

Development of Mathematical Model for Market-Oriented Cloud Computing

K.Mukherjee

Department of Computer Science &
Engineering.
Birla Institute of Technology
Mesra, Ranchi, India

G.Sahoo

Department of Information
Technology
Birla Institute of Technology
Mesra, Ranchi, India

ABSTRACT

Cloud computing is an emerging approach to a shared infrastructure of computing resources in which large pools of systems or clouds are linked together via the internet to provide IT services, such as securely managing billions of online transactions and other highly data-intensive application. Cloud computing is being driven by several factors, including the dramatic growth of connected devices, such as mobile phones, smart cards and other devices. It can be used to break many computing tasks into many smaller pieces and managed in parallel on a massive scale. Thus, Cloud computing offers a simplified, centralized platform for use when needed, lowering costs and energy use. But cloud computing is in its nascent stage of progression. There are no mathematical models regarding the performance analysis of cloud computing. In this paper, we intend to introduce Mathematical Model for cloud computing.

Keywords

Web Service; Ant System; Scalability

1. INTRODUCTION

Cloud computing provides a platform for the execution of massive tasks on cloud instead of the execution of tasks on users' Personal Computers, Servers etc. It is very beneficial for small organizations that cannot afford huge investment on their IT sector but at the same time expect maximum benefit from this supporting industry in order to survive in today's complex competitive business world. Cloud computing can help such organizations by providing massive computing power, unlimited storage capacity, less maintenance cost, availability of useful web-services etc. As per Buyya et. al [7], "A Cloud is a type of parallel as well as distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumer". The definition clearly implies that there is a Service Level Agreement (SLA) between the provider and the consumer for getting services from cloud on pay per user basis. Actually Cloud computing offers the three layers of abstraction such as Infrastructure as a Service (IaaS), Platform as a Service(PaaS), Software as a Service (SaaS). SaaS provides different types of applications as a Service for the end user. It includes different useful web-services. PaaS provides a standard platform for better execution of application with proper exploitation of physical resources using Middleware Services. PaaS includes Database services, Middleware Services etc, In this endeavor we intend to

throw light on Software as a service (SaaS). IaaS provides the infrastructure of cloud consisting of physical resources like CPU, Storage, and Network etc. Again middleware of PaaS consists of Core Middleware and user level middleware. Core Middleware provides a set of services [7] including "Admission Controller", "Service Request Examiner Monitoring", "Accounting Billing" etc; whereas the user level middleware provides access point of services as delivered by the Core Middleware. The "Service Request Examiner and Admission Controller" accepts the request after ensuring that there is an availability of resources for carrying out the request. For this, the "Service Request Examiner and Admission Controller" interacts continuously with "Virtual Machine Monitor" mechanism regarding resource availability; while the "Pricing Mechanism" decides the charges for request service. The final cost of the request service is fixed by the "Accounting Mechanism", based on the actual usage of resources for the requested service. The role of "Virtual Machine Monitor Mechanism" is to keep track of the availability of Virtual Machines (VM). The "Dispatcher Mechanism" executes the tasks on allocated Virtual Machines, where as the "Service Request Monitor" keeps track of the execution of the service request. The Virtual Machines execute the job on physical machines, which consist of multiple computing servers, data storage etc. It is observed that performance monitoring of any application in PaaS is always complex, different and challenging. In the absence of precise knowledge about the availability of resources at any point of time and due to the dynamic resource usage scenario, prediction-based analysis is not possible. Thus, performance analysis in PaaS must be characterized by dynamic data collection (as performance problems must be identified during run-time), data reduction (as the amount of monitoring data is large), low-cost data capturing (as overhead due to instrumentation and profiling may contribute to the application performance), and adaptability to heterogeneous environment. Just like the day to day utility services such as water, gas, electricity etc are essential for the smooth running of our daily lives; the trend of cloud computing [16] is also to provide different IT resources on demand on a pay per user basis. The IT resources include different computational web services of different nature like tax calculation web service, weather information web service, shipping status web service etc. The aim of Cloud computing is to be global and to provide such computational web services to the masses, ranging from the end user that hosts its personal documents on the internet to enterprises outsourcing their entire IT infrastructure to external data centres. In fact, Cloud has given a new approach to make IT a real utility which is global and complete for the end users. The agenda of computational web service of SaaS is to offer simple

web methods that computational clients can call to perform application specific computation on their own data. Web Services are programmable and reusable. They are available anywhere via the internet. Programs built on the basis of this model will run across multiple websites extracting information from each of them and combining and delivering it in a customized form to any device anywhere in the world. The potential of web services is unlimited. For example, a software company may provide a web service to calculate income tax. Any company that wants to calculate income tax can subscribe to this web service. The company offering the service can dynamically update it to accommodate new taxation rates. The subscribers need not to do anything to get the new updates. In future a collection of such web services may replace packaged software. As web services break down the distinction between the internet, standalone applications and computing devices of every kind, they enable business to collaborate and offer an unprecedented range of integrated and customized solutions that enable their customers to act on information at any time, any place and on any device.

Organization of this paper is as follows: Related work is discussed in Section II. Proposed Mathematical Model of Cloud Computing is discussed in Section III. Proposed Algorithm is discussed in Section IV. Section V and Section VI give details of the experimental setup and results. Section 7 concludes with a direction of the future work..

2. RELATED WORK

The potentiality of computing utility was first pointed by Leonard Kleinrock[1], one of the pioneer researcher, who seeded the idea of internet. He had predicted long back, regarding the wide use of “computer utilities” as our day to day utilities like “Electrical Utilities”, “Telephone Utilities” etc. Today, his vision came into existence in the form of Cloud computing. Cloud computing provides different utilities in the form of web-services, which are pay per user basis. Most of the existing work in web service performance focuses on the latest trend of technologies and standards. Andreozzi et al. [2] present a model for rigorous representation of service characteristics. Gouscos et al. [3] presents a simple approach to model certain web service management attributes. Thomas et al.[4] represent distributed web service by modeling the flow of messages and methods in a web service transaction. Tue et al.[5] discuss design strategies to improve the performance of web service. Levy et al.[6] present an architecture and prototype implementation of performance management system for cluster based web service. Cardellini et al. [15] consider different categories of web applications, and evaluate how static, dynamic and secure web service requests affect performance and quality of service at distributed web sites. Buyya et. al.[7] have proposed the use of market oriented resource management in order to regulate the supply and demand of cloud resources at market equilibrium. Again Buiya et. They have proposed the architecture for market oriented allocation of resources within clouds. Xinhui et. al. [9] have proposed total cost of ownership of cloud computing using a suite of metrics and formulas. But none of above mentioned works have given any systematic mathematical model regarding the analysis of cloud computing.

3. PROPOSED MATHEMATICAL MODEL OF MARKED ORIENTED CLOUD COMPUTING

In this paper, we present a mathematical model of Market-oriented cloud. The architecture of market oriented cloud computing[7] implies that there are three important agents viz. “Virtual Machine(VM) Monitor”, “Dispatcher” and “Service Request Monitor”, which collectively guide the service request examiner and controller, whether to accept a job or not and if accepted, how to execute the job effectively and efficiently. For these agents to work effectively and efficiently, we propose Bee and Ant colony system. It is observed that honey bees [17] work in a decentralized, self organized manner. The honey bee colonies practice division of labor between the forgers, who work in the field collecting the nectar and the food-stores who work in the hive to store nectar. The forger bees search the nectar at random and after getting specific nectar it returns back to the hive and starts dancing in order to motivate the follower forgers to access that very nectar. The duration of this dance is closely correlated with the search time experienced by the dancing bees. The VM Monitor is considered as a hive that consists of bee agents. From the bee hive, the agents start to explore for the purpose of collecting the information regarding resource availability. After returning to the hive, they start to perform dances on the dancing floor in order to communicate about the food source i.e. the availability of Virtual Machines. The orientation as well as the duration of dance of bees inform the other bees regarding the exact place as well as its richness[12][13][14]. There are two types of dance of bees viz. tremble dance(T.D) and waggle dance(W.D). The W.D of bees implies that there is no availability of Virtual Machine for carrying out the new job, whereas the T.D dance of bees implies the availability of V.M for carrying out new jobs. The bees existing in the “Service Request Monitor hive” observe the tremble dance of the bees on the dancing floor by VM Monitor bees and hence start to find the nectar (i.e. available VMs), in order to reschedule the pending jobs on the available VMs and hence enhance the performance of the Cloud. The waggle dance helps the “Service Request Controller and Examiner” to take the decision of not accepting any new job, whereas tremble dance helps the “Service Request Controller and Examiner” to entertain new jobs and to carry out jobs on V.M by the Dispatcher. Now in order to enhance the performance of dispatcher, we propose that dispatcher should use ant colony system in order to assign jobs to the available V.Ms. Ant System was proposed by Dorigo et al.[18]at the year 1992. In Ant System, ants built tour based on probabilistic action choice rule, called random proportional rule. Ant Colony works to find the shortest paths between food sources and nest[18].Due to their restricted visibility, ants interact among themselves, by dropping a chemical known as pheromone, which guides them on their way from the nest to the food source and vice versa. In this paper neither ant system nor bee colony has been discussed in details. In this paper, we have proposed the Mathematical Model of cloud and traversing cost of job assignment to different VMs’ by the Dispatcher. Thus role of VM Monitor and Service Request Monitor are beyond the scope of this paper. Here we have focused only on the role of Dispatcher in order to allocate jobs to

available VMs' based on ant colony system. For this, we have proposed a Mathematical Model.

In order to present our new mathematical model, let us consider undirected graph $G(V,A,C)$, where $V = \{ 0,1,2, \dots,n\}$ is a set of nodes representing Virtual Machines. Here we assume that 'node 0' represents the dispatcher, whereas node 1 and onwards representing available Virtual Machines.

Let $A = \{ (i,j): i,j \in V, i \neq j\}$ is a set of arcs joining the nodes, having a travel cost (i.e. distances).

Let $C = \{ c_{ij}: i,j \in V, i \neq j\}$ is the cost of traverse between the nodes .

Let Ω be the total number of ants each with capacity ξ_c , assign jobs to M_c (Virtual Machine) based on the stochastic demands. Here Virtual Machine demands are stochastic variables Ψ_i , where $i = \{ 0,1,, \dots,n\}$, which are independently distributed with known distributions. Here the actual demand of each Virtual Machine is not known unless and until the ant is not arriving at the Virtual Machine. Here we assume that the demand Ψ_i does not exceed the ξ_c and follows a discrete distribution and probability mass function p_{id} .

$p_{id} = \text{Prob}(\Psi_i = d)$, Where d is demand of Virtual Machine and $d \in W$, set of Whole Number.

According to the Virtual Machine's actual demand the ant decides whether to proceed to the next Virtual Machine or to go back to the job executor for restocking of new job. Sometimes the choice of restocking is better one, when the ants are not carrying any jobs to assign or do not have sufficient load to serve proceeding Virtual Machine's demand. This action is called preventive restocking. The goal of preventive restocking is backtracking of the ants when they do not have sufficient jobs to assign proceeding Virtual Machine. Thus ants make back and forth trip to the job executor and Virtual Machine for assigning the jobs to the available Virtual Machine.

Let $Z_{i,j,k}$ = Binary flow variable

Obviously $Z_{i,j,k} = 1$, if path (i,j) is traversed by k^{th} agent.

$= 0$, otherwise

$J_{i,k}$ = Quantity of job is supplied at i^{th} Virtual Machine by k^{th} agent.

Ω = Total Number of ants initially available at the Job Executor.

δ = Remaining jobs associated with the ants.

A_c = Capacity of ants for carrying out jobs.

$\phi_{jk}^P(\delta)$ = Expected Cost of the move forward action.

$\phi_{jk}^r(\delta)$ = Expected cost of preventive restocking.

If $\phi_{jk}(\delta)$ is the job assignment to the j^{th} Virtual Machine by the k^{th} ant then our proposed objective function is:

$\phi_{jk}(\delta) = \text{Minimum}(\phi_{jk}^P(\delta), \phi_{jk}^r(\delta))$

Where the parameters are given below:

$$\phi_{jk}^P(\delta) = C_{j,j+1} + \sum_{d: d \leq \delta} \phi_{j+1,k}(\delta-d)p_{j+1,d} + \sum_{d: d > \delta} [2C_{j+1,0} + \phi_{j+1,k}(\delta+A_c-d)p_{j+1,d}] \quad 1$$

Again $\phi_{jk}^r(\delta)$ be given as:

D

$$\phi_{jk}^r(\delta) = C_{j,0} + C_{0,j+1} + \sum_{d=1}^D \phi_{j+1,k}(A_c-d)p_{j+1,d} \quad 2$$

Now we discuss two cases as follows:

Case I: When δ is greater than next Virtual Machines' demand.

For $(j+1)^{\text{th}}$ Virtual Machine, we have only $\sum \phi_{j+1,k}(\delta-d)p_{j+1,d}$

$d: d \leq \delta$

Case II: When δ is less than next Virtual Machines' demand.

For $(j+1)^{\text{th}}$ Virtual Machine, we have

$\sum [2C_{j+1,0} + \phi_{j+1,k}(\delta+A_c-d)p_{j+1,d}]$

Again

$2C_{j+1,0}$ is the cost of traveling to the job executor and back to route

$C_{j,0}$ is the distance cost from j^{th} customer back to Job Executor.

$C_{0,j+1}$ is the distance cost from job executor i.e. '0' to $j+1^{\text{th}}$ Virtual Machine.

4. PROPOSED ALGORITHM

Based upon the above discussion, the proposed algorithms are given below:

Procedure User_Request_Submission (jobs)

Begin

Boolean flag

flag \leftarrow SRBAC(jobs)

if (flag = 0) then

write("Request can not be carried out")

else

write("Request is accepted")

End

function Boolean SRBAC (jobs)

Begin

Boolean flag.

flag \leftarrow VMMonitor(Request for checking of

Resource Avability)

If (flag =0)

```

        Return(false)
Else
    do
        submit Expected_Cost (jobs)
        Return(true)
    end
End

Function Boolean VMMonitor (Request_for_
                            _Checking_of_
                            _Resource_Avability)

Begin
    Boolean flag
    Bee stores in hive.
    Forger Bees from Hive starts to search for nectar
    i.e. Availability of VMs.
    Forger Bees return back and start dancing on
    the dancing floor.
    If (dance be Waggle Dance)
        then Return( false.)
    else
        do
            Call Server_Request_Monitor()
            Return(True)
        end
    End

```

Procedure Server_Request_Monitor()

```

Begin
Reschedule the Pending jobs
End

```

Procedure Expected_cost (job)

```

Begin
Cost ← Submit_jobs_to_Dispatcher (jobs)
Write(Expected Cost for Traversing is equal to Cost)
End

```

Function Submit_jobs_to_Dispatcher(jobs)

```

Begin
[Initialization]
Set  $\delta \leftarrow A_c$ 
Set  $j \leftarrow \xi_1$  //  $\xi_1$  is last VM accessed by  $I_{th}$  Ant
Set  $T1 \leftarrow 0$ 

```

```

Set  $T2 \leftarrow 0$ 
Set  $k \leftarrow 1$ 
 $N \leftarrow$  Max_No_of_Virtual_Machine
Repeat until  $k \neq N$ 
Repeat until  $\delta \neq 0$ 
Repeat until  $j \neq 1$ 
Compute  $\phi_{jk}^r(\delta)$  using 2
 $T1 \leftarrow T1 + \phi_{jk}^r(\delta)$ 
 $j \leftarrow j - 1$ 
Compute  $\phi_{jk}^p(\delta)$  using 1
 $T2 \leftarrow T2 + \phi_{jk}^p(\delta)$ 
 $\delta \leftarrow \delta - 1$ 
 $k \leftarrow k + 1$ 
 $T \leftarrow T1 + T2$ 
Return (T)
End.

```

5. EXPERIMENTAL SETUP

A private cloud has been setup based on Ubuntu's 10.04 Server edition, that consists of two Servers – Server A and Server B. Server A acts as the cloud, cluster, warehouse and storage controller and Server B acts as node controller. We configured Machine A on a Core2duoX6800 processor based machine with 2GB DDR 2 RAM and 80 GB Hard disk. Machine B is running on an AMD PhenomeII X4 965 processor with 4 GB DDR 3 RAM and 250 GB Hard disk. The nodes communicate through a fast local area network. This paper demonstrates the results of local accessing of multiple web services running on the web servers. Experiments have been carried out on above mentioned cloud environment. In order to execute the web services on Machine A, we have used the following commands:

```

If [ ! -e ~/.euca/mkey.priv ]; then
    mkdir -p -m 700 ~/.euca
    touch ~/.euca/mykey.priv
    chmod 0600 ~/.euca/mykey.priv
    euca-add-keypair mykey > ~/.euca > ~/.euca/mykey
fi

```

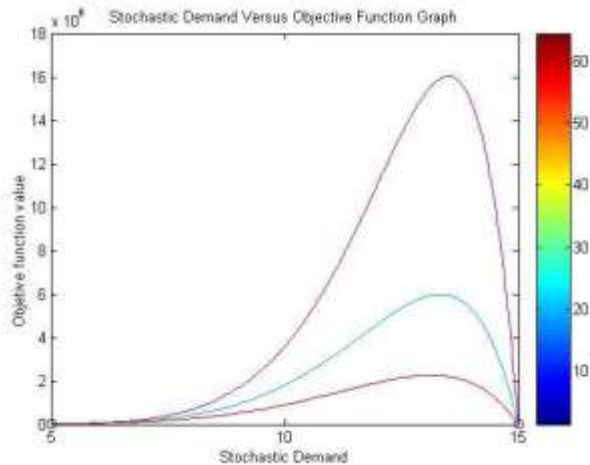
Thus creating keypair, we now log into our instance as root. After that, we can monitor state of instance using the command:

```
watch -n5 euca-describe-instances
```

Thus we start to access different web services.

6. RESULTS

We have experimented with two different web services. However, in order to demonstrate the effectiveness of our system, we submitted multiple images of the same web services as separate jobs. The jobs have been initiated at different times. The analysis of stochastic demand versus objective function, based upon our experiments, is shown in figure 1.



The figure 1 clearly implies that as stochastic demand gradually increases our proposed objective function slowly increases and provides more optimized value when the stochastic demand is within a moderate to maximum range. At maximum stochastic demand, we do not find the best performance exhibited by our proposed objective function. This fact has been pointed by the color code. The reddish part of the color code implies that the objective function has attained its optimized value and further increase of stochastic demand starts its retardation. This fact is true for all categories of jobs including very small amount of jobs, moderate jobs or for huge jobs. This fact is shown by different color curve in figure 1. This surf deteriorations of the objective function is due to more back and forth movements of the ants in order to assign the jobs to the available VMs, provided that the stochastic demands of VMs' are very high. Again we observe that initially when stochastic demand is very less then also objective function attains its optimized value gradually due to other parameters like rescheduling of the pending jobs by "Service Request Monitor" (which takes place before "Dispatcher" that dispatches the new jobs to the available VMs.). In this experiment we have considered three values of the stochastic demands, 5- 10 value of stochastic demand implies low demand where as 10-15 values implies high demand and value 15 implies extreme stochastic demand.

8. CONCLUSIONS AND FUTURE DIRECTION OF WORK

In this endeavor we have emphasized on the role of "Dispatcher" in order to allocate jobs to available VMs'. We have seen that as stochastic demand of VMs' is within a specific range, the performance of the "Dispatcher" is the best and the objective function is optimized. But further increase in stochastic demand leads the ants to move back and forth more than ever in order to assign the jobs to "Dispatcher" and hence a negative impact is observed in the performance of the objective function. Further the roles of "Virtual Machine Monitor" and "Service Request Monitor" have not discussed in this paper, we intend to do so in our forthcoming endeavor.

9. REFERENCES

- [1] L.Keleinrock. A vision for the Internet. ST Journal of Research, Nov.2005, pages 4-5
- [2] S.Adreozzi, P. Ciancarini, D. Montesi, R. Moretti, "Towards a model for quality of web and grid service" In Proc 13th IEEE international Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises(WET ICE'04), 2004, page 271-276,.
- [3] D. Gouscos, M. Kalikakis, and P. Georgiadis. "An approach to modeling Web service QoS and provision price ", in Proceeding 3rd International Conference on Web Information Systems Engineering Workshops, 2003,pages 121-130.
- [4] J.P Thomas, M.Thomas and G.Ghinea "Modeling of Web service flow", in Proceeding IEEE International Conference on E-Commerce (CEC 03), 2003, pages 391-398,.
- [5] S.Tu , M.Flanagin Y. Wu, M. Abdelguerfi, E. Normand, V. Mahadevan, " Design Strategies to improve performance of GISWeb services", in Proceeding International Conference on Information Technology : Coding and Computing(ITCC04), 2004, pages 444-448,.
- [6] R.Levy, J Nagarajaro, G. Pacifici, M. Spreitzer , A. Tantawi, and A. Youssef, " Performance management for cluster based Web services", in Proceeding IFIP/IEEE 8th International Symposium on Integrated Network Management, 2003, pages 247-261 ,.
- [7] Rajkumar Buyya, Chee Shin Yeo and Srikumar Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities", The 10th IEEE International Conference on High Performance Computing and Communications, IEEE Computer Society, 2008, pages 5-13.
- [8] Foster I., C. Kesselman, J. Nick, and S. Tuecke, "Grid Services for Distributed System Integration", IEEE Computer, June 2002, pages 37-46,.
- [9] Xinhui Li, Ying Li, Tiancheng Liu, Jie Qui, Fengchun Wang "International Conference on Cloud Computing" IEEE Computer, 2009 , pages 93-100,
- [10] Furlinger K., M. Gerndt, "A Lightweight Dynamic Application Monitor for SMP Clusters", HLRB and KONWIHR joint Reviewing Workshop 2004, March 2004.
- [11] Furmento N., A. Mayer, S. McGough, S. Newhouse, T. field, and J. Darlington, "ICENI: Optimization of Component Applications within a Grid environment", Parallel Computing, 2002, pages 1753-1772,.
- [12] H.FWedde and M.Farooq, "A comprehensive review of nature inspired routing algorithm for fixed telecommunication networks", Journal of System Architecture, 2006, vol. 52, no. 8-9, pages 461-484,.
- [13] C. Blum and D. Merkle, Eds., Swarm Intelligence: Introduction and Applications, ser. Natural Computing Series. Springer, 2008, pages 344-349.

- [14] A. Abraham, C. Grosan, and V. Ramos, *Stigmergic Optimization Studies in Computational Intelligence*. Secaucus, NJ, USA: Springer-Verlag New York., 2006, pages 124-129
- [15] V. Cardellini, E. Casalicchio, and M. Colajanni, "A performance study of distributed architectures for the quality of Web services", in *Proceeding 34th Annual Hawaii International Conference on System Sciences*, 2001
- [16] L. Vaquero, L. Rodero-Marino, J. Caceres, M. Linder, "A break in the clouds: towards a cloud definition", *SIGCOMM Computer Communication Review*, 2009, pages 137-150.
- [17] T.D. Seely, "The Wisdom of the Hive", Harvard University Press, 1995, pages 295-296. Marco Dorigo and Thomas Stutzle "Ant Colony Optimization", Prentice-Hall of India, India, 2001
- [18] Marco Dorigo and Thomas Stutzle "Ant Colony Optimization", Prentice-Hall of India, India, 2001