AN APPROACH TO IMPROVE EFFICIENCY OF CLOUD COMPUTING

Vrajesh Sharma  
Ph.D. Scholar,  
IKG Punjab Technical University

Nipun Chhabra  
Ph.D. Scholar,  
IKG Punjab Technical University

Manju Bala  
Khalsa College of Engineering & Technology,  
Amritsar, (Punjab), India.

Abstract- Cloud Computing is an invasive and emerging technology which allows the users to pay as they need and enables hosting of pervasive applications from consumer, scientific, and business domains. Cloud Computing is offering utility-oriented IT services to users world-wide [2]. Here we have tried to improve the efficiency of cloud computing platform for data intensive applications with minimal cost and minimal time variance while reducing the scheduling delay and prediction error by categorizing the jobs in the virtual environment.

Keywords-Cloud Computing, Optimizing the Workflows, Reallocations of Virtual Machines, Performance Isolation, Reduced Prediction Error, Improved Quality of Service, Service Integration.

I. INTRODUCTION

Though, cloud computing is an emerging trend in computer science, the same idea has been there around us for quite long. This computing is named with a metaphor of “cloud” because the data and applications exist on a "Ubiquitous Network" of Web servers. So “Cloud Computing” is a newly coined term for an old desire to use computing as a utility, which is emerging as a commercial reality. As the growing cost of tuning and managing computer systems is leading to outsourcing of commercial services to hosting centers, it endows with an adept and better way to carry out the tasks which were submitted by the users. Cloud Computing can be categorized into three categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and software as a Service (SaaS). IaaS provides servers and storage on demand and the consumer needs to pay for it, accordingly. PaaS provides the framework to the user to build and develop his application. SaaS facilitates the user to avail an application on demand through the browser; the service being sold using this technique is referred as Utility Computing. As computing systems are being budge to cloud environment gradually/progressively, the main feature which is being appreciated the most, is a pay-as-you-use basis.

Fig. 1 Examples of SaaS, PaaS and IaaS
As an emerging trend, cloud computing successfully offers many of the services to their customers globally via network. In recent days, the data intensive applications play an important role in cloud services like recommendation system, search engine, financial analysis, space exploration etc. The data intensive applications are not typical and the wide usage of cloud enables the data intensive applications to be executed with less time. In cloud services, millions of user requests, which are diverse in nature, can arise at a time resulting in a tough time for scheduling, so for optimizing and rendering a superior cloud service, the scheduling strategies are desired.

II. Virtualization

The key enabling factor for cloud computing is the virtualization technology, e.g., Xen (a hypervisor), which provides an abstraction layer on top of the physical resources and allows multiple operating systems and applications to run on the same hardware, simultaneously. VM technology allows multiple virtual machines to run on a single physical machine.

![Virtual Machine Diagram](image)

Fig. 2  Diagrammatic Representation of a Virtual Machine

As shown in the Fig. 2 virtual machine monitors (VMM) encapsulate various/different applications into each separate guest virtual machine (VM), therefore, a cloud provider can control VM consolidation and its migration in order to achieve admirable resource utilization and high availability in large data centers. Ideally, an application which is running inside a VM should achieve such performance as it owns a portion of the machine to itself. Performance should be Independent of co-located applications and VMs that share the same physical resource (Performance Isolation).

Although, extensive work has been done to achieve this so-called performance isolation, including various techniques to ensure CPU fair sharing but still little attention has been paid to data-intensive applications that perform complex analytics tasks on a large amount of data.

Traditionally, these applications are optimized for the storage systems which are hard drive based and where large sequential reads and writes can be issued but this assumption breaks down in a shared, virtualized environment, and therefore the previously optimized I/O requests are no longer advantageous.

Instead, multiple data-intensive applications will be in competition for the limited available bandwidth as well as throughput to network and storage systems, which most of the times, leads to high Input/Output interference and resulting in low performance. In such cases, the combined effects from concurrent applications, when deployed on shared resources, becomes very difficult to predict, and the interference as a result of competing I/Os remains cumbersome for achieving high-performance computing in a virtualized environment. [10] Unlike grid computing, users can install their programs/softwares on the virtual machines (VMs) and can determine how to execute their programs on the cloud computing system [6].

In the era, when there is so much of the advancement in the field of computer and internet technologies, various models and paradigms of cloud computing were successfully used in recent years. For avoiding
potential violations of service-level-agreement (SLA) which is an essential part in cloud computing. VM live migration strategy named iAware, has been proposed which is a lightweight interference-aware strategy [4]. Nowadays, cloud computing systems do provide a better way to process the submitted jobs in terms of scalability, responsiveness and flexibility but still most job and task scheduling troubles on cloud systems are either NP-hard or NP-complete. As per our observations, most rule-based scheduling algorithms (e.g., exhaustive and deterministic algorithms) which are widely used because they are rather simpler and easier in the implementation, unfortunately, are inappropriate at this huge-scale and are inapt for such complex scheduling problems; because the outcome of these scheduling approaches are usually far from optimal. This entails that there is an abundant scope to perk up the scheduling algorithms for cloud computing systems. So this has come up as a promising research direction, which uses progressive heuristics to scheduling on cloud computing systems and has fascinated many researchers from various research domains [7].

III. Scheduling
In Cloud Computing, scheduling plays a pivotal role in efficiently managing the computer services; it is the progression of captivating decisions regarding the allocation of available capacity and /or resources to jobs and /or customers on time. Millions of users share cloud services by submitting their millions of computing tasks to the cloud computing environment [1]. For the cloud environment, scheduling of these millions & millions of tasks turns out to be a spar and thus a scheduling crisis arises in cloud environment, making it complicated to sort out, primarily in the case of hefty, complex and composite jobs like workflows.

The necessity for a scheduling algorithm arises from the requirement where even most up to date systems need to perform multitasking. Job scheduling has been found to be one of the core and challenging issues in cloud environment and it performs a vital role in efficiently managing the various computer services. Due to the simplicity, clemency and lenience in implementation, a wide variety of rule-based/heuristic algorithms are widely used in cloud computing. The dire need for scheduling algorithms has sprung up from the fact that even the most state-of-the-art systems need to perform multitasking. So, the Job scheduling has become one of the main and exigent/demanding issues, in cloud computing. The modern heuristics progresses towards the optimal solutions for scheduling in cloud computing and therefore this field has fascinated many researchers from various research domains.

The need of Job Scheduling in cloud computing arose
1.) To balance the workload (load balancing).
2.) To ensure quality of service.
3.) To establish economic principles.
4.) To bring out minimal running time and throughput.

Primarily, the focal point of scheduling strategies is on efficiency, space, time and cost. The datacenter broker finds out and / or discovers the resources present in the network and gathers status information about them; in resource selection phase, the aimed resource is chosen as per the requirements of task and resource. This is a very crucial and deciding stage in task allocation following which the task is allocated to the selected resource for execution.

There have been several studies which attempted to describe the scheduling problem in cloud computing as the workflow problem; and based on the inferences it can be further divided into two levels as under:
1) Service-level (platform layer and static scheduling).
2) Task-level (unified resource layer and dynamic scheduling).

Recent studies have proposed QoS Aware Data Replication (QADR) and Minimum Cost and Maximum Flow (MCMF) algorithms to minimize the data replication cost. The node combination techniques were also presented to reduce the data replication time [3].

IV. TRACON

Ron C. Chiang and H. Howie Huang, et al (2014), [5] have discussed that the performance of a request running in a virtual machine (VM) should not be dependent on co-located/concurrent applications and Virtual Machines, that share the physical machine; however, undesirable interference effects exist and are especially stern for data intensive applications. In this research, TRACON, which is a novel Task and Resource Allocation CONtrol framework, was proposed that lessens the interference effects from co-located/concurrent data-intensive applications and considerably improves the overall system performance.

The existing approach focuses only on the load in the virtual machine for the appropriate application. Each virtual machine contains different number of applications. This approach focuses on the following problems to handle the request from application users.

1.) The **interference prediction model** gathers the application performance from the resource consumption observed/collection from multiple VMs.

2.) The **interference aware scheduler** is chiefly designed to utilize the model and generate the optimized assignments of tasks and physical resources.

3.) The **task and resource monitor** collects the application characteristics at the time of running and it feeds to both the model and scheduler.

But, it is not sufficient to apply the scheduling process in a successive manner.

The present approach is unaware of applications performed in the virtual machine for distributed environment.

The task and resource monitor collects the application characteristics at the time of running and it feeds to both the model and scheduler.

Selection of different type of application virtual machine to execute the process will cause the failure of task.

Without considering load and time for application execution in the virtual machine may increase the cost of task completion as well as reduce the quality of service.

V. Proposed System to improve the efficiency

To overcome the shortcomings of the existing system and to improve the efficiency of the cloud platform a novel arrangement has been proposed.

The proposed system categorizes the virtual environment based on the available application in each machine, to reduce the prediction error and time.
First, the sets of VM’s are created and then passed on to the data center. The computation of cost with the inclusion of VM capacity is the next important stage.

1.) Based on the cost of VMs, the clusters are formed. K-means clustering algorithm is used to form the clusters of VM’s. This environment reduces the scheduling delay in the virtual network using pair swapping clustering algorithm.

2.) Jobs prioritization on the basis of size and importance level supports the efficient scheduling during the demand condition.

3.) Allocation of jobs to the high capacity VMs according to the clusters is the final stage of proposed work.

4.) From this proposed system, the quick allocation of high-priority jobs and the optimal allocation of low and medium jobs would be achieved.

This environment reduces the scheduling delay in virtual network using pair swapping clustering algorithm. These clusters of VM’s would be formed by applying K-Means clustering algorithm.

So prior to moving on to any datacenters, first the sets of VM’s are created and then passed on to the datacenter, which is nearest and thus allowing the not only fast accessing of servers but also provides efficient deployment/utilization of existing resources.

Reallocation of VM’s in this way enhances performance of processor, memory and various network processes /operation by reducing workload on datacenters.

The incentive approach is used to predict the ratio of the application and to reduce the prediction error in an appropriate virtual machine. Hence, the waiting time, network time and execution time measures are considered for avoiding the time delay of application execution.

The dependency, between the different applications, is reduced by using service integration i.e. combining all the dependent applications in one and running them on one server.

Scheduling Strategies considering parameters are as under:

1.) Load
2.) Waiting Time, Execution Time, Network Time
3.) Delay

These parameters can be calculated using CloudSim simulator.

VI. Parameters

**Expected Waiting Time** is Process of computing the Queuing time for each job in the appropriate resource type machines. Waiting time will be computed by using the following mathematical formation. Expected waiting time computed by computing the load of previously allocated jobs in the yth machine having Ith resource type for a given job x divide by corresponding machine speed.

\[ EW_{WT_{x,y}} = \sum_{j=0}^{n-1} \frac{WL_j}{MS_{y,i}} \]

**Expected Computation Time** is Process of computing the execution time for each job in the appropriate resource type machines. Computation time will be computed by using the following mathematical formation. Expected computation time computed by
computing the load of current jobs divide by speed of the \( y \)th machine having \( I \)th resource type.

\[
ECT_{xy} = \frac{WL_j}{MS_{y_i}}
\]

**Network Transfer time** consist of Input size of the current Job and bandwidth between the broker and Current virtual machine

It will be computed by using the following equation

\[
NTT_{xy} = WL_j/BW_{y_i}
\]

**Network Transmission time** consist of output size of the current Job and bandwidth between the broker and Current virtual machine

It will be computed by using the following equation

\[
NTT_{xy} = WL_j/BW_{y_i}
\]

**Energy cost** for \( x \) job will be computed by using the energy consumption of the particular job in the \( y \)th virtual machine which one having the \( I \)th resource type and unit price for the energy utilization. Energy consumption for the machine will be computed by size of the \( x \)th job and energy consumption of the \( y \)th machine for a single million instruction.

\[
EC_{xy} = WL_x \times EY_{y_i} \times EP_x
\]

**Network cost** for \( x \) job will be computed by using the network consumption of the particular job in the \( y \)th virtual machine which one having the \( I \)th resource type and unit price for the network utilization. Network consumption for the machine will be computed by input and output size of the \( x \)th job and network consumption of the \( y \)th machine for a single million instruction.

\[
NC_{xy} = \left[ NTT_{xy} + NTT_{xy} \right] \times BP_{y_i}
\]

**Computation cost** will be consists of the execution cost for the \( x \)th job which one having the WL load executed in the \( y \)th virtual machine which one having the \( I \)th resource type.

**VII. CONCLUSIONS**

From this approach, we described that the Reallocation of VM’s in the proposed way enhances performance of processor, memory and various network processes /operation by reducing workload on datacenters. The proposed approach reduces the prediction error in an appropriate virtual machine, so, the waiting time, network time and execution time measures are considered for avoiding the time delay of application execution. Moreover, the dependency between the different applications are reduced by using service integration i.e. combining all the dependent applications in one and running them on one server.

**VIII. REFERENCES**


Applications in Virtualized Environments” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 5, MAY 2014
