



Time Critical Analysis of Resource Technique in Cloud Computing

Sanjeev Dhawan¹, Nitin Kaushik²

¹Assistant Professor of Computer Science & Engineering,

²Research Scholar, M. Tech. (Computer Engineering),

^{1,2}Department of Computer Science & Engineering,

University Institute of Engineering & Technology,

Kurukshetra University, Kurukshetra-136 119, Haryana, INDIA

Abstract: Cloud computing distributes the computational tasks on the resource pool which consists of massive computers so that the service consumer can gain maximum computation strength, more storage space and software services for its application according to its need. A huge amount of data moves from user to host and hosts to user in the cloud environment. Based on the above two considerations, how to select appropriate host for accessing resources and creating a virtual machine (VM) to execute applications so that execution becomes more efficient and access cost becomes low are the challenging tasks. In this paper, an attempt has been made to propose a host selection model based on minimum network delay to minimize propagation time of input and output data by selecting nearest host into the network and cloudlet.

Keywords: Cloud Computing; Resource; Task Scheduling; Virtual Machine

I. Introduction

Cloud computing can be defined as "a type of parallel and 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 providers and consumers" [1]. Cloud computing is a model that enables on demand access to a shared pool of configurable computing resources [2,]. Cloud computing is an evolving technology. Cloud computing delivers infrastructure, platform, and software that are made available as subscription-based services in a pay-as-you-go model to consumers. These services are referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) in industries. Cloud computing is Internet-based computing. Although many formal definitions have been proposed, NIST provides a somewhat more objective and specific definition here: "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."

II. Types of Clouds

A. Private Cloud

A cloud that is used exclusively by one organisation. The cloud may be operated by the organisation itself or a third party. The St. Andrews Cloud Computing Co-laboratory and Concur Technologies are example organisations that have private clouds.

B. Public Cloud

A cloud that can be used (for a fee) by the general public. Public clouds require significant investment and are usually owned by large corporations such as Microsoft, Google or Amazon.

C. Community Cloud

A cloud that is shared by several organisations and is usually setup for their specific requirements. The Open Cirrus cloud test bed could be regarded as a community cloud that aims to support research in cloud computing.

D. Hybrid Cloud

A cloud that is setup using a mixture of the above three deployment models. Each cloud in a hybrid cloud could be independently managed but applications and data would be allowed to move across the hybrid cloud. Hybrid clouds allow cloud bursting to take place, which is where a private cloud can burst-out to a public cloud when it requires more resources.

III. Cloud Computing Architecture

Generally speaking, the architecture of a cloud computing environment can be divided into four layers i.e. the hardware/ datacenter layer, the infrastructure layer, the platform layer and the application layer. These may be described as follows:

A. The Hardware Layer

This layer is responsible for managing the physical resources of the cloud, including physical servers, routers, switches, power and cooling systems. In practice, the hardware layer is typically implemented in data centers. A

data center usually contains thousands of servers that are organized in racks and interconnected through switches, routers or other fabrics. The components of the hardware layer include hardware configuration, fault tolerance, traffic management, power and cooling resource management.

B. The Infrastructure Layer

This layer is also known as the virtualization layer, the infrastructure layer creates a pool of storage and computing resources through partitioning the physical resources by using virtualization technologies such as Xen, KVM and VMware. The infrastructure layer is an essential component of cloud computing. Its many key features, such as dynamic resource assignment, are made available only through virtualization technologies.

C. The Platform Layer

This layer is built on the top of an infrastructure layer. This layer consists of operating systems and application frameworks. The purpose of the platform layer is to minimize the burden of deploying applications directly into VM containers. For example, Google App Engine operates at the platform layer to provide API support for implementing storage, database and business logic of typical web applications.

D. The Application Layer

At the highest level of the hierarchy, the application layer consists of the actual cloud applications. Different from traditional applications, the cloud applications can leverage the automatic-scaling feature to achieve better performance, easy availability and lower operating cost. Compared to traditional service hosting environments, such as dedicated server farms, the architecture of cloud computing is more modular. Each layer is loosely coupled with the layers above and below allowing each layer to evolve separately. This is similar to the design of the OSI model for network protocols. The architectural modularity allows cloud computing to support a wide range of application requirements while reducing management and maintenance overhead costs.

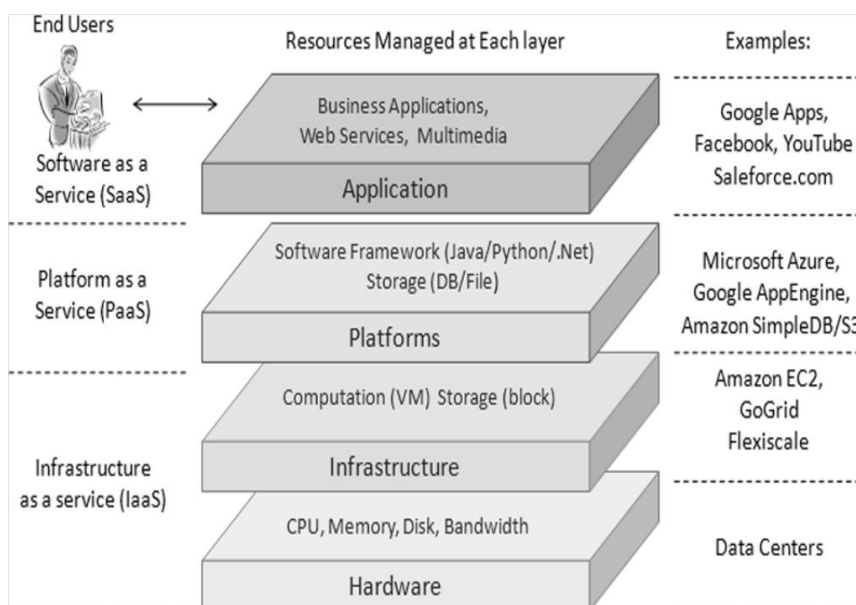


Figure 1: A Layered Model of Cloud Computing [10]

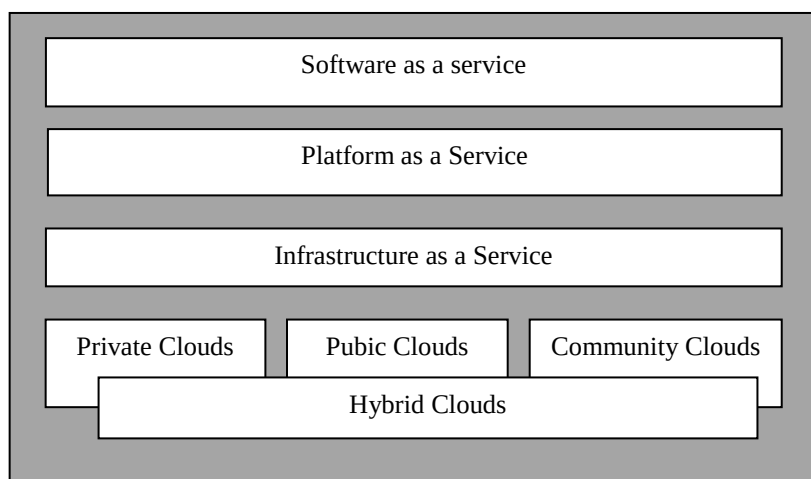


Figure 2: Cloud Computing Deployment and Service Models

IV. Related Work

In an area of cloud task scheduling, Kun Li *et al.* [1] experienced a rapid development of cloud computing both in academia and industry. They promoted by this business which determines its focus on user applications. This technology aims to offer distributed, virtualized, and elastic resources as utilities to end users. It has the potential to support full realization of 'computing as a utility' in the near future. With the support of virtualization technology [2, 3], cloud platforms enable enterprises to lease computing power in the form of virtual machines to users. Because these users may use hundreds of thousands of virtual machines (VMs) [4] as it is difficult to manually assign tasks to computing resources in clouds [5, 6]. Shu-Ching *et al.* [7] proposed an efficient algorithm for task scheduling in the cloud environment. They depicted that a good task scheduler should adapt its scheduling strategy to the changing environment and the types of tasks. Therefore, a dynamic task scheduling algorithm, such as Ant Colony Optimization (ACO) [8, 9], is appropriate for clouds. ACO algorithm is a random search algorithm, like other evolutionary algorithms. It imitates the behavior of real ant colonies in nature to search for food and to connect to each other by pheromone laid on paths traveled. Enda Barrett *et al.* [2] discussed the scheduling of workflow applications to involve the mapping of individual workflow tasks to computational resources based on a range of functional and non-functional quality of service requirements. In workflow based applications dependencies exist amongst tasks which requires the generation of schedules in accordance with defined precedence constraints. These constraints pose a difficult planning problem, where tasks must be scheduled for execution only once all their parent tasks have completed. In general the two most important objectives of workflow schedulers are the minimization of both cost and make span. The cost of workflow execution consists of both computational costs incurred from processing individual tasks, and data transmission costs. With scientific workflows potentially large amounts of data must be transferred between compute and storage sites. In addition the system employs a genetic algorithm to evolve workflow schedules. The overall architecture is presented, and initial results indicate the potential of this approach for developing viable workflow schedules on the cloud.

In a more pronounced approach, Boonyarith Saovapakhiran *et al.* [3] used heterogeneous computing platforms such as Grid and Cloud computing for job flow maximization to check their global performance. Under the assumption of jobs comprised of subtasks forming DAG jobs, they focused on how to increase utilization and achieve near-optimal throughput performance on heterogeneous platforms. They analyzed and proposed algorithm which can be analytically derived to aggregate multiple jobs using good scheduling to maximize the throughput. Consequently, its limit is asymptotically converging to a certain value and can be written in the form of the service time of subtasks. Moreover, Hsu and Thinn [4] investigated the deployment of Cloud computing on a large set of virtualized computing resources in different infrastructures and various development platforms. One of the significant issues in cloud computing system is the scheduling of virtual resources and virtual machines (VMs). To address this issue, they proposed an efficient approach for virtual machines scheduling in VM management also called EVMSA (Efficient Virtual Machines Scheduling Algorithm) that provides the effective and efficient resource allocation. The proposed approach is going to test the evaluation on open source private cloud architecture. The major contribution of this paper is to improve the resource utilization such as CPU, memory, disk and to minimize the turnaround time of VMs.

V. Proposed Work for Resource Selection Technique

A cloud environment can be considered as a set of K centers $D = \{d_1, d_2, \dots, d_K\}$, which are located in different place and connected by links of different bandwidths. For an application composed of a set of N independent jobs $J = \{j_1, j_2, \dots, j_N\}$ ($N \gg K$), each job j is subset of J , requires a set of K datasets, denoted by F_j , which are accessed on a subset of D . Consider a task j that has been submitted to a VM, which is created on data center d , for execution. Now we want to find out nearest data center d (where we can generate VM for that particular job) which is having less propagation delay. Furthermore, by considering that a task j has been submitted to a VM, which is created on data center d for execution. For each dataset, the time needed to transfer it from d_f to d is denoted by $T_{if}(d_f, d)$. The estimated data transfer time for the $V_m, T_i(j)$ is the maximum value of all the times for transferring all the datasets required between the VM.

Where $R_i(d_f)$ is the time span from requesting for f_d to getting the first byte of f . In addition, the data access cost $C(j)$ in our research is a function of $c(f)$, the access cost of each replica f . Here, we consider that each replica is lying on local data center or on remote data center.

- *Begin Main*
- *For a VM, create the adjacency matrix A with Broker forming the rows and hosts forming the columns*
- *Let k be a broker making request*
- *Sort the k^{th} row of matrix A in the ascending order of the propagation delay and Store sorted host IDs in $S[i]$.*
- *$i = 0$*
- *repeat step 4 and 5 until VM created on $S[i]$*
- *$i++$*

➤ End Main

The above algorithm gives the steps of getting data center which is best for creating VM on it and having shortest path. Here we construct the delay matrix which contains delay of each pair of node using shortest path algorithm. Now when broker sends request to host for resources, first broker will select the host which has minimum delay for communication. For this generate one array *S* which stores the *host id* in ascending order of delay. Thereafter, it allocates host for VM and selects first element in array and check VM is created or not. If resources are available for that VM then new VM are created else select next host from *S* until suitable host is not found or all hosts are check.

VI. Result Analysis

The test method in this evolution contains four data centers and five brokers. Data center contain number of hosts that are connected by high capacity network. In our experiments, we randomly generate the bandwidth and delay of links and then submit the number of jobs. The results of the experiment are shown in the figure 3 and 4.

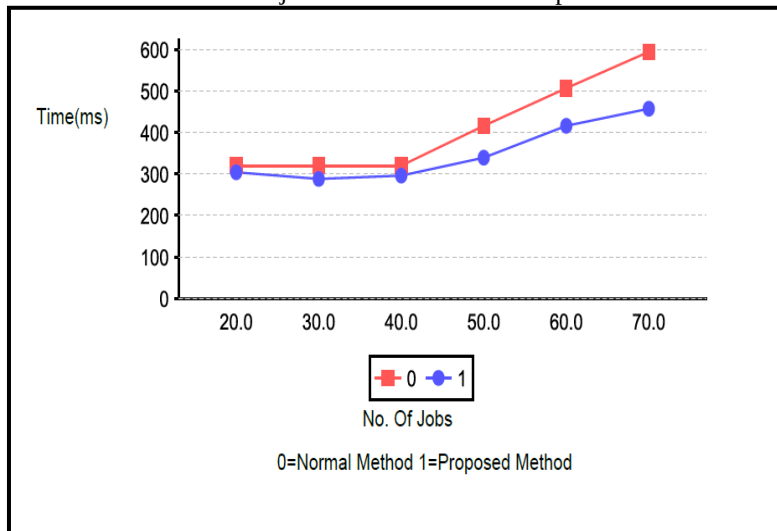


Figure 3: Time comparison between normal and proposed method using number of jobs

Result in the figure 4 shows that when we use proposed algorithm for getting nearest data center and creating VM on it for execution of job then we got better output with less propagation delay and less time to execute the job. Service quality can be further improved by the application of load balancing at the application level across data centers.

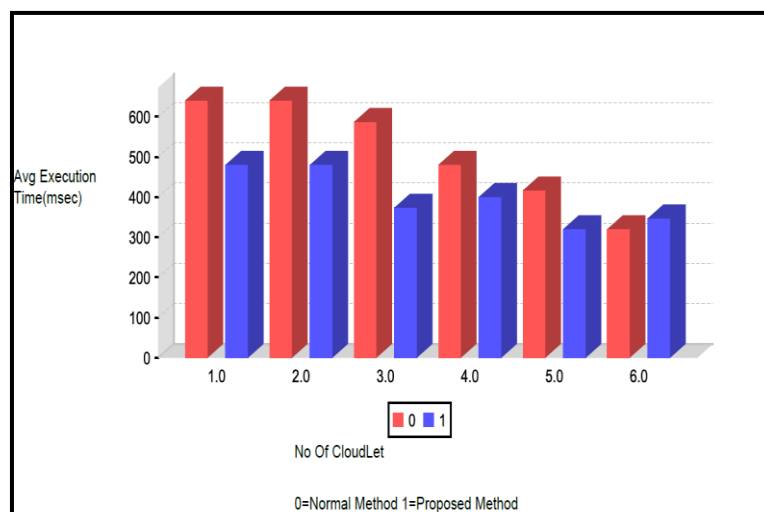


Figure 4: Average time comparison between normal and proposed method using Cloudlets.

The results in the figure 4 shows that by using more data center, the performance is better and it takes less execution time for completing the jobs. Here we are getting the nearest host in the data center and numbers of cloudlets are working from the range 1 to 6. We also compared two techniques for values of makespan as shown in figure 5.

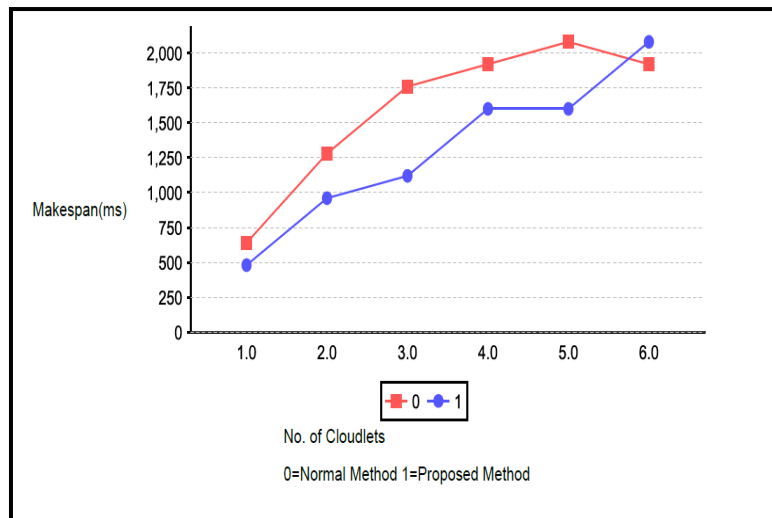


Figure 6: Make span comparison between normal and proposed method using Cloudlets.

VII. Conclusion and Future Work

In this research paper, a novel technique for job submission in cloud environment is proposed. The proposed technique consider both cloudlets transfer time and file transfer time while selecting appropriate hosts for cloudlet (job) submission on distributed resource with an objective to minimize execution time and cost. In this research paper, we compare normal method and proposed method random submissions with respect to the make span and turnaround time of execution. The proposed technique out performs other techniques for all parameters by increasing the locality. It selects the hosts within that regions, in other words it selects the host with minimum propagation delay. The future work involves implementation of algorithm on actual cloud environment and performance comparison for real workload traces in cloud environment. In coming future the algorithms are to be made to increase the efficiency and use the technique on the real world.

VIII. References

- [1] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang, "Cloud Task scheduling based on Load Balancing Ant Colony Optimization", IEEE Sixth Annual China Grid Conference, 2011, 978-0-7695-4472-4/11.
- [2] Enda Barrett, Enda Howley, Jim Duggan, "A Learning Architecture for Scheduling Workflow Applications in the Cloud", 2011 Ninth IEEE European Conference on Web Services 978-0-7695-4536-3/.
- [3] Boonyarith Saovapakhiran, George Michailidis, Michael Devetsikiotis, "Aggregated-DAG Scheduling for Job Flow Maximization in Heterogeneous Cloud Computing", IEEE, 2011 978-1-4244-9268-8/11.
- [4] Hsu Mon Kyi, Thinn Thu Naing, "An Efficient Approach for Virtual Machines Scheduling on a Private Cloud Environment", IEEE, 2011, 978-1-61284-159-5/11.
- [5] V. Nelson, V. Uma, "Semantic based Resource Provisioning and Scheduling in Inter-cloud Environment", IEEE, 2012, 978-1-4673-1601-9/12.
- [6] Jinhua Hu, Jianhua Gu, Guofei Sun Tianhai Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", IEEE, 2010, 978-0-7695-4312-3/10.
- [7] Shu-Ching, Wang, Kuo-Qin, Yan, Shun-Sheng, Wang, Ching-Wei, Chen, "A Three-Phases Scheduling in a Hierarchical Cloud Computing Network", IEEE Third International Conference on Communications and Mobile Computing, 2011 978-0-7695-4357-4/11.
- [8] Praveen K. Gupta, Nitin Rakesh, "Different Job Scheduling Methodologies for Web Application and Web Server in a Cloud Computing Environment", IEEE Third International Conference on Emerging Trends in Engineering and Technology, 2010.
- [9] Saurabh Kumar Garg, Chee Shin Yeo, Arun Anandasivam, Rajkumar Buyya "Energy-E_client Scheduling of HPC Applications in Cloud Computing Environments", IEEE, 2009.
- [10] J. Lee, B. Tierney, and W. E. Johnston, "Data Intensive Distributed Computing; A Medical Application Example," in HPCN Europe 99: *Proceedings of the 7th International Conference on High-Performance Computing and Networking*. London, UK: Springer-Verlag, 1999, pp.150-158