PhD Candidate (GISFI 2nd Batch) : Aaradhana Arvind Deshmukh

Title of thesis **:** Green Cloud Computing : Balancing load for the deployment of SaaS [Software as a Service]

Supervisor **:** Dr. Albena Dimitrova Mihovska

.Co-Supervisor **:** Dr. Ramjee Prasad

AAU PhD Degree

PhD study Plan

Title:

Green Cloud Computing : Balancing load for the deployment of SaaS [Software as a Service]

PhD Candidate (GISFI 2nd Batch)

Aaradhana Arvind Deshmukh

(aad@es.aau.dk)

Supervisor

Dr. Albena Dimitrova Mihovska

CTiF, AAU, Denmark

Co-Supervisor

Dr. Ramjee Prasad

CTiF, AAU, Denmark

AALBORG UNIVERSITET

PhD Study Plan:

Doctoral Programme	**:**	Wireless Communications
Study plan 2 months	**:**	☐
Study plan 11 months	**:**	☒
Project title	**:**	Green Cloud Computing : Balancing Load for the Deployment of SaaS [software as Service].
Name of PhD student	**:**	Aaradhana Arvind Deshmukh
Email	**:**	**aad@es.aau.dk**
Supervisor	**:**	Dr. Albena Dimitrova Mihovska
Co-supervisor	:	Dr. Ramjee Prasad
Department	**:**	Department of Electronics System
Date of enrolment	**:**	15th March 2011
Expected date of Completion	**:**	14th March 2014

Signature

01/04/2012 _____
Date PhD student [Aaradhana Arvind Deshmukh]

Study plan approved

_____ _____
Date Supervisor Albena Dimitrova Mihovska

_____ _____
Date Head of Doctoral Programme Gert Frølund Pedersen

Table of Contents

Green Cloud Computing : Balancing load for the deployment of SaaS [Software as a Service]

1 Research work Summary/Abstract :

Due to usage and expansion of web applications, manipulations like online banking, online shopping, e-commerce and various service web sites faced with more dynamic requests causes ever-increasing server resource load and overhead. Software as a service is a newly emerging computing paradigm, where in remote and virtualized computing resources are utilized by users by means of software access on 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 single server is no longer able to balance growing number of service requests, a more expensive server with better performance 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.

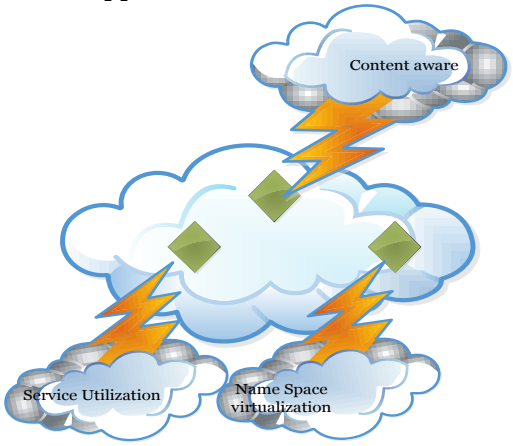

Figure 1 Proposed issues related to load balancing

For the past decades in computer networking many load balancing issues have been known. 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. As shown in Figure 1 this research will bring about with the help of simulation and using mathematical and analytical modeling 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 has combine properties of threshold group-oriented signature schemes and multi-signature scheme. Therefore to fully ensure the data security and save cloud user's computation resources, it is of critical importance to collaborate and contribute equally to produce a valid multiparty signature.

2 The scientific content of the PhD project:

A Background of the project

Cloud computing is integrating personal computers into a single global computer. The term cloud computing stands for type of service that allows any organization to deliver an application or service to their employee, customer i.e end user through Internet.
There is a great analogy between the shift from the concept of traditional computers to cloud computers and what YouTube has done. In the past, when we wanted to play a video, we had to download the video from its source and convert it into a format that was compatible with our personal computers. Then, YouTube came into the picture. Using YouTube we can watch a video directly online, no downloading or conversions are needed. Cloud computing uses the same concept; by storing programs, memory and applications to one main computer that acts as a network HUB and networking sub computers with the main computer. "Cloud computing" is a term, which involves virtualization, distributed computing, networking, software and web services. A cloud consists of several elements such as clients, data centre and distributed servers. It includes fault tolerance, high availability, scalability, edibility, reduced overhead for users, reduced cost of ownership, on demand services etc.
Central to these issues lies the establishment of an effective load balancing algorithm. The load can be CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal [1].
To summarize, instead of installing the program or the application on each individual computer, these applications are provided on the cloud. Users only need an application that allows them to log on to the cloud. Then, they can use the programs and applications as if they were a part of their own computers. Consequently, millions of users could benefit from this program, as they won't need to install any programs at all.
This research extracts four basic concepts of SaaS i.e. Software as a Service as a basis of the green cloud :-

1. Multitenancy :- System deliver application to multiple client organization from single instance of software
2. Provisioning :- It is an automated process that manages configuration, computing resources and processes and optimized availability
3. Scalability :- System scale storage needs to reduces user cost and complexity
4. Billing Module: - A set of predefined on demand policies.

Load balancing algorithm ensures that every computing resource will be distributed efficiently and in the end improves 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 distribution of requests, maintains technical performance metrics. It also provides alerts to quickly notify administrators when any problem may occur.

B State-of-the-art for the PhD project

- **Existing Load balancers are as follows :-**
  1. **DNS load balancer: -** Is distribution of requests among different servers through resolving domain name to different IP address of servers.
  2. **Database Load Balancer: -** Is balance database access requests among clusters of database servers, to achieve database scalability and high availability.
  3. **Switch based Load Balancer: -** Is a balancer which mainly relies on ASIC (Application Specific Integrated Circuit) chips to perform packet re-writing function.

- **Existing Load balancing algorithms:-**

A specific need dependent algorithm is needed for distributing the traffic among a group of servers or clusters [2]. With the help of various performance metrics as shown in Figure 2, all algorithms are programmed into server load balancers. It can have conjunctions with any of the persistence dynamic load balancing method. They are also assigned to individual virtual IPs.

Practically there are two broad classes of load balancing schemes :

1. Static Load balancing scheme :-

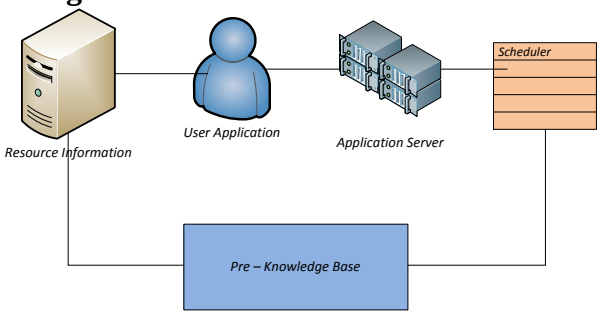

Figure 2 Static load balancing

Figure 2 shows the 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 [3,4].

The goal of the static load balancing method is to reduce the overall execution time of a concurrent program while minimizing the communication delays. 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.

Following concepts visualized comparing with dynamic scheme :-

- Decisions are made before execution.
- More complex than dynamic scheme
- Overhead always comes before **execution**

2. Dynamic load balancing scheme [5,6]:-

As shown in Figure 3 the dynamic load balancing scheme differs from the static algorithms in that 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

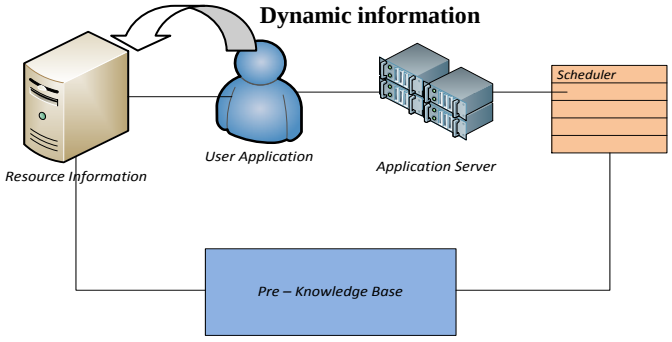

Figure 3 Dynamic load balancing

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.

Following concepts visualized comparing with static schemes :-

- Decisions are real time.
- Less Complex than static scheme
- Overhead depend on real-time assignment.

So based on above comparison between the static and dynamic schemes and with the help of following two main issues the research focuses on dynamic scheme.

- Decisions on load balancing are based on current state of system.
- No prior knowledge is needed need

Dynamic scheme categorized into following scenarios :-

I) Stream based scenario:-

It has been realized as a layered approach. Network level is responsible for a particular operation. It contains series of operation and each operation can be performed at one level in the network. Often, it is difficult to predict, because the user's requests are not fixed which executed in shortest time and average latency [5].

II) Cloud computing scenario:-

A threshold exists in the balancer which controls the requests at which a node should segregate some of its work to other node. The threshold determines the load and work for distribution as well as re-distribution of tasks. The smaller the threshold is, the more balanced the system. Bottleneck elimination strategy was also introduced in which it is used to balance the load within each stage of execution [6,7].

III) Discrete event scenario :-

It divides the each complex computational region into spatially equal pieces. Here, there is no centralized controller. So each node has a threshold that lets it judge whether it is lightly or heavily loaded. The random destination algorithm is particularly effective. Currently many virtual machines operate this scenario [8].

When we are having a scalable, ubiquitous application in the cloud, then we choose a suitable load balancing solution as shown below project flow in Figure no 3. In that way, we can spread the workload across many cloud machines. Then it is necessary to evaluate a potential solution from the point of view of both cost and performance.

The Green cloud flow has been defined as shown in Figure 4 :

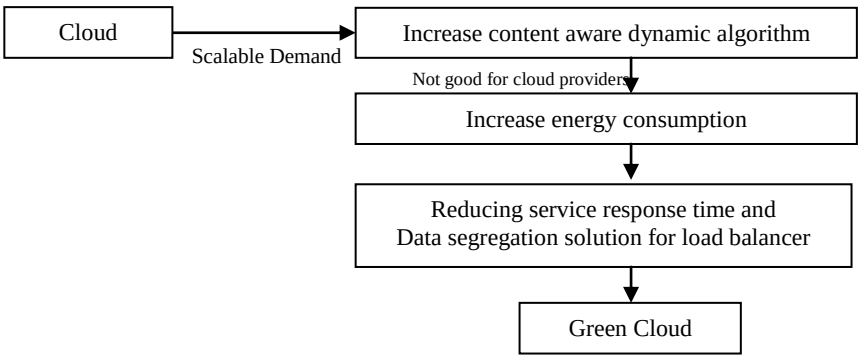

Figure 4 : Project flow

- **Research focuses on following five Load Balancing Performance Metrics [LBPM]in existing cloud systems [9-14,16,17,18-26]:-**
  1. **LBPM 1. Throughput: -** Number of tasks execution calculation.
  2. **LBPM 2. Performance: -** Efficiency of system.
  3. **LBPM 3. Scalability: -** For finite no. of nodes it is an ability of an algorithm to balance a load.
  4. **LBPM 4. Latency :-** Time delay between packet communication
  5. **LBPM 5. Security: -** Server always recommended firewall by allowing only traffic that is destined for desired services.

  The goal of the research is to determine load balancing mechanism distributes dynamic workload evenly across all nodes. It is always used to achieve high user satisfaction and utilization ratio, while ensure that not a single node will be overloaded hence improving the overall performance of the system [15].

Therefore for future research, the dynamic content aware approach was selected for coverage based on the following reasoning :-

- **Various techniques / algorithms are developed for Load balancing which are based on the above performance metrics as follows :-**

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.

Table 1 :- Comparative survey of load balancing algorithms

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 provide increased, energy efficiency, performance for wired and wireless network operation centres.

C Project's objectives

The main research objective is to propose and develop ubiquitous strategy for load balancing, useful to scale different types of applications.

I) Ist Year objectives :-

The first year research had the following objectives :

- Study and complete a literature survey for the strategy of content aware load balancing issues on cloud network. A part of the research it has been proposed to deploy an application on a cloud with efficient improved scalability, throughput, latency, which can fulfil the requirements of Green ICT.
- Study the technical , commercial as well as functional aspects of SAAS for well defined cloud application addressing issues such as :
  - Performance of application
  - User response.

➢ Work completed :-

I) Focus was on identifying the main bottleneck problem. For this purpose, I surveyed various load balancing algorithms. The main bottleneck problem in a static load balancing algorithm is based on gathering more information of clients, requests and server side which limits the system scalability.

The following are observations for load balancing based on performed work :-

Managing accessibility for application performance is a key challenge for QoS [1]. 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:

a) Scaling
b) Scheduling
c) Load balancing

a) Scaling

Scaling 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 load for each virtual machine and if there is with virtual machine processes running in it, shutdown the particular virtual machine.

For each instance i in $N_{VirtualResources}$
If $(NUsers_i == 0)$
Shut down the instance
Decrement $N_{VirtualResources}$
End for

b) Scheduling

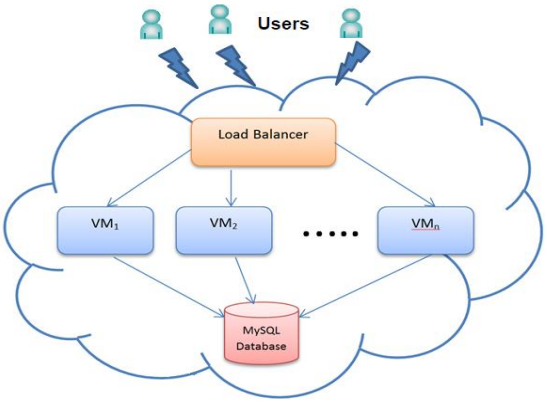

Figure 5 Architecture to balance load on virtual machine

Scheduling is used 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.

As shown in Figure 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.

c) Load balancing

Based on state of art and conclusions of literature survey, the dynamic content aware load balancing was chosen as the focus of the studies of this work. The results of this part of research are being submitted as a journal paper to the Springer Journal of Wireless Personal Communications.

II) Security is another challenge for cloud computing. Research focused on combining load balancing based security requirements for any application. A security algorithm was proposed combining two security properties i.e. threshold multisignature and threshold group oriented signature. Currently, the conceptual findings have been included in an intended publication to be submitted to the International Wireless Personal Communications Symposium (WPMC) 2012.

II) IInd Year objectives :-

The research for the IInd year studies is focused on the following:

1) Scale performance of server based tasks to calculate service utilization issue.
2) Automatically detect failure of server and repartitioning of the client traffic like redundancy. Aggregate HTTP requests and then distribute traffic [segregation] with content aware dynamic requests.
3) Define classification methods of load balancing for name space virtualization
4) Formation of mathematical model for calculating two kind of scenarios
5) Content aware request received by server based on probability conditions.
6) Replying to client with checking availability in server farm where the actual name space will be assign.
7) It is planned to publish the expected results in one journal and two conference papers.

The proposed research scenario is as follows: -

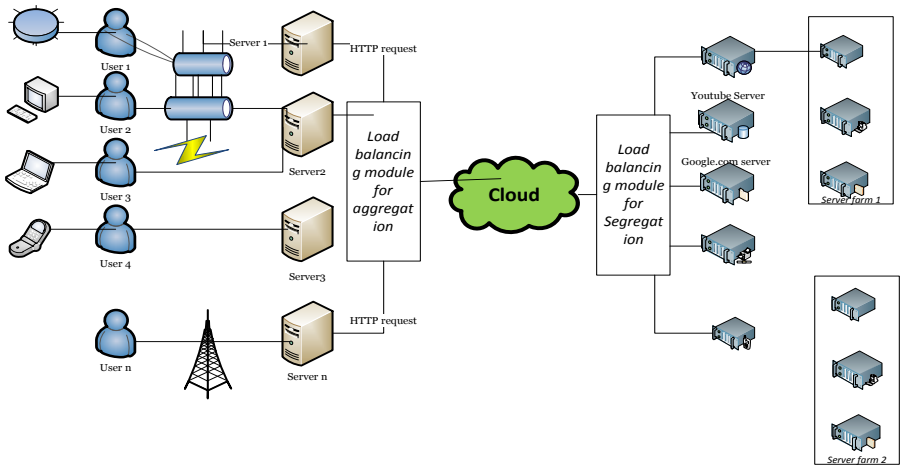

Figure 6 Content aware secure load balancing model

As shown in Figure 6 the load balancer has two major sides of control, i.e. client to server 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.

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 fulfil the execution of content aware request.

D Key methods

The proposed research 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 Software as a Service. The cloud has a virtualized environment for running various kinds of applications. The decision of resource allocation is taken by the cloud environment itself.

The research is based on theory-assisted design and application to practical situations. Powerful network simulators like jmeter and opnet are available to design and analyse 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's network. And Apache JMeter is an Apache project that can be used as a load testing tool for analyzing and measuring the performance of a variety of services, with a focus on web applications [29] . So testing preliminary level of load testing preference is given to jmeter. But to proceed further opnet will provide better performance results.

E Projects expected outcome

The expected outcome of the research is to develop Green cloud (load balancing strategy) would include

- A working cloud base deploying application from different sites and return the result.
- Data pre-processing service.
- Develop the test cases for implemented application based content aware on the parameters scalability, User response time, data segregation.
- Develop new content aware dynamic load balance techniques to make application more efficient.
- Implement new load balancing mechanism.

F Time schedule with milestones

Task		Year 1				Year 2				Year 3			
		March 2011-12				March 2012-13				March 2013-14			
1	Background Study												
2	Literature survey												
3	Requirement gathering and analysis.												
4	Novel concept development and Problem Specification and Delineation												
5	Framework design and challenges												
6	Implementation												
7	Simulation												
8	Performance analysis and optimization												
9	Result, conclusion, dissemination of the PhD study.												
10	Attending PhD courses*												
11	Papers and Conferences												
12	Writing of the Thesis												
13	Stay Abroad **												

* 3 months at AAU and 9 months in India every year.

** I will attend courses organized by Aalborg University through video conferencing and courses related to my research area in well known Institutes in India.

These are the milestones planning to achieve as part of my PhD with tentative months.

- **Milestone 1 (M1):** Understanding load balancing algorithms and their issues
- **Milestone 2 (M2):** Determining motivations, questions and research challenges, direction and problem specifications.
- **Milestone 3(M3):** Defining the functional requirements, which will cover detailed study

of security schemes. . Survey paper will be published in conference and journal.

- **Milestone 4(M4):** Analysis of performances issues in load balancing mechanism
- **Milestone 5 (M8):** Design of content aware traffic distribution algorithm.
- **Milestone 6 (M12):** Evaluating phases, defining performance metrics and comparison with existing works and writing conference/journal paper.
- **Milestone 7 (M14):** To do few courses during this 6 month period and submission of Journal Papers. To develop the secure load distribution assignment technique and to publish it in journal or conference.
- **Milestone 8 (M16):** Studying and understanding literature of authentication methods, developing algorithm for authentication and privacy, Implementation.
- **Milestone 9 (M20):** Analyzing the transmission control mechanisms, to develop the cluster based virtualized mechanism and to check its effect by simulation.
- **Milestone 10 (M24):** To develop a data intensive based computing for load balance.
- **Milestone 11 (M26):** Submission of Journal Paper based on multidimensional resource constraints.
- **Milestone 12 (M28):** To start writing thesis and simultaneously do refinements towards finishing PhD, by the end of M26.
- **Milestone 13 (M32):** Write up the PhD work, and publish all the work done.

G Outline of Content of Thesis

Abstract
Preface
Acknowledgements
Publications
Contribution to Published Papers
List of Abbreviations
List of Figures
Chapter 1: Introduction

1.1	Motivation and Objective	
	1.1.1	Design Criteria for load balancing on cloud
	1.1.2	SAS deploy and evaluate
1.2	Project Formulation and Challenges	
1.3	Solutions	
	1.3.1	Content aware dynamic load balancing
	1.3.2	Load balanced strategy
1.4	Contribution of Thesis	
1.5	Overview of Thesis	

Chapter 2: Literature Review

2.1	Load balancing	
	2.1.1	Introduction
	2.1.2	Layer application
2.2	Software as service	
	2.2.1	Evaluation for load balancing
	2.2.2	Profile based approach
	2.2.3	Schedule Based task distribution
	2.2.4	Strategy for Network Operation Centres
	2.2.5	Comparison of load balancing issues
2.3	**Virtualization**	
	2.3.1	Virtualization for energy efficiency
	2.3.2	Effects of energy efficiency
	2.3.4	Virtualization for name space
Chapter 3	**Strategy and Methodology**	
	3.1	Strategy modelling
	3.2	Load balancing performance issues
	3.3	Load balancing performance methodology
Chapter 4	**Algorithms**	
	4.1	Strategy modelling algorithms
	4.2	Load balancing performance issues
	4.3	Load balancing performance methodology

	4.4	Comparative study of algorithms
	4.5	Simulation and results
Chapter 5	**Scalability architectures :- Green Cloud**	
	5.1	Terminology and concepts
	5.2	How SaaS can be used for PaaS,Iaas
	5.3	SaaS deployment :- March towards Green Cloud
Chapter 6	**Cloud Security**	
	6.1	Introduction
	6.2	Security mechanism :- Threshold group oriented signature and multi-signature scheme
	6.3	Identity and Access Management
	6.4	Future aspects
Chapter 7:	**Conclusions and Future Work**	
	7.1	Summary of Contributions
	7.2	Future Work
	7.3	Concluding Remark
	References	

H Tentative Publications Titles

1. Aaradhana Deshmukh, Dr. Albena Mihovska, Dr. Ramjee Prasad " Content aware load balancing dynamic algorithm"
2. Aaradhana Deshmukh, Dr. Albena Mihovska, Dr. Ramjee Prasad "Cloud computing security schemes: TGOS [Threshold Group Oriented Signature] and TMS [Threshold Multi signature scheme] ".
3. Aaradhana Deshmukh, Dr. Albena Mihovska, Dr. Ramjee Prasad "Green Cloud computing : Virtualized Load balance "
4. Aaradhana Deshmukh, Dr. Albena Mihovska, Dr. Ramjee Prasad "Green-Cloud Computing An Resource utilization challenges in processing and storage
5. Aaradhana Deshmukh, Dr. Albena Mihovska, Dr. Ramjee Prasad "Effect of virtualization on green cloud"
6. Aaradhana Deshmukh, Dr. Albena Mihovska, Dr. Ramjee Prasad "Cloud Computing futuristic approach"

3 Agreement on the relationship between supervisor and student

Roles

- The student and the supervisors are together responsible for time management in the project. Time plan for the Ph.D. study should be reviewed every six months.
- Supervisor, Co-supervisor (India) will provide technical assistances and supervision.
- The student should be able to get access to lab equipment and technical assistance from both AAU and STES. In cases when advanced equipments are required, the student should make a request at least one month in advance and the supervisors should help the student as much as they can.

Type of collaboration

- Telephone conferences every second week of the month between the student, the AAU and INDIAN supervisors.
- Workshops every six months.
- Minutes will be made for telephone conferences and workshops.
- Feedback regarding the progress and quality of work will be given during the meetings, conferences and workshops.

Supervision meetings

- Most meetings are scheduled and arranged jointly by the student and the supervisors. In case of special needs, both student and supervisors can call for a meeting.
- Agenda will be provided by the student one day prior to each meeting.
- Common documents will be distributed and maintained via e-mail and/or AFS servers at AAU.
- Every year 3 months PhD Student will be in direct contact with Supervisors at AAU and telephonic and video conferencing meetings with Indian co-supervisor.

Remaining 9 months PhD Student will be in direct contact with Indian Co-supervisor and Telephonic and video conferencing with AAU Supervisors.

Workplace participation

- The student is involved in group activities at AAU and STES both.
- The student will frequently meet with research group at AAU and renowned Institutes and multinational companies in India
- Group meetings at STES are usually organized once in a year.

Development plans

- Building Professional network

Writing papers

- Paper writing is on the basis of collaboration between the student and the supervisors. In most of cases, the student prepares the first draft and the supervisors give feedback and comments timely.
- The student will present his/her work at biannual workshops.
- Characteristics of and expectation to the research
- Novel Ideas towards standardizations and patents
- Developing the cooperation and updating the agreement
- This agreement will be evaluated every six months.

4 Plan for PhD courses *

Courses	Place/ Organizer	ECTS	General or Project course	Status
Design choices and tradeoffs in computer system	Prof.Bruce Shriver	3	Project	Completed
Distributed Source Coding and Multiple Descriptions	Jan Østergaard	3	Project	Completed
Performance evaluation of heterogeneous dynamic communication network	Hens Peter Scwefel	2.5	Project	Completed
Bayesian Statistics, Simulation And Software - With A View To Application Examples	Kasper K. Berthelsen, Associate Professor	3	Project	Completed
Analysis and Design of High Performance Future Internet Infrastructure	Jens Myrup Pedersen, M.Tahir Riaz	2	Project	Completed
Management of Research and Development	Professor Frank Gertsen	2.5	General	Completed
Sensors and RFID Networks	Neeli Rashmi Prasad	2.5	Project	Planned
Network Coding: Theory and Applications	Professor Muriel Medard, Frank Fitzek	4	Project	Planned
Advanced Topics in IT Security	Associate Professo Rene Rydhof Hansen	2	Project	Planned
Writing and Reviewing Scientific Papers	Professor Jakob Stoustrup	3.75	General	Planned
Intellectual Property Rights	Morten Dahlgaard Andersen	2	General	Planned
Subtotal (Planned& Registered)		**13.5**		
Total (Completed)		**17.5**		
Total		**31**		

- Based on the PhD Courses Catalogue and this list may change according to the courses available in India.

5 Plan for dissemination of knowledge and findings from the project

Publication Plan:
I will publish papers on my research work in the following conferences and Journals

Sr. No	Name of the Conference and Journals	Conference dead line	Conference Dates	Location	Conference Web Site
1.	WPMC'12, 15th International Symposium on Wireless Personal	May 18,2012	Sept 24-27,2012	Taipei, Taiwan	http://wpmc2012.ntu.edu.tw/

	Multimedia Communication				
2.	SNDS – 2012 International Conference on Security in Computer Networks and Distributed Systems	June 28,2012	Oct 11-12, 2012	Thiruvanathpuram, India	http://www.snds-conference.org/callpaper.html
3.	Wireless VITAE 2013	Sept 2012	June 24-26,2013	Princeton, New Jersey, USA	http://www.wirelessvitae.org/2013/
4.	Wireless Personal Multimedia Communications	2013	Journal	Springer	http://www.springer.com/engineering/signals/journal/11277
5.	IEEE Communication Society Journals	2013	Journal	IEEE	http://www.comsoc.org/publications/journals
6.	IEEE/ACM Transactions on Communication and Networking	2013	Transactions	IEEE/ACM	http://www.ieee.org http://www.acm.org

6 Agreements on immaterial rights to patents, etc. produced during the PhD project

All the rights about patents obtained during the PhD course will be shared between the university and the PhD student, following the standard procedures in AAU.

7 Plans for external collaboration

In connection with the project work the candidate will stay with Sinhgad Technical Education Society in India, except for two 3 month stays at Aalborg University during the first two years.

8 Financing budget for the PhD project

- CTIF, Aalborg University will provide the research facility.
- Expenses for tuition fee, lodging, boarding and travelling will be borne by Sinhgad Technical Education Society, Pune, India

9. Short 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 Project 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 com puter," 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] Trieu C. Chieu, Ajay Mohindra, Alexei A. Karve and Alla 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] E. Knorr, "Software as a Service: The Next Big Thing",http://www.infoworld.com/article/06/03/20/76103_12FEsaas_1.html.
[19] Jyoti Namjoshi, Archana 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