. Verify update by checking

if(Rnew = = ($R'$)sign $R'$by succeed

ii. Finally server update $R'$ signature.

- Performance Assessment

The existing scheme supports both public auditing and data dynamics but in the case of failure the TPA becomes the bottleneck to the single point failure. Hence, in the proposed scheme the main focus is on how to assess performance of the TPA. For that purpose this scheme maintains the following queue:

Let λ be the arrival time of customer's request.

Let μ be the service time of customer's requests.

Let Δ be the delay in the queue.

Let the time at which service begins is Sb.

Let the waiting time be Wi.

Let the Total time be T.

Let P be the utilization factor = λ /μ

The goal is to determine the time taken by the TPA to verify each verification request. Each verification request is placed in the queue before the TPA processes it. Before computing total time we need to consider delay in the queue. We need to compute the time at which service begins is Sb.

$$Sb = \lambda + \Delta \qquad (7)$$

Let the Service time be μ.

Let the Wi be the waiting time and computed as:

$$Wi = \Delta + \mu \qquad (8)$$

Finally, let T be the total time taken to complete each request can be computed by:

$$T = \lambda + Wi \qquad (9)$$

Where,

| | | |
|---|---|---|
| T- Total Time | λ - Arrival Time | Wi- Waiting Time |
| μ – Service time | Sb- time at which service begins | Δ- Delay in the queue |

## 6.10 METHODOLOGY AND ALGORITHM

### 6.10.1 ZERO KNOWLWDGE PROOF PROTOCOL

In cryptography, a zero-knowledge proof or zero-knowledge protocol is a method, by which one party (the *prover*) can prove to another party (the *verifier*) that a given statement is true, without conveying any information apart from the fact that the statement is indeed true.

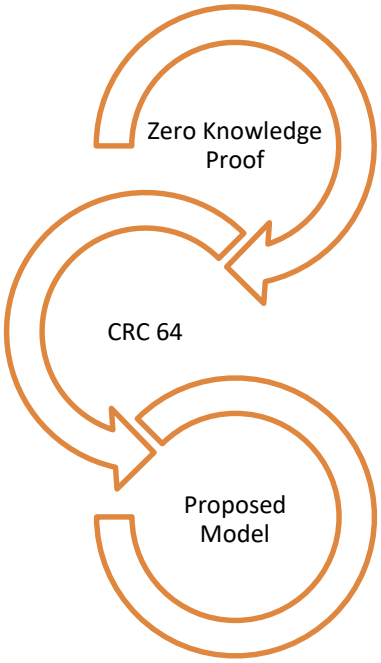



FIGURE 6.2 FUSION OF ALGORITHM

If proving the statement requires knowledge of some secret information on the part of the prover, the definition implies that the verifier will not be able to prove the statement in turn to anyone else, since the verifier does not possess the secret information. Notice that the statement being proved must include the assertion that the prover has such knowledge (otherwise, the statement would not be proved in zero-knowledge, since at the end of the protocol the verifier would gain the additional information that the prover has knowledge of the required secret information). If the statement consists *only* of the fact that the prover possesses the secret information, it is a special case known as *zero-knowledge proof of knowledge*, and it nicely illustrates the essence of the notion of zero-knowledge proofs: proving that one has knowledge of certain information is trivial if one is allowed to simply reveal that information; the challenge is proving that one has such knowledge without revealing the secret information or anything else. For zero-knowledge proofs of knowledge, the protocol must necessarily require interactive input from the verifier, usually in the form of a challenge or challenges such that the responses from the prover will convince the verifier if and only if the statement is true (i.e., if the prover does have the claimed knowledge). This is clearly the case, since otherwise the verifier could record the execution of the protocol and replay it to someone else: if this were accepted by the new party as proof that the replaying party knows the secret information, then the new party's acceptance is either justified – the replayer *does* know the secret information – which means that the protocol leaks knowledge and is not zero-knowledge, or it is spurious – i.e. leads to a party accepting someone's proof of knowledge who does not actually possess it.

A zero-knowledge proof must satisfy three properties:

1. Completeness: if the statement is true, the honest verifier (that is, one following the protocol properly) will be convinced of this fact by an honest prover.
2. Soundness: if the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.
3. Zero-knowledge: if the statement is true, no cheating verifier learns anything other than this fact. This is formalized by showing that every cheating verifier has some *simulator* that, given only the statement to be proved (and no access to the prover), can produce a transcript that "looks like" an interaction between the honest prover and the cheating verifier.

The first two of these are properties of more general interactive proof systems. The third is what makes the proof zero-knowledge.

Zero-knowledge proofs are not proofs in the mathematical sense of the term because there is some small probability, the *soundness error*, that a cheating prover will be able to convince the verifier of a false statement. In other words, zero-knowledge proofs are probabilistic "proofs" rather than deterministic proofs.

### 6.10.2 CRC (CYCLIC REDUNDANCY CHECK)

A Cyclic Redundancy Check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data. Blocks of data entering these systems get a short *check value* attached, based on the remainder of a polynomial division of their contents. On retrieval, the calculation is repeated and, in the event the check values do not match, corrective action can be taken against data corruption.

CRCs are so called because the *check* (data verification) value is a *redundancy* (it expands the message without adding information) and the algorithm is based on *cyclic* codes. CRCs are popular because they are simple to implement in binary hardware, easy to analyse mathematically, and particularly good at detecting common errors caused by noise in transmission channels. Because the check value has a fixed length, the function that generates it is occasionally used as a hash function. CRCs are specifically designed to protect against common types of errors on communication channels, where they can provide quick and reasonable assurance of the integrity of

messages delivered. However, they are not suitable for protecting against intentional alteration of data.

### 6.11 PROPOSED SYSTEM

The proposed system supports both data dynamics and public auditing. For public auditing this scheme uses the concept of Zero Knowledge Proof (ZKP) and CRC-64 hashing technique. In general, a ZKP is a way by which the prover (i.e., the server ) can demonstrate to the verifier (i.e., the TPA ) that a given statement is true, without disclosing any information apart from the fact that the statement is really true.

Cloud storage administrations have risen as an approach to address the powerful use of extra room to meet unstable development of individual and endeavor information. They enable customers to scale their extra room necessities to meet extending needs while improving usage and reasonability. For model, the capacity mists, for example, Amazon S3 , Google reports, and RackSpace show the attainability of capacity benefits in another processing worldview by advertising (nearly) boundless capacity for nothing or at low costs yet with high accessibility (24X7 days). Given its imaginative nature contrasted with the standard model of administration arrangement, cloud computing brings up new issues as far as security [3]. There is a need to comprehend the dangers related with it just as to construct advancements to address those dangers. Cloud storage administrations are not verify commonly.

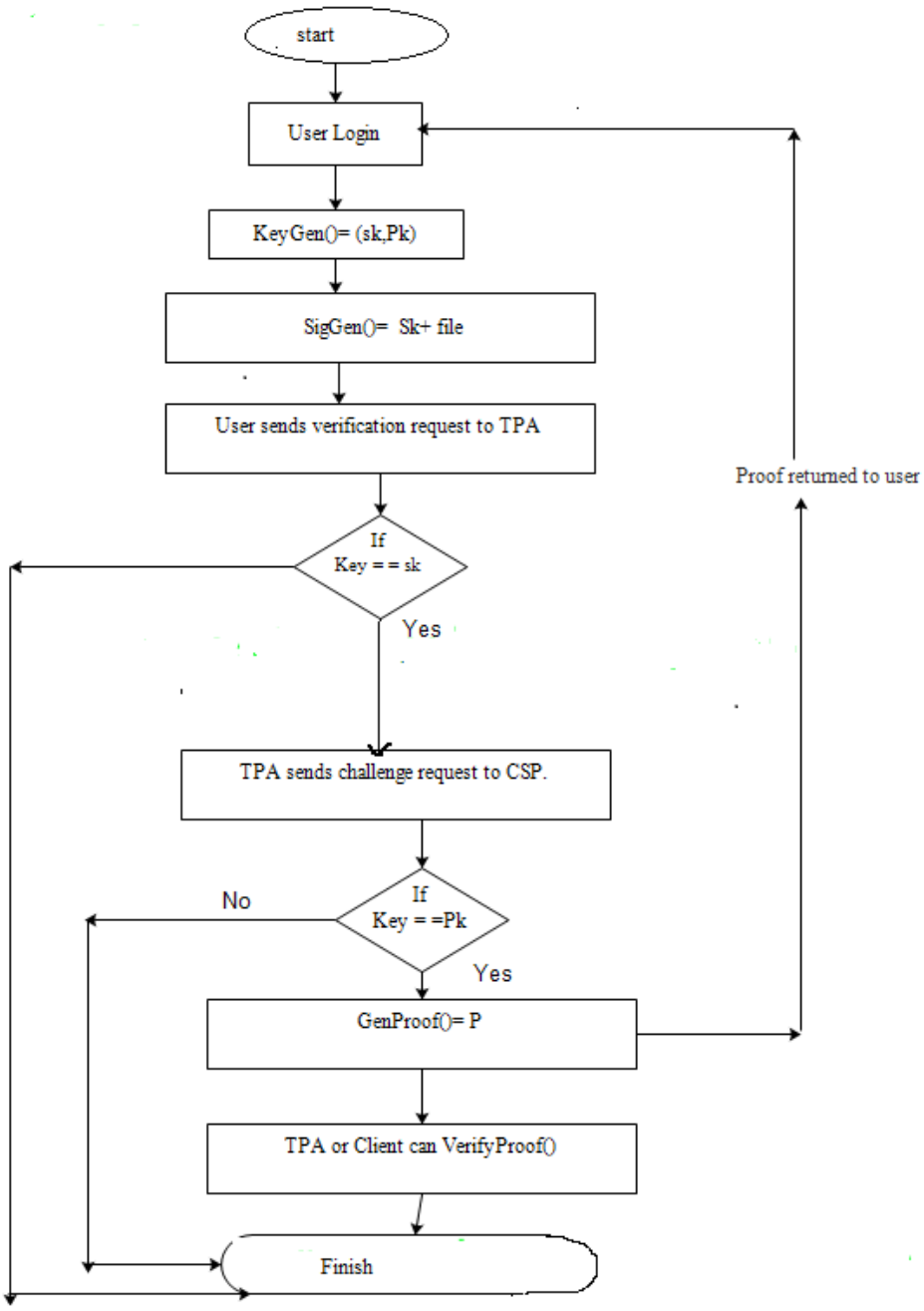



FIGURE 6.3 BASIC STEPS

The security provokes identified with distributed storage administrations need more profound consideration. As far as security, distributed storage administrations must be overseen and worked at comparable security levels to big business stockpiling frameworks. Be that as it may, cloud clients normally have no influence over the distributed storage servers utilized. This implies there is an inalienable danger of information presentation to third parties on the cloud or by the cloud supplier itself (information privacy); to information altering by the outsider on the cloud or by the cloud

supplier itself (information respectability); and to refusal of information by outsiders on the cloud or by the cloud supplier itself (information accessibility).

The distributed storage ought to guarantee information privacy, respectability and accessibility (CIA) both in movement (while transmitting over systems) and very still (while putting away at suppliers' circles). The issues of CIA in distributed storage administration have been contemplated in writing [4-6].

We saw that secrecy has gotten a more noteworthy consideration from the scientists lately [7-10]. We have additionally tended to the issue of information secrecy (C) in our prior work [11] by proposing a distributed storage administration engineering and comparing information encryption calculations and conventions with the assistance of a key the board administration. In spite of the fact that the issues of information honesty and accessibility are similarly significant, there has been less core interest on these issues.

Basically, the proposed scheme operation is divided into four steps, as follows:

| |
|---|
| $(p_{k,s_k})$ ← Keygen $(1^k)$ |
| $(\Phi, sig_{sk}\ (H(R)))$ ← SigGen($s_k$,F) |
| (P) ← GenProof(F,Φ,*chal*) |
| {TRUE,FALSE} ← *VerifyProof*($p_{k,}$*chal,P)* |
| $(F'$,Φ',$P_{update})$ ← ExecUpdate(F,$\Phi, update$)6. {TRUE,FALSE, $sig_{sk}\ (H(R'))$} |
| *VerifyUpdate* $(p_k$,*update,* $P_{update})$ |

1. KeyGen(): The KeyGen operation is carried out between the user and the cloud server. By using this step, public and secret keys are generated.
2. SigGen() : The SigGen operation is used to perform the preprocessing of the user's data . In this step, metadata of the user's file is generated and these metadata are signed using a secret key.
3. GenProof(): In the GenProof operation, the TPA gives a challenge message to the cloud server to prove that the data stored in the cloud is accurate and unmodified. The server generates the proof for the respective challenge message.
4. Verify Proof():In verify proof operation, the TPA returns the proof generated by the server to the user. After that, the user verifies his data by using a secret key.

Basically, CRC-64 is an error sensing code, which is mostly used to verify the data transmitted over a network. CRC-64 is used to check that data travelling through the network (cloud) is unaltered. In addition to the efficient provisioning of public auditing with blockless verification, the scheme uses the homomorphic authenticator technique. The homomorphic authenticators are nothing but metadata produced from distinct data blocks, which can be safely combined in such a way to assure a verifier that a linear arrangement of data blocks is properly calculated by verifying only the aggregated authenticator. In the proposed scheme, CRC-64 is used for hashing purposes, where the existing scheme uses merkle hash tree for hashing.

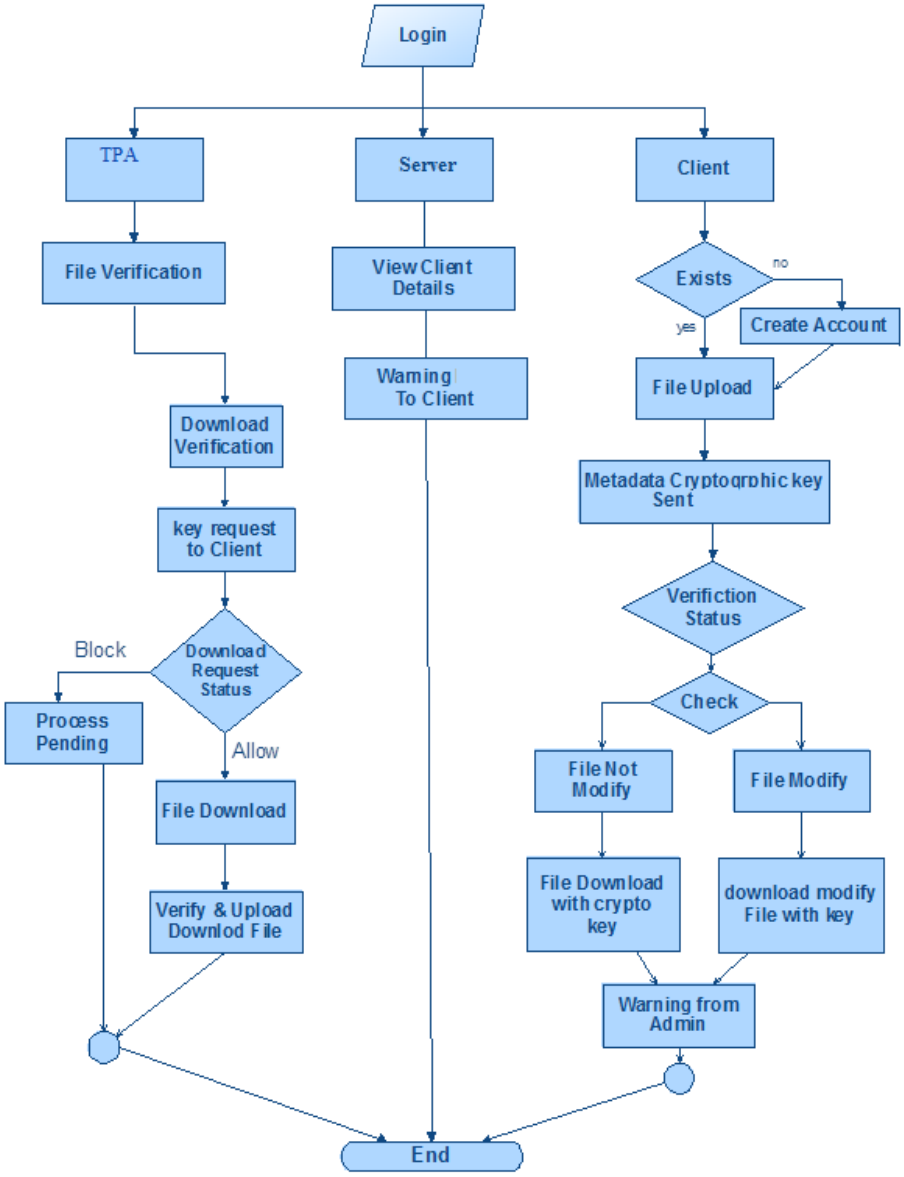



FIGURE 6.4 DATA FLOW DIAGRAM

Figure 6.4 shows the data flow sequence for both user and server. In the above diagram entities are shown by the rectangle and circle shows the processes. A text in between two lines shows

the database object. A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

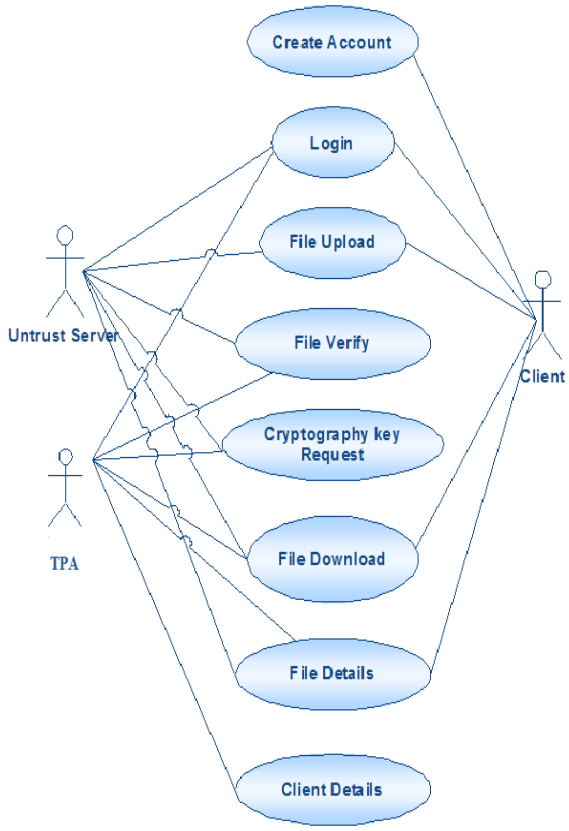



FIGURE 6.5 USE CASE DIAGRAM

A use case illustrates a unit of functionality provided by the system. The main purpose of the use-case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases. UML sequence diagrams are used to show how objects interact in a given situation. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom (i.e., **L**ower equals **L**ater).

A popular use for them is to document the dynamics in an object-oriented system. For each key collaboration, diagrams are created that show how objects interact in various representative scenarios for that collaboration. The proposed system supports both data dynamics and public auditing. For public auditing this scheme uses the concept of Zero Knowledge Proof (ZKP) and CRC-64 hashing technique. In general, a ZKP is a way

by which the prover (i.e., the server ) can demonstrate to the verifier (i.e., the TPA ) that a given statement is true, without disclosing any information apart from the fact that the statement is really true.

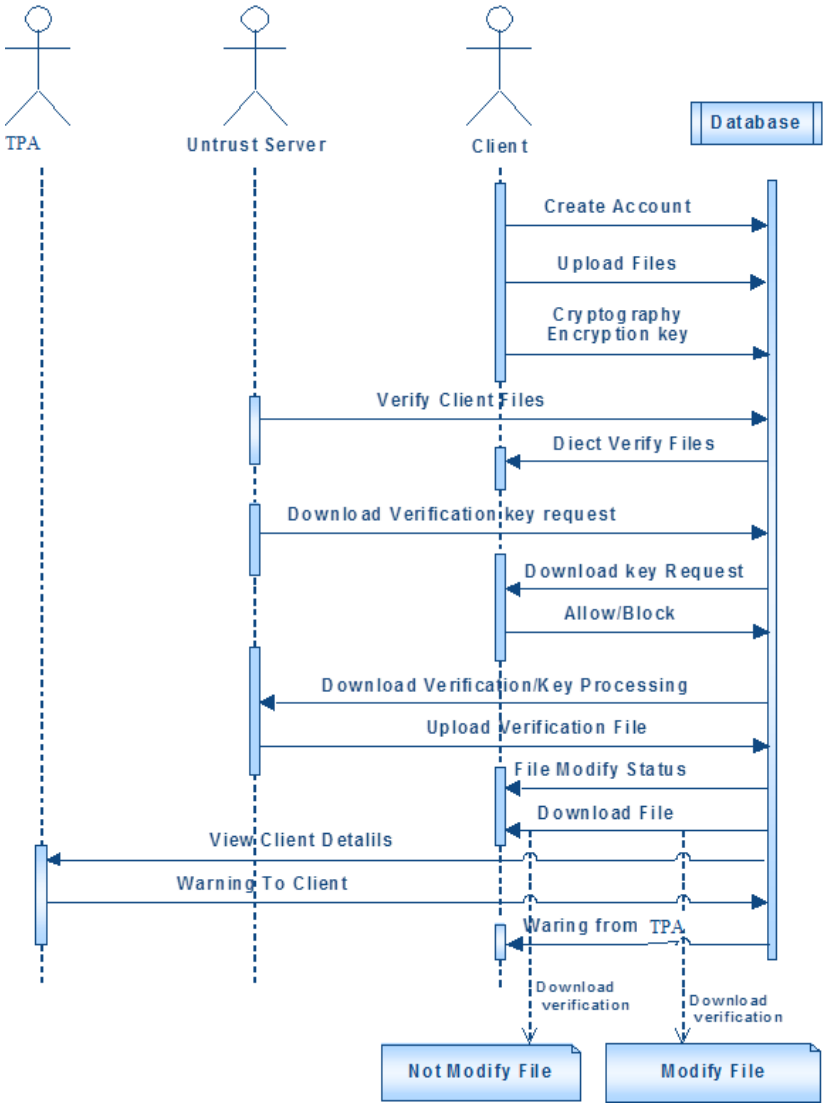
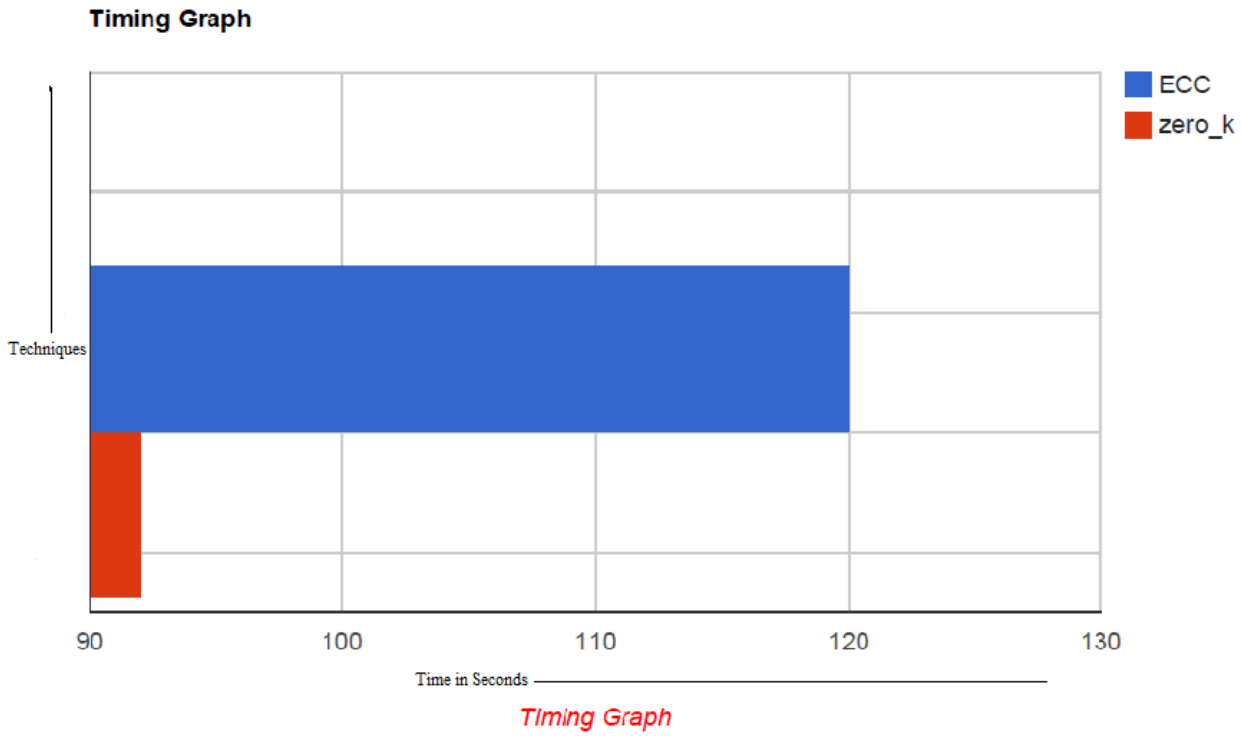



FIGURE 6.6 SEQUENCE DIAGRAM

Basically, CRC-64 is an error sensing code, which is mostly used to verify the data transmitted over a network. CRC-64 is used to check that data travelling through the network (cloud) is unaltered. In addition to the efficient provisioning of public auditing with block less verification, the scheme uses the homomorphic authenticator technique. The homomorphic authenticators are nothing but metadata produced from distinct data blocks, which can be safely combined in such a way to assure a verifier that a linear arrangement of data blocks is properly calculated by verifying only the aggregated authenticator. In the proposed scheme, CRC-64 is used for hashing purposes, where the existing scheme uses merkle hash tree for hashing. Zero Knowledge Proof (ZKP) and CRC64 gives faster results than existing Elliptical Curve Cryptography (ECC) Techniques.




FGURE 6.7 GRAPH REPRESENTING TIME REQUIRED TO PROCESS EACH USER REQUEST.

Figure 6.7 shows the overall result generated, which itself depicts that the time taken by the Zero Knowledge proof with application of CRC is much lesser than that of existing ECC techniques. the timing graph that represents the time taken by the TPA to perform the verification of each request. Above graph shows the comparison between the time taken by the ZKP technique and the Existing Elliptical Curve Cryptographic (ECC) technique. The ZKP takes less time than the ECC technique.

## 6.12 SUMMARY

For integrity verification the existing system provides the public auditing scheme using a TPA. The TPA performs the auditing of the client's data on behalf of the client using the challenge response protocol. In addition to the auditing, the existing scheme also supports data dynamics operations, such as block insertion, block deletion, block modification etc. to check whether the changes done by the client are incurred properly to the data stored on cloud. As the TPA is going to perform a number of auditing tasks simultaneously it becomes essential to determine the performance of the TPA, so that it can perform properly in the event of failure. For this purpose the proposed scheme provides a functionality to supervise the performance of TPA for smooth functioning. Our Proposed scheme mainly focuses on Time reduction. As ZKP are probabilistic rather than deterministic, it becomes essential to achieve higher level of security. Though ECC techniques takes much time for verification, they are more secure. Hence to

improve the efficiency of algorithm can be considered as a future work. The proposed scheme mainly focuses on time reduction. It becomes essential to achieve higher level of security. Though ECC techniques takes much time for verification, they are more secure. Hence to improve the efficiency of algorithm can be considered as a future work.

## REFERENCES


1. G. Ateniese, R.Burns, R.Curtmola,J.Herring,L.Kissner,Z.Peterson,D.Song " Provable Data Possession at Untrusted Stores," 14th ACM Conference on Computer and Communications Security (CCS 2007).
2. G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth Int'l Conf. Security and Privacy in Comm. Networks (SecureComm'08), pp. 1-10,2008.
3. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS '09), 2009.
4. A. Juels and B.S. Kaliski Jr., "Pors: Proofs of Retrievability forLarge Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.
5. H. Shacham and B. Waters, "Compact Proofs of Retrievability,"Proc. 14th Int'l Conf. Theory and Application of Cryptology andInformation Security:Advances in Cryptology (ASIACRYPT '08),pp. 90-107, 2008.
6. C. Erway, A. Kupcu, C. Papamanthou , and R. Tamassia,"Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), 2009.
7. Qian Wang, Cong Wang, KuiRen , Wenjing Lou And Jin Li " Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" IEEE transaction Paper on Parallel and Distributed Systems vol 22 No 5, pp. 847-859, May 2011.
8. Araddhana A. Deshmukh, Dr.Albena Mihovska, Dr.Ramjee Prasad Center for" A Cloud computing security schemes:- TGOS [Threshold Group-Oriented signature ] and TMS [Threshold multisignature schemes] "TeleInfrastruktur (CTIF), Dept of Electronic Systems, Aalborg University Aalborg, Denmark,2012. .
9. E.-C. Chang and J. Xu, "Remote Integrity Check with Dishonest Storage Server," Proc. 13th European Symp.Research in Computer Security (ESORICS '08), pp. 223-237, 2008.
10. Gowtham Kanagaraj, Naveen Shanmugasundaram And Sathish Prakash "Adaptive Load Balancing Algorithm Using Service Queue", 2nd International Conference on Computer Science and Information Technology (ICCSIT'2012) Singapore April 28-29, 2012 .
11. K.D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Report 2008/175, Cryptology ePrint Archive, 2008.
12. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. 14th European Symp. Research in Computer Security (ESORICS '09), pp. 355-370, 2009.
13. Qian Wang, Cong Wang, KuiRen , Wenjing Lou And Jin Li " Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" IEEE transaction Paper on Parallel and Distributed Systems vol 22 No 5, pp. 847-859, May 2011.
14. Snehal P. Sawant , Araddhana. A. Deshmukh," Survey on Public Auditing and Data Dynamics in Cloud Data Storage" International Journal of Science and Research (IJSR),ISSN (online) 2319-7064.
15. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. UCB-EECS-2009-28, Feb 2009.
16. D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in Proc. of Eurocrypt 2003, volume 2656 of LNCS. Springer-Verlag, 2003, pp. 416–432.
17. A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 584–597.
18. M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proc. of HotOS'07.
19. D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in Proc. of ASI- ACRYPT'01. London, UK: Springer-Verlag, 2001, pp. 514–532.

# Chapter 7
# Journey towards Green cloud and future business model

## 7.1 INTRODUCTION

One of the major issues of cloud computing is load balancing because overloading of a system may lead to poor performance which can make the technology unsuccessful This model requires clients to buy never-ending or membership-based permit and deal with the organization themselves, including progressing between various renditions. Thus, clients need specialized ability and high starting venture for purchasing programming. They, likewise, need to pay for overhauls as yearly support charge. With the rise of Software as a Service (SaaS), applications are moving far from PC-based or proprietorship based projects to web conveyed facilitated administrations. The product administrations are provisioned on a compensation as-you-go premise to beat the restriction of the conventional programming deals demonstrate. Utilizing the SaaS show, suppliers increase enduring, on-going income from their clients. In return for the on-going charges, the clients get the advantage of consistently looked after programming. Henceforth, there is no extra permit charge for new forms and the multifaceted nature of progressing to new discharges is overseen by SaaS suppliers. Because of the SaaS model's adaptability, versatility and cost-viability, it has been progressively received for circulating numerous venture programming frameworks, for example, banking, web based business programming.

SaaS suppliers, for example, Computer Associates (CA) get their benefits from the edge between the operational expense of foundation and the income produced from clients. Hence, SaaS suppliers are investigating arrangements that limit the general framework cost without unfavorably influencing the clients. Consequently, the focal point of this paper is on investigating approaches to limit the expected foundation to satisfy client need with regards to SaaS suppliers offering facilitated programming administrations.

A client sends demands for using endeavor programming administrations offered by a SaaS supplier, who utilizes three layers, to be specific application layer, stage layer and foundation layer, to fulfill the client's demand. The application layer deals with all application benefits that are offered to clients by the SaaS supplier. The stage layer incorporates mapping and planning approaches for interpreting the client's Quality of Service (QoS) prerequisites to framework level parameters and apportioning Virtual Machines (VMs) to serve their solicitations. The framework

layer controls the genuine commencement and evacuation of VMs. The VMs can be rented from IaaS suppliers, for example, Amazon EC2 or private virtualized bunches claimed by the SaaS supplier. In the two cases, the minimization of the quantity of VMs will convey reserve funds. The reserve funds are more prominent when SaaS suppliers utilize the outsider IaaS suppliers since no capital use is required.

The concept of Parallel and Distributed computing is trending recently. With the help of Parallel Computing and Load Distribution (NLB and SLB), the computation time for any process or task provided to any system can be reduced. Using a pool of computers connected together to a network and distributing a non-atomic task among them using a pool manager and task dispatcher enables the task to be completed in much less time.

Educational institutions have computer labs. The computers in these labs are active or in use only during the working hours of institution. Out of the working hours, these computers are always in shutdown state. When seen as a resource, we can say that we lack in the optimal use of these resources. Institutions have a strong infrastructure to utilize these resources. These computers when connected to a network can help in formation of a pool of computers. Using this pool of computers, we can enhance the computational capacity of any system by distributing the non-atomic task over the pool of computers.

Cloud computing has been considered as an answer for explaining venture application dissemination and design difficulties in the customary programming deals show. Relocating from conventional programming to cloud empowers on-going income for programming suppliers. Nonetheless, so as to convey facilitated administrations to clients, SaaS organizations need to either keep up their own equipment or lease it from framework suppliers. This necessity implies that SaaS suppliers will acquire additional expenses. So as to limit the expense of assets, it is additionally essential to fulfill a base administration level to clients.

The present works in cloud processing are centered generally on augmenting the benefit of IaaS suppliers, yet works identified with the SaaS supplier considering SLAs are still in their earliest stages. Numerous works do not think about the client-driven administration, where assets must be progressively modified dependent on clients' requests. In this manner, in this research, we look at the asset allotment procedures, which permit a financially savvy use of assets in clouds to fulfill progressively changing client requests in accordance with SLAs.

The goal is to limit the cost by improving the asset assignment inside a VM. These arrangements likewise consider QoS parameters, and foundation heterogeneity with respect to numerous kinds of VMs and different administration commencement times [1].

7.2 PROPOSAL FOR GREEN CLOUD

Computing has been considered as an answer for understanding venture application circulation and setup challenges in the conventional programming deals show. Relocating from conventional programming to the cloud empowers on-going income for programming suppliers. To convey facilitated administrations to clients, SaaS organizations need to either keep up their own equipment or lease it from framework suppliers. This necessity implies that SaaS suppliers will bring about additional expenses. In order to limit the expense of assets, it is essential to fulfill a base administration level to clients. This will be helpful where a substantial pool of framework are associated in private or open systems, which gives powerfully versatile foundation to application, information and document stockpiling. [1].

It is proposed to use asset distribution calculations for SaaS suppliers who need to limit framework cost and SLA infringement. The calculations are planned in an approach to guarantee that SaaS suppliers can deal with the dynamic difference in clients, mapping client solicitations to framework level parameters and taking care of heterogeneity of VMs. In this, the clients' QoS parameters (e.g.,reaction time), and framework level parameters, (e.g., administration inception time) would be considered. By introducing a broad assessment it can be shown how the calculations limit the SaaS supplier's expense and the quantity of SLA infringement in a dynamic asset sharing cloud environment [2].

The clients' solicitations for the venture programming administrations from a SaaS supplier will be considered by consenting to the pre-characterized SLA statements and presenting their QoS parameters. Clients can progressively change their necessities and utilization of the facilitated programming administrations. The SaaS supplier can utilize their own framework or re-appropriated assets from open IaaS suppliers. For example, "Saleforce.com" gives CRM programming as an administration utilizing its own framework, and "Force.com" offers this product utilizing outsider foundation. The SaaS supplier's goal is to plan a demand with the end goal that its benefit is expanded, while the clients' (QoS) prerequisites are guaranteed. The stage layer of a SaaS supplier utilizes mapping and booking instruments to translate and examine the clients' QoS parameters, and dispenses separately [3],[4].

**7.3 EXISTING SYSTEM**

Currently, SaaS providers such as Compere ERP provide an individual VM for each customer to maintain service level requirements in terms of response time and capacity. However, this causes wastage of hardware resources which results in high infrastructure cost since customers may not use complete VM capacity which is reserved to serve their requests. A

multi-tenancy approach can reduce the needed infrastructure, but care must be taken in providing access to resources so that Service Level Agreements (SLAs) are not violated. [1].

The current works in cloud computing are focused mostly on maximizing the profit of IaaS providers, but works related to the SaaS provider considering SLAs are still in their infancy. Many works do not consider the customer driven management, where resources have to be dynamically rearranged based on customers' demands. Thus, in this Chapter, we examine the resource allocation strategies, which allow a cost effective usage of resources in clouds to satisfy dynamically changing customer demands in line with SLAs. [3].

The key contributions are as follows:

- Definition of SLA with customers based on QoS parameters.
- Mapping strategy by interpreting customer request requirements to infrastructure level parameters.
- Design and implementation of scheduling mechanisms to maximize the SaaS provider's profit by reducing the infrastructure cost and minimizing SLA violations. The scheduling mechanism determines where and which type of VM has to be initiated by incorporating the heterogeneity of VMs in terms of their price, dynamic service initiation time, and data transfer time. In addition, it manages to reduce incurred penalties for handling dynamic service demands when customers are sharing resources. [1].

  This proposal also presents a performance analysis of the proposed algorithms based on the customer's perspective: (i) arrival rate, (ii) proportion of upgrade requests; from SaaS.

  The basis for the proposed framework is that client requires for big business administrations from a SaaS supplier by consenting to SLA parameters and referencing QOS

  parameters. The SaaS supplier can utilize their supplier. The SaaS supplier's goal is to plan work according to the SLA given at the season of enrollment. It ought to be skilled to change its state once SaaS changes its SLA. The framework has to carry out a sperate procedure to plan, which occupations ought to be put into execution. It has also to be adaptable, which implies that its execution ought not to debase with the expansion of hubs and occupations. It ought to be setup, and consider different planning policies that can be adjusted to fuse QoS parameters, framework handles dynamic VM exchanging according to

SLA and bolster dynamic burden booking. The fundamental errand is of limiting the quantity of SLA infringement in a dynamic asset sharing.

To fulfill the client's solicitations so as to grow the showcase store and limit the cost, the accompanying inquiry must be tended to. We consider the client's demand for the endeavor programming administrations from a SaaS supplier by consenting to the predefined SLA conditions and presenting their QoS parameters. The client can powerfully change their necessity and utilization of the facilitated programming administrations. The SaaS supplier can utilize their very own framework or re-appropriated assets from open IaaS suppliers. A booking component to amplify an SaaS supplier's by decreasing the framework cost and limiting SLA infringement is structured. The planning component figures out where and which sort of VM must be started by consolidating the heterogeneity of VM's regarding their value, dynamic administration inception time and information exchange time. It likewise figures out how to lessen punishments for taking care of dynamic administration requests when client is sharing asset.

The primary motivation behind the scheduler is to augment the utilized utility. The scheduler will intend to upgrade the asset usage inside the client forced limitations. Along these lines, client fulfillment is the essential worry instead of amplifying CPU use. The scheduler will dispense work situated in the activity parameters, which are work particular put together by the client with the activity.

The proposed planning is executed as another booking approach in the CloudSim reproduction instrument with the end goal of assessment. It is a system for demonstrating and reproduction of the cloud computing foundation and administrations in repeatable and controllable condition free of expense and tunes the execution bottleneck before sending on genuine mists. At the supplier side, reenactment situations permit assessment of various types of assets renting schematic under fluctuating burden and valuing appropriation, studies could did the supplier in enhancing the asset get to cost with spotlight on improving benefit. It helps in finding and evacuation of mistakes before actualizing continuously.

A SaaS supplier can expand the benefit by limiting the asset cost, which relies upon the number and sort of started VMs. Consequently, this calculation is intended to limit the quantity of VMs by using previously started VMs. Calculation 1 portrays the ProfminVmMaxAvaiSpace calculation, which includes two fundamental demand types: a) first time lease and b) redesign administration. Let the demand of a client c incorporates

ask for sort (reqType), item type (proType) account type (accType) number of account (accNum). The calculation checks the demand type, if the demand rype is 'first time lease' the it finds the VM, with sort/(VM) that matches to the administration ask for parameters utilizing mapping table.

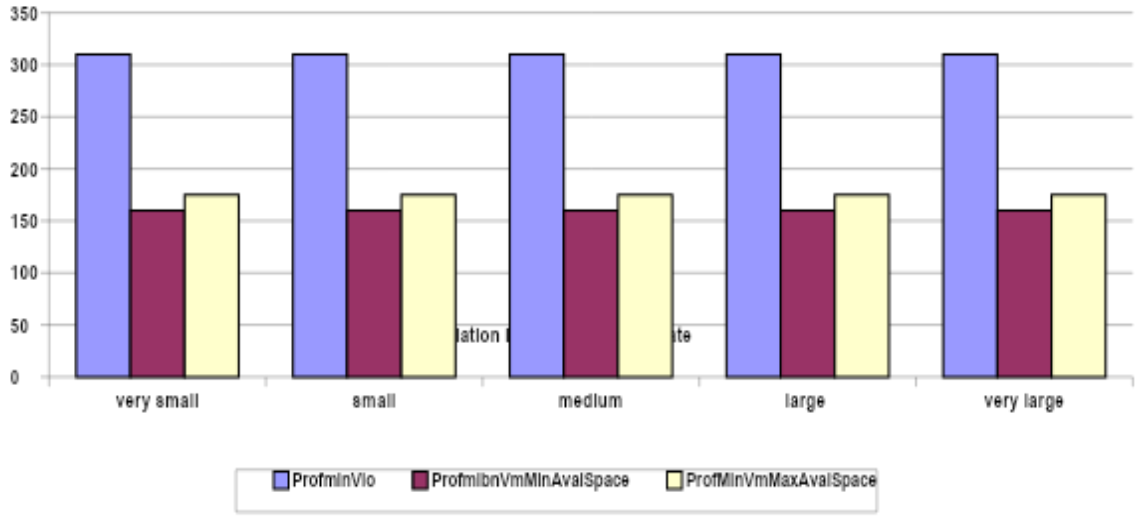



FIGURE 7.1 ALGORITHMIC COMPARISONS FOR PROFMINVMMAXAVAISPACE CALCULATION

The framework essentially utilizes two calculations: Calculation 1: ProfminVmMaxAvaiSpace

At that point, it checks whether there is as of now started VM, and conveyed with same kind of item as client c asked. On the off chance that there is a started VM, where item proType has been sent as client ask for, the calculation checks whether this VMil has enough space to put the demand of client c as per his/her asked for number of records (accNum) and the accessible space on this VM. On the off chance that there are more than one VMil with enough accessible space to put the demand c, at that point the demand c is doled out to the machine with most extreme accessible space (Worst-fit way) as showed in Fig.7.1 (b).

The dark space demonstrates inaccessible space, x hub shows the id of VM with same sort and sent with same kind of item as client c asked for; y pivot demonstrates number of records a VM can hold). On the off chance that there is no started VM with sort l, at that point check the following kind of VM - VMi(l+1) which is sent with a similar programming item type as demand c determined, rehash step (2 to 13).If the request type is ‘upgrade’, then the type of upgrade is checked. If upgrade type is ‘add account’, then it first checks VM*i*, which has placed the previous request *c’* from the same company. If VM*i* is not capable to place new request without exceeding the number of account limitation, then the suitable type *l* of VM is found that

has the maximum available capacity to place request *c*. Then move the previous data from VM*i* to new VM and release the space occupied by old request from VM*i*. On the other hand, if upgrade to more advanced product edition, the new request is placed to the suitable VM by using the *MaxAvaiSpace* Strategy (Fig.7.1(b)) and then the customer data is migrated to new VM and release the space occupied by old request from *VM i*.

***First Time Rent (c) {***
***1 If*** *(there is initiated VMi with type l matches to the VM type requested by c) {*
***If*** *(VMi deployed the same product type as c required) {*
***For each*** *initiated VMi with type l (VMil) {*
***If (****VMi has enough space to place c****) {***
*put VMi into vmList* ***} }***
***Sort (****vmList****)*** *according to the available space*
*Schedule to process c on* ***VMmax****, which has maximum available space*
***}***
***Else {***
*Initiate new VM with type l and deploy the product type as request c required*
***} }***
***Else While (l+j<=L) loop {***
***If*** *(there is initiated VM with next type l+j, where type l+j matches to the VM type required by request c) {*
***Repeat from Step 2 to 13***
***j++***
***} }***
***} Upgrade(c)*** *{*
***If (****upgrade type is 'add account'****) {***
*get Id i and type l of VM, which processed the previous request from same company as c*
***If (*** *VMi has enough space to place c){*
*Schedule to process c on VMi.*
*}* ***Else {***
*Repeat step 1 to 21 of* ***First Time Rent(c)***
*Transfer data from old VM to new VM*
*Release space in old VM*
***} }***
***If (****upgrade type is* ***'upgrade service'){***
*Repeat step 7 to 9 of* ***Upgrade(c) }***

**Algorithm 2:** ProfminVmMinAvaiSpace

To overcome the disadvantages of algorithm 1, we reduce the space wastage by using minimum available space (*MinAvaiSpace*) Strategy (Fig. 7.3) instead of *MaxAvaiSpace* Strategy. When there are more than one *VM* with type *l*, deployed with the same product type as customer request *c* required, the VMs with enough available space to serve *c* are selected. Then request *c* is scheduled to the machine with the **minimum** available space (Best-fit manner) (Step 9). The rest of steps are the same as Algorithm 1.

## 7. 4 BUSINESS LOGIC AND ARCHITECTURE

### 7.4.1 BUSINESS LOGIC

Business rationale is the programming that oversees correspondence between an end UI and a database. The fundamental segments of business rationale are business tenets and work processes. A business rule portrays a particular strategy; a work process comprises of the undertakings, procedural advances, required info and yield data, and instruments required for each progression of that method. Business rationale portrays the succession of activities related with information in a database to do the business rule. In PC programming, business rationale or area rationale is the piece of the program that encodes this present reality business decides that decide how information can be made, shown, put away, and changed. It is stood out from the rest of the product which may be worried about lower-level subtleties of dealing with a database or showing the UI, framework foundation, or by and large interfacing different pieces of the program.

Web based business and computerized administrations rely upon server farm and proceed in augmentation of data innovation. History of Software centers mostly around psychologist wrapped programming deals show. Client need to buy membership based permit. It additionally included administration of advancement and proposed and pays for non required programming or equipment cost. Despite the fact that prerequisites are diminished, we have to pay for those pathetic administrations. At that point there is presentation of cloud parameters which made accessible us uses like every single other utility you are charged dependent on what you expend. There was a progress from conventional programming framework to Software as a Service (SaaS). Cloud administration gives after their administrations in either (or both) of this two ideal models "Foundation as a Service (IaaS) and Platform as a Service (PaaS) with the assistance of programming as an administration in application layer of cloud parameters.

The main focus of system is on satisfaction of both the parties' viz. clients and service provider. The profit of service provider depends on reducing number of SLA violations. The system mainly works for service providers. If there are SLA violations, the service provider has to pay penalty. A user login is provided for service provider and can create an account. After logging in the service provider can view his profile with details, product type, account type and no. of account. on selecting first time result, a service level such as request type, product type,

account type, contract length(in months), no. of account, no. of records and response time after submission of services profile i.e. first time request, it is saved in database. Service provider can view statistic in graphical format. The graph is plotted no. of account against VM's.

Service provider can select upgrade type which may be add new account or upgrade product type and product type as standard, professional and enterprise. Single account gets added each time. After submission, a graph is displayed which give comparison between previous graph and new graph. This can be done by two ways as our algorithm states viz.

- ProfMinVmMaxAvaiSpace Upgrade
- ProfMinVmMinAvaiSpace Upgrade

The account can be added as per SLA. If SLA is violated, a warning is shown on screen that SLA is violated. We are not restricting addition of account as it will be inconvenience for client of service provider but service provider is instructed through SMS that he has violated SLA and should upgrade service. Thus the SLA violation are checked or avoided.

### 7.4.2 PROPOSED ARCHITECTURE

The system works in all is layers viz. SaaS, PaaS and IaaS. Software of a Service is provided by GUI. The accepting inputs become easy due to Graphical User interface. These inputs are stored in database. Here, database plays role of Platform as a Service. It connects Cloudsim i.e. IaaS with GUI i.e. SaaS. The third and most important part of architecture is Infrastructure as a Service. Here, Cloudsim acts as an IaaS. Cloudsim consists of various datacenters. Data centers are the backbone of the growth in an e-commerce and digital services of continue to grow in size as the demand for information technology increases. Datacenter resource is heterogeneous, even if a data center and initially provisioned with homogeneous resources.

Each Data center contains n number of hosts (Fig. 7.4). These hosts combine together to form a cloud. Each Host can contain various operating with virtual machines. Cloudlet is a basic user entity that performs actual execution task. Cloudlet is an extension to the Cloudlet, the ID of the VM running it. Broker act as a middleware. It firstly tries to create virtual machines on datacenters

dynamically after initialization of CloudSim. Broker creates VM's in specific datacenter that exists on specific host.

After the creation of VM's, broker send cloudlets on various VM's, the message of receiving cloudlets on VM is shown. Then there is execution of cloudlets. After execution, the VM's are destroyed. After successful execution, the broker shut down. Eventually, the broker also shuts down and the simulation is completed. The output is shown in tabular format as well as in graphical format on GUI for user convenience.

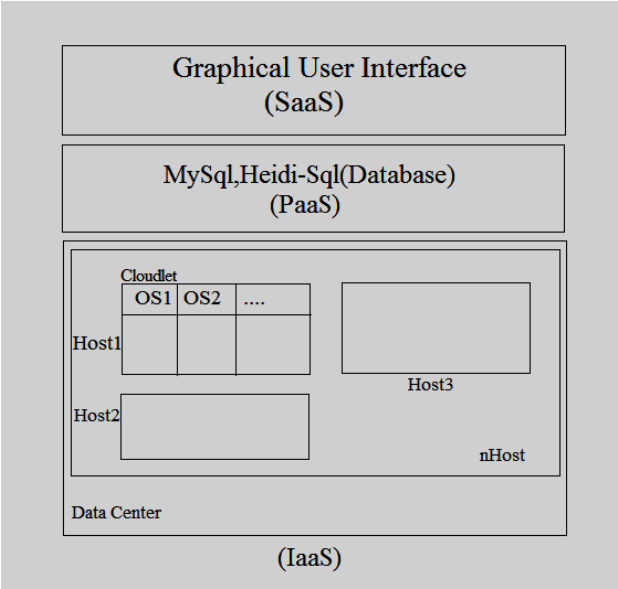



FIGURE 7.2 SYSTEM IMPLEMENTATION ARCHITECTURE

We will analyze ways to increase the efficiency of the algorithms in terms of total profit and shall also consider the SLA negotiation process in cloud computing environments to improve customer satisfaction levels. We would also like to add different types of services and other pricing strategies such as spot pricing to increase the profit for service providers. Moreover, investigating the knowledge based scheduling for maximizing a SaaS provider's profit to improve our algorithms' time complexity. Moreover, we will look into the penalty limitation by considering system failures.

Using IaaS with SLA rendering is a process of automatic generating a photo-realistic or non-photo-realistic image from a 2D or 3D model by means of computer program. When a cluster of computers connected together for rendering is created, it is termed as render farm. These computers form a distributed system.

Making the use of Idle Computers in labs for creating a render farm is presented in this project. Also, using the idle computers for harvesting the frames turns out to be resource efficient, hence avoiding the need of excess dedicated hardware.

## 7.5 GREEN ARCHITECTURAL DESIGN

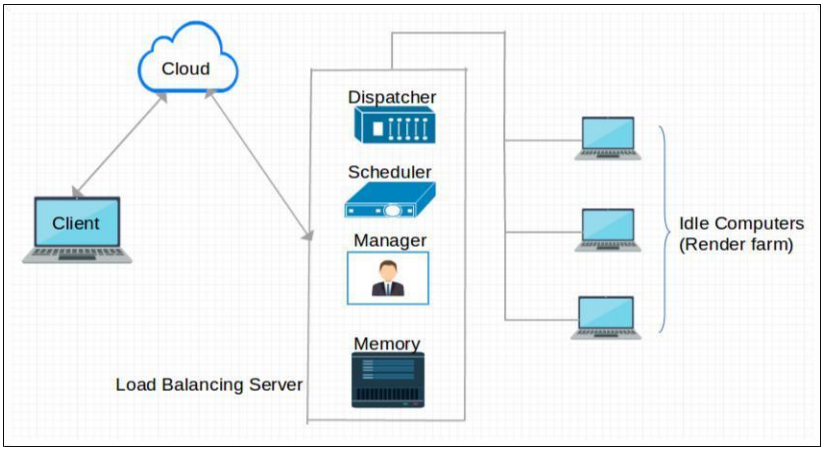



FIGURE 7.3 SYSTEM ARCHITECTURE FOR GREEN APPROVAL

The proposed system architecture involves Client systems, Cloud Server, Load Balancing Server and Idle Computer pool (Render Farm). The clients will be uploading the les to be rendered onto the cloud server. These les will be fetched from the cloud server by the Load Balancing Server. Once the les are fetched, the task of rendering those les will be distributed over the render farm. After completing the task of rendering, the les will be uploaded to the memory of the server, from there the output will be uploaded onto cloud server from where the user can download the required output le. The logical output will be Client-Cloud Server-Load Balancing Server-Render Farm and then back to client in the reverse order. The goal is to design a system, which will provide enhanced computational power to the user, which is required for performing high end tasks such as Graphics Rendering.

Green i.e. usage with optimization include following key strategies :-

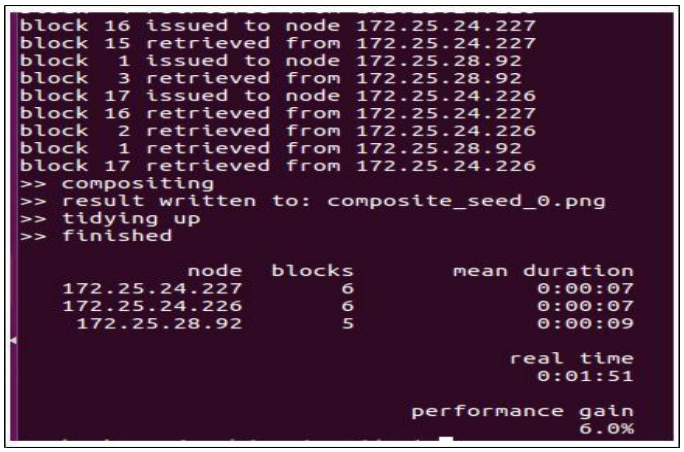
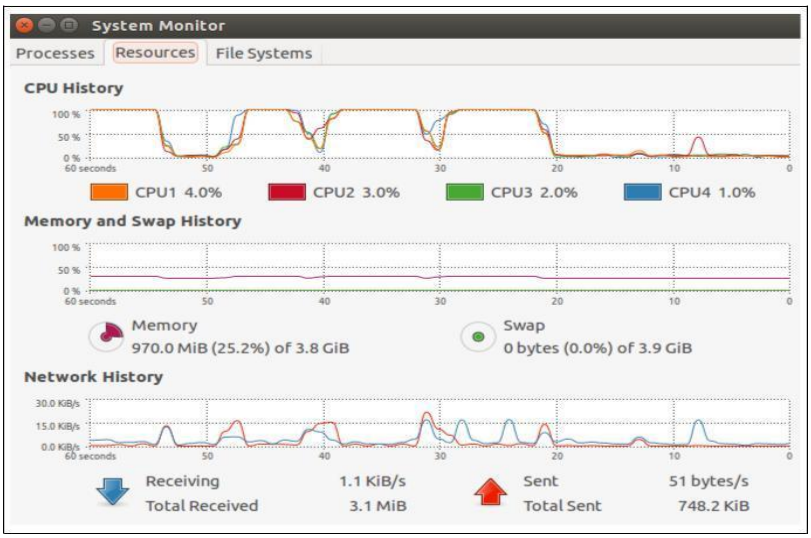



*FIGURE 7.4 SERVER CLI UI SNAP : WORKING*



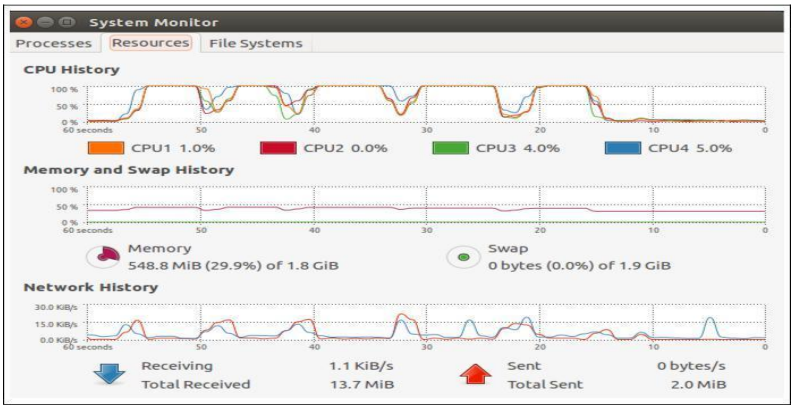
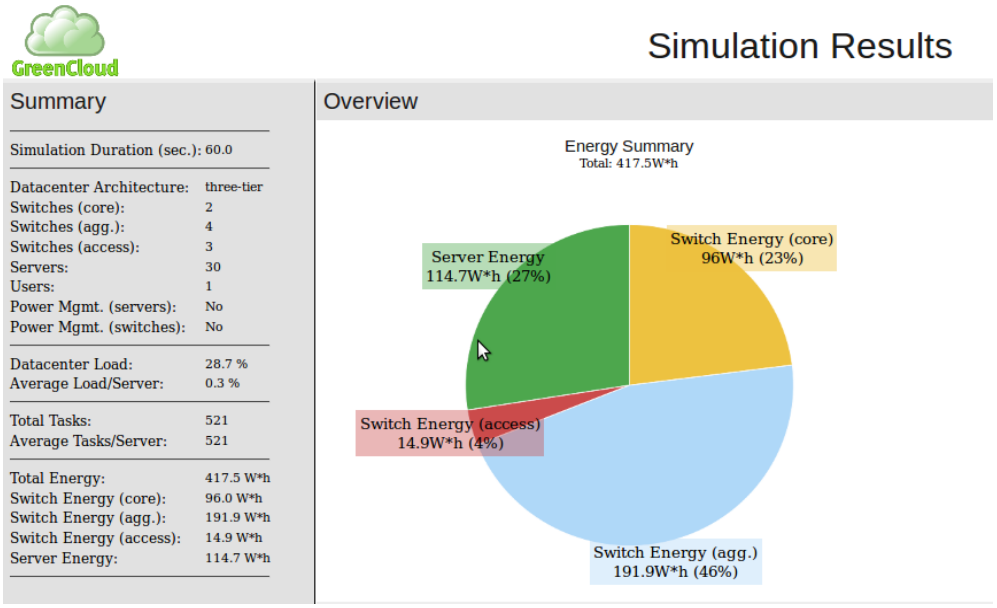




FIGURE 7.5 IMPLEMENTATION SCREENSHOT: OUTPUT

- { Provide users with a convenient, reliable, client graphics rendering platform by using new technologies to effectively integrate computing resources, data resources and software resources.
- { Make the system work across various platforms.
- { Avoid the under-utilization of computers in college labs.
- { Make use of Load Balancing Concept
- { Learn various scheduling strategies and use the best strategy.



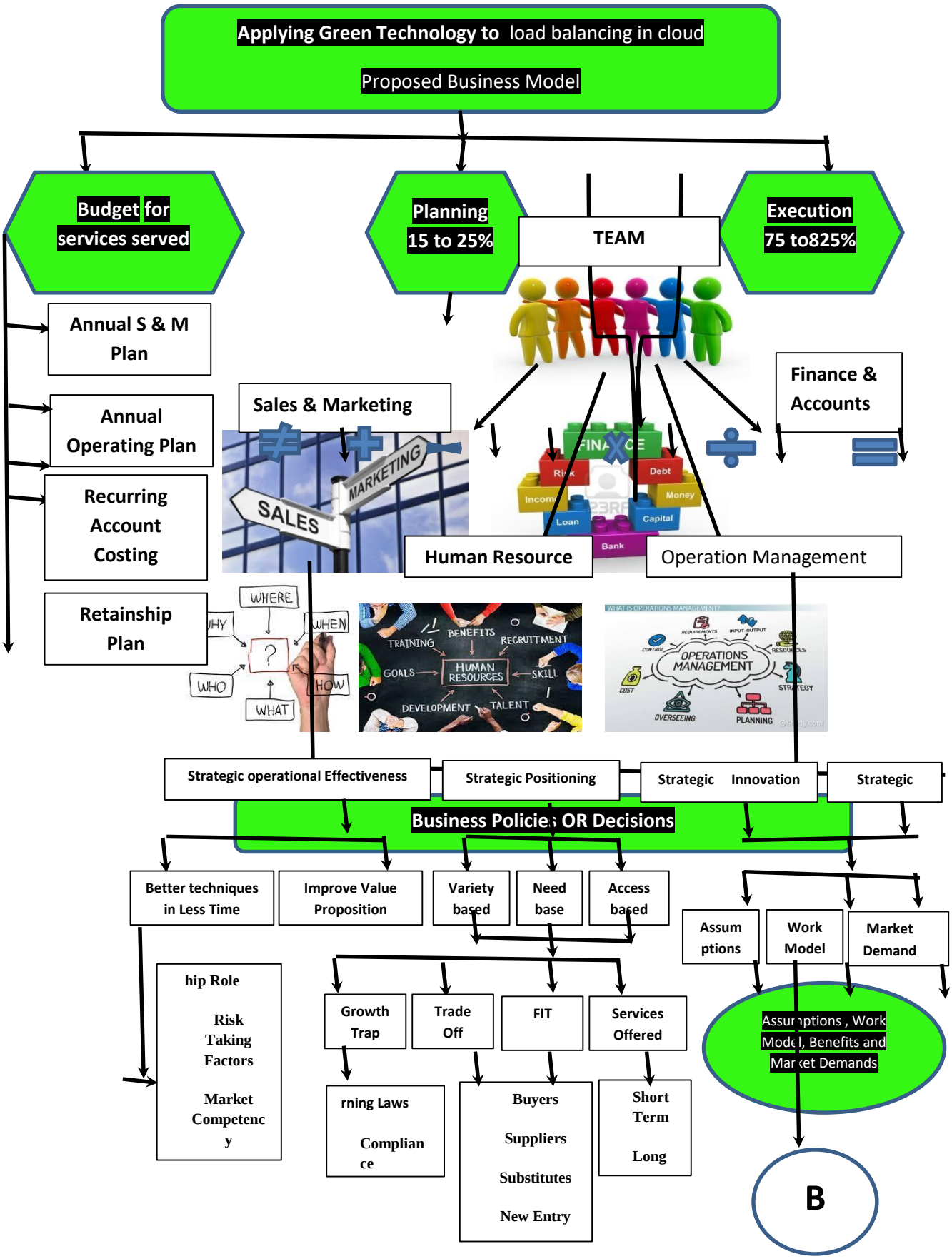



FIGURE 7.6 BUSINESS MODEL FOR SUCCESSFUL IMPLEMENTATION OF GREEN CLOUD

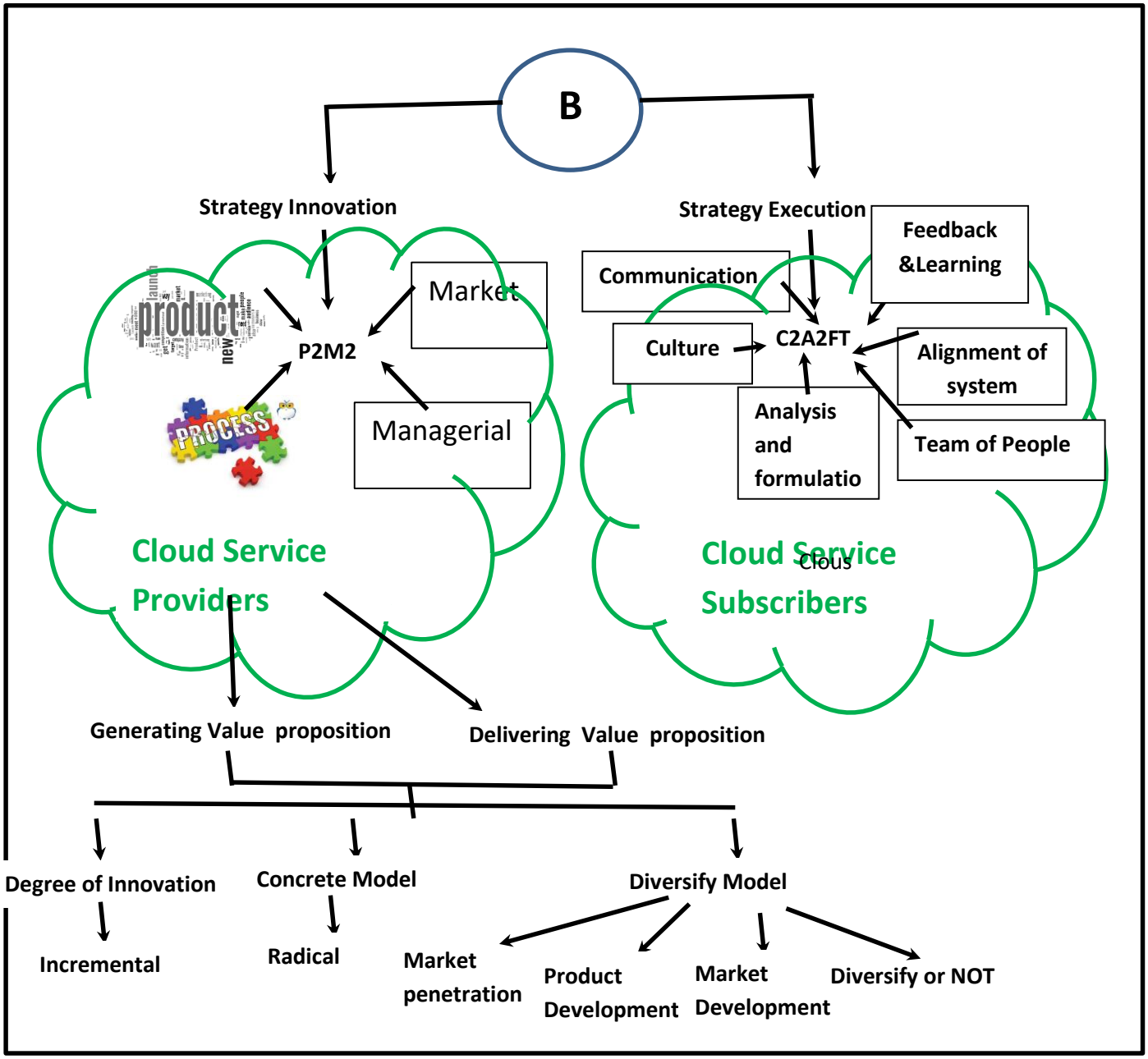



FIGURE 7.7 DEPUTIZE BUSINESS MODEL FOR SUCCESSFUL IMPLEMENTATION OF GREEN CLOUD

## 7.6 SUMMARY

A solution to how idle computers can be used optimally for creating a render farm and how the load on a server be balanced using man-aging agent is presented. Use of Benchmarking and Occupancy algorithms helps in better utilization of resources which stay idle in labs.

## 7.7 FUTURE SCOPE WITH BUSINESS MODEL

Expanding the network of idle nodes on a global level and implementing this concept on block chain device is the future scope.

SaaS additionally gives advantages to natural assurance: through centralization of handling and administration sharing, it unites server farm tasks so as to utilize less hardware.

SaaS suppliers could offer green programming administrations conveyed on green datacenters with less replications or they could utilize calculations that improve programming vitality proficiency without abusing Administration Level Agreements (SLAs). The cloud suppliers have more assets and more inspiration than individual clients need to put resources into natural security. On account of PaaS, the suppliers could offer offices, for example, green calendar and green compilers. To support natural security through green distributed computing, both SaaS and PaaS suppliers have strategies and instruments to accomplish programming level vitality enhancement.

The expansion in the prominence of cloud innovation was because of the advantages it brought to singular purchasers and organizations. These advantages include: adaptability, fiasco recuperation, diminished interest in ICT assets, improved joint effort between individuals from an association.

**REFERENCES**


[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M. (2009). *Above the Clouds: A Berkeley View of Cloud Computing. Technical Report.* University of California at Berkeley

[2] Balding C. (2008). *Assessing the Security Benefits of Cloud Computing.* Cloud Security Blog, available at http://cloudsecurity.org/blog/2008/07/21/assessing-the-securitybenefits-of-cloud-computing.html . Accessed 10th May, 2010.

[3] Boss, G., Malladi, P., Quan, D., Legregni, L., Hall, H. (2007), *Cloud Computing.* ww.ibm.com/developerworks/websphere/zones/hipods/. Retrieved on 20th May, 2010.

[4] Catteddu, D. and Hogben, G. (2009). *Cloud Computing: benefits, risks and recommendations for information security.* Technical Report. European Network and Information Security Agency.

[5] Creswell, J. W. (2007): *Qualitative inquiry and research design : choosing among five traditions*. 2nd ed., Sage Publications, Thousand Oaks, Calif.

[6] Fellowes, W. (2008). Partly Cloudy, Blue-Sky Thinking About Cloud Computing. Whitepaper. 451 Group.

[7] Gartner (2008). *Gartner Says Cloud Computing Will Be As Influential As E-business*. Gartner press release, 26 June 2008. http://www.gartner.com/it/page.jsp?id=707508. Retrieved 3rd May 2010

[8] Gleeson, E. (2009). *Computing industry set for a shocking change.* Retrieved May 10, 2010 from http://www.moneyweek.com/investment-advice/computing-industry-set-for-ashocking-change-43226.aspx

[9] Greenberg, A., Hamilton, J., Maltz, D. and Patel, P. (2009). *The Cost of a Cloud: Research Problems in Data Center Networks.* ACM SIGCOMM Computer Communication Review, 39, 1.

[10] Khajeh-Hosseini, A., Greenwood, D., Sommerville, I., (2010a). *Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS.* Submitted to IEEE CLOUD 2010

[11] Li, X., Li, Y., Liu, T., Qiu, J. and Wang, F. (2009). *The Method and Tool of Cost Analysis for Cloud Computing.* In IEEE International Conference on Cloud Computing (CLOUD-II 2009), Bangalore, India, September 2009, 93-100.

[12] Marshall, C. & Rossman, G.B., (1989), *Designing Qualitative Research*, Newbury Park, California: Sage.

[13] Vogels W, (2008). Beyond Server Consolidation. Queue 2008; (p 6:20–26). Available

at:http://portal.acm.org/citation.cfm?id=1348590&coll=Portal&dl=ACM&CFID=78225754&CFTOKEN=19192256&ret=1#Fulltext.

[14] Weishäupl T., Donno F., Schikuta E., Stockinger H., Wanek H. (2005). *Business In the Grid: The BIG Project.* In: Proceedings of the 2nd International Workshop on Grid Economics and Business Models (GECON 2005). http://hst.home.cern.ch/hst/publications/gecon-2005-BIGproject.pdf. Accessed 5th May, 2010.

[15] Preece, J., Rogers, Y. & Sharp, H., (2002). “Interaction Design - Beyond Human- Computer Interaction”, John Wiley & Sons.

# APPENDICES

**Appendix A: List of Publications**

**Appendix B: Co-Author's Statement**

# Appendix A. List of publications

The contributions have been or are in the process of being, validated through peer-review publication in journal, conference proceedings, and in book chapters are listed below

A. **Book Chapter :**

1. **Araddhana A. Deshmukh**, Pankaj Chaudhary , ,Dr. Albena Mihovska, Dr. Ramjee Prasad,' NUNI ( New User New Item) Problem for SRSs Using Content aware multimedia based Approach' Springer Book Science Business Media 2016 Innovations in Computer Science and Engineering Advances in intelligent systems and Computing 413 DOI 10.1007/978-981-10-0419-3_34, Corpus ID: 61759505 https://www.springerprofessional.de/en/nuni-new-user-and-new-item-problem-for-srss-using-content-aware-/7498956
2. **Aradhana A. Deshmukh**, Albena Mihovska , Ramjee Prasad, "Secure Outsourced Integrity Test For Cloud Storage", Advances in Intelligent Systems and Computing", 1077 ICT Systems and Sustainability, http://www.springer.com/series/11156, Proceedings of ICT4SD 2019, Volume 1, ISSN 2194-5357 ISSN 2194-5365 (electronic) Advances in Intelligent Systems and Computing ISBN 978-981-15-0935-3 ISBN 978-981-15-0936-0 (eBook), https://doi.org/10.1007/978-981-15-0936-0, © Springer Nature Singapore Pte Ltd. 2020, PP 309-320

B. **Journal Publications:**

1. **Araddhana A. Deshmukh**, Reshma Gade, Dr. Albena Mihovska, Dr. Ramjee Prasad, "Inference attack on URL visiting history of the social networking", International Journal of Control Theory and Applications., I J C T A, 9(10), 2016, pp. 1-11 © International Science Press, ISSN: 0974-5572, Volume :- No.10 (2017). Issue No :- 9 (2017). Pages :- 145-152
2. Aaradhana Deshmukh, *Reshma Gade*, Albena Mihovska and Ramjee Prasad. **Araddhana A. Deshmukh**, Smita Ashok Yeole, Dr. Albena Mihovska, Dr. Ramjee Prasad, "An Approach: A Novel Protected Data Deduplication Scheme with Dynamic Ownership Management to Secure Cloud Data", International Journal of Pure and Applied Mathematics, Volume 118 No. 24 2018, ISSN:1314-3395(on-line version),url: http://www.acadpubl.eu/hub/Special Issue,http://www.acadpubl.eu/hub/

3. **Araddhana A. Deshmukh,** Smita Ashok Yeole, Dr.Albena Mihovska, Dr. Ramjee Prasad," An Approach: A Novel Protected Data Deduplication Scheme with Dynamic Ownership Management to Secure CloudData International Journal of Pure and Applied Mathematics Volume 118 No. 24 2018ISSN: 1314-3395 (on-line version)url:http://www.acadpubl.eu/hub/Special Issue

C. **International conferences**

a. **FIRST AUTHOR**

1. **Araddhana A. Deshmukh,** Dr. Albena D. Mihovska, Dr. Ramjee Prasad, "Zoneminder as 'Software as a Service' and Load Balancing of Video Surveillance Requests", Networking and Electronic Commerce Research Conference 2012 (NAEC 2012), http://www.atsma.org/Conferences/NAEC2012,Italy,Riva del Garda Italy, October 11-14 2012
2. **Araddhana A. Deshmukh,** Dr. Albena D. Mihovska, Dr. Ramjee Prasad "Cloud computing security schemes:- TGOS [Threshold Group-Oriented signature ] and TMS [Threshold multisignature", 2 nd World Congress on Information and Communication Technology http://www.mirlabs.org/wict12/#cfp, WICT 2012, Trivandrum India, October 30-November 2, 2012 [ The paper also shortlisted by www.mirlabs.net/WICT 2012 for journal http://www.mirlabs.net/wict12/#journal .]
3. **Araddhana A. Deshmukh,** Dr. Albena D. Mihovska, Dr. Ramjee Prasad, "Optimized Resource Stipulation for Video on Demand Via Zoneminder", Global Wireless Summit 2014 in Aalborg from May 11-14, 2014 Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2014 4th International Conference on, Aalborg, Denmark, May 11-14, 2014, [Based on the GWS2014 paper we have been invited to submit a CONASENSE journal paper in the Special issue on "Advances in Control for the provision of Smart Services",DOI: 10.1109/VITAE.2014.6934492, https://ieeexplore.ieee.org/document/6934492
4. **Araddhana A. Deshmukh,** Pankaj Chaudhary ,Dr. Albena Mihovska, Dr. Ramjee Prasad,' NUNI ( New User New Item) Problem for SRSs Using Content aware multimedia based Approach' 3rd International Conference on Innovations in Computer Science & Engineering Hosted by: Guru Nanak Institutions In Association with CSI Hyderabad Chapter, Division 5, Education and Research, CSI –India Dates: 7th – 8th August 2015,

Venue: Guru Nanak Institutions, Hyderabad, India, Springer publication, ://www.springer.com/series/11156

5. **Araddhana A. Deshmukh,** Reshma Gade, Dr.Albena Mihovska, Dr. Ramjee Prasad," Inference attack on URL visiting history of the social networking", Springer International Conference on Intelligent Computing and Application, ICICA 2016, Department of computer Engineering, D.Y.Patil College of Engineering, Akurdi, Pune44
6. **Araddhana A. Deshmukh,** Dr.Albena Mihovska, Dr. Ramjee Prasad," Cloud Security using COSDACC [ Cluster of Servers and Access Control], IEEE 5G Summit and 30th GISFI Standardisation Series Meeting, ,SRM University, Kattankulathur, Chennai. 16-17 November 2017
7. **Araddhana A. Deshmukh,** Dr.Albena Mihovska, Dr. Ramjee Prasad" Secured Outsourced Integrity Test for Cloud Storage", Fourth International conference on Information and Communication Technology for Sustainable Development ( ICT4SD 2019) with World Publication and Conference Summit PubCon2019 Goa, India, as per the guidelines of Springer, 5-6 July 2019
8. **Aradhana A. Deshmukh,** Dr. G. S. Mundada, Dr. Albena Mihovska, " An Amendment of 3T (Technology,Trend, Target) using IT (Indian Talent) for Indian Engineering: A vital technocracy of Smart City", 34th Indian Engineering Congress, NOVOTEL Complex, Hyderabad, Dec 27-29, 2019 PP 325-331
9. **Aradhana A. Deshmukh**, Albena Mihovska , Ramjee Prasad," Anatomization and Perception of Mental Disorder Because Usage of Online Social Network Data", *"International Symposium on 5G & Beyond for Rural Upliftment"* BIT Sindri, IIT(ISM) Dhanbad, the IEEE 5G summit and 35th GISFI Standardization Series Meeting (GSSM) *on February 8-9.*2020

b. **SECOND AUTHOR**

1. Dipti Madankar , **Araddhana A. Deshmukh**, Dr. Albena Mihovska , Vaishali Maheskar , ' SQAMPS: Secure Quorum Architecture for Mobile Presence Services with End-to-End Encryption, WCI '15 Proceedings of the Third International Symposium on Women in Computing and Informatics , August 10-13, 2015, Kochi, India, ACM ISBN 978-1- 4-4503-3361-0/15/0
2. Shubhangi Opale , **Araddhana A. Deshmukh**, Dr. Albena Mihovska, Dr. Ramjee Prasad , "'Privacy-Preserving Friend Matching Protocol approach for Pre-match in Social

Networks", IEEE Global Wireless Summit 2015: Wireless Personal Multimedia Communication - Novotel Convention Centre, Hyderabad, Indien Varighed: 13 dec. 2015 → 16 dec. 2015 http://wpmc-symposium.org/2015/ WPMC 2015 18th International Symposium on Wireless Personal Multimedia Communications, Global Wireless Summit 2015, Hyderabad , Dec 13-16, 2015

3. Snehal Sawant , **Araddhana A. Deshmukh** ,Dr. Albena Mihovska, Dr. Ramjee Prasad ,"'Public Auditing and Data Dynamics in Cloud with Performance Assessment on Third Party Auditor'" WVITAE 2015, 5th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), IEEE Global Wireless Summit 2015: Wireless Personal Multimedia Communication - Novotel Convention Centre, Hyderabad, India Duration: 13 Dec 2015 → 16 Dec 2015 http://wpmc-symposium.org/2015/Global Wireless Summit 2015, Hyderabad , Dec 13-16, 2015

# Appendix B : Co-author's Statements

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh

This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | Zoneminder as 'Software as a Service' and Load Balancing of Video Surveillance Requests |
| Authors: | **Araddhana A. Deshmukh**, Dr. Albena Mihovska, Dr. Ramjee Prasad |

The article/manuscript is: Published ☐ Accepted ☐ Submitted ☐ In preparation ☐

1. If published, state full reference: http://www.atsma.org/Conferences/NAEC2012/NAEC2012_Program_and_Schedule.pdf

If accepted or submitted, state journal:
Has the article/manuscript previously been used in other PhD or doctoral dissertations?

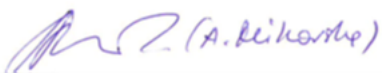
No ☑ ☐ Yes ☐ If yes, give details:

The PhD student has contributed to the elements of this article/manuscript as follows:

A. Has essentially done all the work
B. Major contribution
C. Equal contribution
D. Minor contribution
E. Not relevant

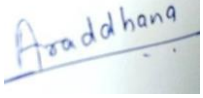
| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | C |
| 2. Planning of the experiments/methodology design and development | B |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | B |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 10/09/2012 | Dr. Albena Dimitrova *Mihovska* |  |
| 10/09/2012 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix

Date: 10/09/2012



_____
Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh

This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | Cloud computing security schemes:- TGOS [Threshold Group-Oriented signature ] and TMS [Threshold multisignature] |
| Authors: | **Araddhana A. Deshmukh**, Dr. Albena Mihovska, Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

1. If published, state full reference: https://ieeexplore.ieee.org/document/6409075

If accepted or submitted, state journal:
Has the article/manuscript previously been used in other PhD or doctoral dissertations?

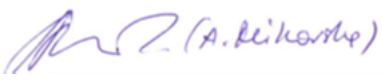
No √ ☐ Yes ☐ If yes, give details:

The PhD student has contributed to the elements of this article/manuscript as follows:

A. Has essentially done all the work
B. Major contribution
C. Equal contribution
D. Minor contribution
E. Not relevant

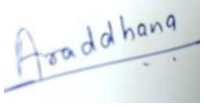
| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | B |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 10/11/2012 | Dr. Albena Dimitrova *Mihovska* |  |
| 10/11/2012 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix

Date: 23/10/11/2012



_____
Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh

This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | Optimized Resource Stipulation for Video on Demand Via Zoneminder |
| Authors: | **Araddhana A. Deshmukh**, Dr. Albena Mihovska, Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

1. If published, state full reference: https://ieeexplore.ieee.org/document/6934492

If accepted or submitted, state journal:
Has the article/manuscript previously been used in other PhD or doctoral dissertations?

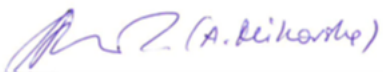
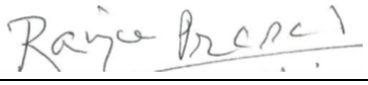
No √ ☐ Yes ☐ If yes, give details:

The PhD student has contributed to the elements of this article/manuscript as follows:

A. Has essentially done all the work
B. Major contribution
C. Equal contribution
D. Minor contribution
E. Not relevant

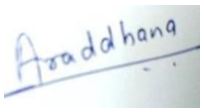
| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | C |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 18/05/2014 | Dr. Albena Dimitrova *Mihovska* |  |
| 18/05/2014 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix

Date: 18/05/2014



_____

Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh
This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | NUNI ( New User New Item) Problem for SRSs Using Content aware multimedia based Approach' |
| Authors: | **Pankaj Chaudhary , Araddhana A. Deshmukh**, Dr. Albena Mihovska, Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

1. If published, state full reference: https://link.springer.com/chapter/10.1007/978-981-10-0419-3_34
   If accepted or submitted, state journal:

Has the article/manuscript previously been used in other PhD or doctoral dissertations?

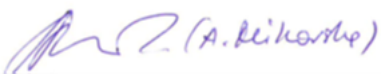
No √ ☐ Yes ☐ If yes, give details:

The PhD student has contributed to the elements of this article/manuscript as follows:

A. Has essentially done all the work
B. Major contribution
C. Equal contribution
D. Minor contribution
E. Not relevant

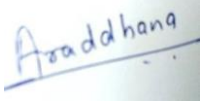
| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | C |
| 2. Planning of the experiments/methodology design and development | C |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 10/08/2015 | Pankaj Chaudhary |  |
| 10/08/2015 | Dr. Albena Dimitrova *Mihovska* |  |
| 10/08/2015 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix

Date: 10/08/2015



_____
Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh

This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | Inference attack on URL visiting history of the social networking |
| Authors: | **Araddhana A. Deshmukh**, Reshma Gade, Dr. Albena Mihovska, Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

1. If published, state full reference:
https://www.serialsjournals.com/index.php?route=product/product/volumearticle&issue_id=185&product_id=365
If accepted or submitted, state journal:
Has the article/manuscript previously been used in other PhD or doctoral dissertations?

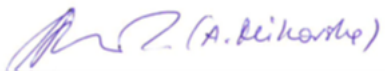
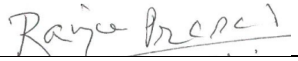
No √ ☐ Yes ☐ If yes, give details:

The PhD student has contributed to the elements of this article/manuscript as follows:

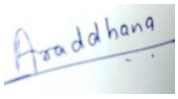
A. Has essentially done all the work
B. Major contribution
C. Equal contribution
D. Minor contribution
E. Not relevant

| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | B |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 10/01/2017 | Reshma Gade |  |
| 10/01/2017 | Dr. Albena Dimitrova *Mihovska* |  |
| 10/01/2017 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix
Date: 10/01/2017



_____
Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh

This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | Cloud Security using COSDACC [ Cluster of Servers and Access Control] |
| Authors: | **Araddhana A. Deshmukh**, Dr. Albena Mihovska, Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

1. If published, state full reference: https://ieeexplore.ieee.org/document/6934492
If accepted or submitted, state journal:
Has the article/manuscript previously been used in other PhD or doctoral dissertations?

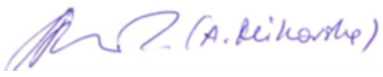
No √ ☐ Yes ☐ If yes, give details:

The PhD student has contributed to the elements of this article/manuscript as follows:

A. Has essentially done all the work
B. Major contribution
C. Equal contribution
D. Minor contribution
E. Not relevant

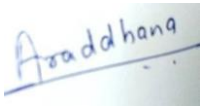
| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | C |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 11/01/2018 | Dr. Albena Dimitrova *Mihovska* |  |
| 11/01/2018 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix

Date: 11/01/2018



_____
Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh

This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | An Approach: A Novel Protected Data Deduplication Scheme with Dynamic Ownership Management to Secure Cloud Data |
| Authors: | **Araddhana A. Deshmukh**, Dr. Albena Mihovska, Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

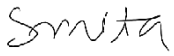
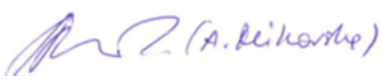
1. If published, state full reference: https://www.gisfi.org/news_meeting_details.php?id=125&mno=31th
If accepted or submitted, state journal:
Has the article/manuscript previously been used in other PhD or doctoral dissertations?

No √ ☐ Yes ☐ If yes, give details:
The PhD student has contributed to the elements of this article/manuscript as follows:

A. Has essentially done all the work
B. Major contribution
C. Equal contribution
D. Minor contribution
E. Not relevant

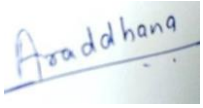
| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | C |
| 3. Involvement in the experimental work/clinical studies/data collection | C |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 18/09/2018 | Smita Yeole |  |
| 18/09/2018 | Dr. Albena Dimitrova *Mihovska* |  |
| 18/09/2018 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix

Date: 18/05/2014



_____
Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh
This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | Secured Outsourced Integrity Test for Cloud Storage |
| Authors: | **Araddhana A. Deshmukh**,Dr. Albena Mihovska, Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

1. If published, state full reference: - https://pure.au.dk/portal/da/publications/secure-outsourced-integrity-test-for-cloud-storage(ee08f3b3-b6b7-4803-85b0-2c386f241181).html

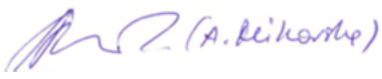
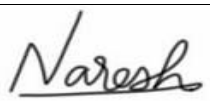
If accepted or submitted, state journal:
Has the article/manuscript previously been used in other PhD or doctoral dissertations?

No √ ☐ Yes ☐ If yes, give details:
The PhD student has contributed to the elements of this article/manuscript as follows:

A. Has essentially done all the work
B. Major contribution
C. Equal contribution
D. Minor contribution
E. Not relevant

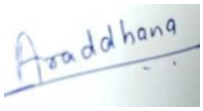
| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | B |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | C |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 18/09/2019 | Dr. Albena Dimitrova *Mihovska* |  |
| 18/09/2019 | Dr. Ramjee Prasad |  |
| 18/09/2019 | Naresh Gardas |  |

In case of further co-authors please attach appendix

Date: 23/05/2020



_____
Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh

This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | An Amendment of 3T (Technology,Trend, Target) using IT (Indian Talent) for Indian Engineering: A vital technocracy of Smart City |
| Authors: | **Araddhana A. Deshmukh**, Dr. G.S.Mundada, Dr. Albena Mihovska, Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

1. If published, state full reference: -

If accepted or submitted, state journal:

Has the article/manuscript previously been used in other PhD or doctoral dissertations?

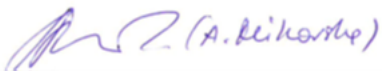
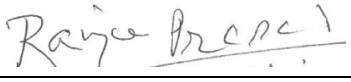
No √ ☐ Yes ☐ If yes, give details:

The PhD student has contributed to the elements of this article/manuscript as follows:

A. Has essentially done all the work
B. Major contribution
C. Equal contribution
D. Minor contribution
E. Not relevant

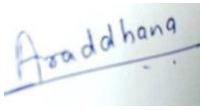
| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | C |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 18/09/2019 | Dr. G.S.Mundada |  |
| 18/09/2019 | Dr. Albena Dimitrova *Mihovska* |  |
| 18/09/2019 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix

Date: 23/05/2020



_____

Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh

This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | Anatomization and Perception of Mental Disorder Because Usage of Online Social Network Data" |
| Authors: | **Araddhana A. Deshmukh**, Dr. Albena Mihovska, Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

1. If published, state full reference :- International Symposium on 5G & Beyond for Rural UpliftmentJointly with BIT Sindri, IIT(ISM) Dhanbad, the IEEE 5G summit and 35th GISFI Standardization Series Meeting (GSSM)February 8-9, 2020Theme: 5G and Beyond for Rural Upliftmen

If accepted or submitted, state journal:

Has the article/manuscript previously been used in other PhD or doctoral dissertations?

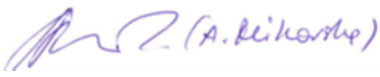
No √ ☐ Yes ☐ If yes, give details:

The PhD student has contributed to the elements of this article/manuscript as follows:

A. Has essentially done all the work
B. Major contribution
C. Equal contribution
D. Minor contribution
E. Not relevant

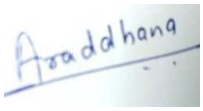
| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | C |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 18/05/2020 | Dr. Albena Dimitrova *Mihovska* |  |
| 18/05/2020 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix

Date: 23/05/2020



_____

Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh
This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | Anatomization and Perception of Mental Disorder Because Usage of Online Social Network Data" |
| Authors: | **Araddhana A. Deshmukh**, Dr. Albena Mihovska, Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

2. If published, state full reference :- International Symposium on 5G & Beyond for Rural UpliftmentJointly with BIT Sindri, IIT(ISM) Dhanbad, the IEEE 5G summit and 35th GISFI Standardization Series Meeting (GSSM)February 8-9, 2020Theme: 5G and Beyond for Rural Upliftmen

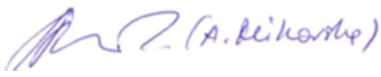
If accepted or submitted, state journal:
Has the article/manuscript previously been used in other PhD or doctoral dissertations?

No √ ☐ Yes ☐ If yes, give details:
The PhD student has contributed to the elements of this article/manuscript as follows:

F. Has essentially done all the work
G. Major contribution
H. Equal contribution
I. Minor contribution
J. Not relevant

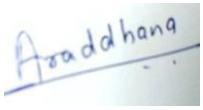
| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | C |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 18/05/2020 | Dr. Albena Dimitrova *Mihovska* |  |
| 18/05/2020 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix

Date: 23/05/2020



_____
Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh

This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | 'Privacy-Preserving Friend Matching Protocol approach for Pre-match in Social Networks |
| Authors: | Shubhangi Opale, **Araddhana A. Deshmukh**, Dr. Albena Mihovska , Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

1. If published, state full reference http://wpmc-symposium.org/2015/
   If accepted or submitted, state journal:

Has the article/manuscript previously been used in other PhD or doctoral dissertations?

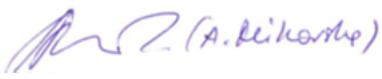
No √ ☐ Yes ☐ If yes, give details:

The PhD student has contributed to the elements of this article/manuscript as follows:

A. Has essentially done all the work
B. Major contribution
C. Equal contribution
D. Minor contribution
E. Not relevant

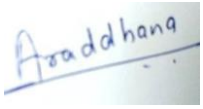
| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | B |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 20/12/2015 | Shubhangi Opale |  |
| 20/12/2015 | Dr. Albena Dimitrova *Mihovska* |  |
| 20/12/2015 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix

Date: 23/05/2020



_____

Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

**Declaration of co-authorship***

Full name of the PhD student: Araddhana Arvind Deshmukh
This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | 'Public Auditing and Data Dynamics in Cloud with Performance Assessment on Third Party Auditor' |
| Authors: | Snehal Sawant, **Araddhana A. Deshmukh**, Dr. Albena Mihovska , Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

1. If published, state full reference :- http://wpmc-symposium.org/2015/
   If accepted or submitted, state journal: http://wpmc-symposium.org/2015/

Has the article/manuscript previously been used in other PhD or doctoral dissertations?

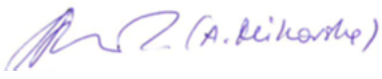
No √ ☐ Yes ☐ If yes, give details:

The PhD student has contributed to the elements of this article/manuscript as follows:

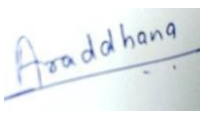
- Has essentially done all the work
- Major contribution
- Equal contribution
- Minor contribution
- Not relevant

| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | B |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 20/12/2015 | Snehal Sawant |  |
| 20/12/2015 | Dr. Albena Dimitrova *Mihovska* |  |
| 20/12/2015 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix
Date: 23/05/2020



_____
Signature of the PhD student

AARHUS BSS SCHOOL OF BUSINESS AND SOCIAL SCIENCES
AARHUS UNIVERSITY

## Declaration of co-authorship*

Full name of the PhD student: Araddhana Arvind Deshmukh
This declaration concerns the following article/manuscript:

| | |
|---|---|
| Title: | 'Public Auditing and Data Dynamics in Cloud with Performance Assessment on Third Party Auditor' |
| Authors: | Snehal Sawant, **Araddhana A. Deshmukh**, Dr. Albena Mihovska , Dr. Ramjee Prasad |

The article/manuscript is: Published √ Accepted ☐ Submitted ☐ In preparation ☐

2. If published, state full reference :- http://wpmc-symposium.org/2015/
If accepted or submitted, state journal: http://wpmc-symposium.org/2015/

Has the article/manuscript previously been used in other PhD or doctoral dissertations?

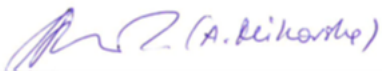
No √ ☐ Yes ☐ If yes, give details:

The PhD student has contributed to the elements of this article/manuscript as follows:

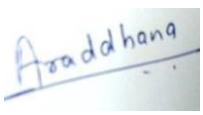
- Has essentially done all the work
- Major contribution
- Equal contribution
- Minor contribution
- Not relevant

| Element | Extent (A-E) |
|---|---|
| 1. Formulation/identification of the scientific problem | B |
| 2. Planning of the experiments/methodology design and development | B |
| 3. Involvement in the experimental work/clinical studies/data collection | B |
| 4. Interpretation of the results | C |
| 5. Writing of the first draft of the manuscript | A |
| 6. Finalization of the manuscript and submission | B |

**Signatures of the co-authors**

| Date | Name | Signature |
|---|---|---|
| 20/12/2015 | Snehal Sawant |  |
| 20/12/2015 | Dr. Albena Dimitrova *Mihovska* |  |
| 20/12/2015 | Dr. Ramjee Prasad |  |

In case of further co-authors please attach appendix
Date: 23/05/2020



_____
Signature of the PhD student