Performance Analysis of multi-cloud Deployment in Many task Applications

K.Senthil, M.Tech., (Ph.D)
Vickram College of Engineering

Abstract

Cloud storage enables network online storage where data is stored on multiple virtual servers. In order to carry out huge tasks in cloud environment, single cloud provider is not sufficient to perform the many tasks applications and services. For Handling Intensive task, need to have multi cloud environment to improve the cost-effectiveness of the deployment and increase availability. Larger tasks are carried out by processing of many tasks at a time in a cloud computing environment. In this paper, for efficient handling of multiple tasks, need to have the performance analysis and optimization of all tasks in the multi-cloud environment. Performance analyses consist of CPU scheduling, Memory utilization, I/O tasks, and resource time sharing and cost benefits. Earlier system concentrates on the deployment of multi cloud architecture and multi-processing needs more accuracy, scalability and efficiency. In the methodology, Job allocation by front end server and service LAN are used. This research will achieved the process of multitasking environment in multi cloud infrastructure by having some effective tools for measuring over all performance and optimization of multi cloud computing services.

Keywords—Cloud computing, computing cluster, loosely coupled architecture, multiple task computing

I. INTRODUCTION

Cloud computing is a model for enable convenient, on-demand network access to a shared pool of configurable computing, that includes networks, servers, storage, applications, and services that can be rapidly provisioned and released with management effort or service provider interaction. The term "cloud", have its origin from the internet which has its schematic representation as a cloud. Cloud computing, as application or service, is not a new concept as it existed in the form of computer system remotely time-sharing computing resources and applications. Cloud computing refers to the many different types of services and applications being delivered in the internet cloud, the devices used to access these services and applications do not require any special applications.

Cloud has enormous advantages, but single cloud environment is not efficient for Many-task computing in order to handle the parallel processing of MTC Applications. Single-Cloud architecture is not capable for scientific applications and does not minimize the sharing of hardware, memory, resources and I/O Response and Cost optimizations. The main aim of the MTC Application is to implement the High Performance Computing characters in the parallel processing. The rest of the paper is organized as follows: Sec. II introduces the importance of multi cloud computing and many task computing applications. Sec. III proposes the allocation, scheduling of the job and nature of virtual machine allocation, Execution parallelization of each individual job. Sec. IV presents performance results and discussions. Finally, Sec. V concludes the paper with some future work.

A. Multi Cloud Computing

Multi-cloud is used to learn a new interface in every time and working in an integrated manner. The aim is to merge the multiple cloud providers. Multi-Cloud have different set of Application Programming Interfaces, resource, formats and encodings and each cloud which have the several simultaneous versions for a single cloud with various abstractions. Multi-cloud can handle the various kinds of Network architectures (VLANS, security groups, NAT, ACL) and Storage architecture (local/attachable disk, backup.)They are truly different applications, with different semantics. Benefits of Multi-Cloud Computing:

- Redundancy
- Disaster recovery
- Leverage unique cloud-specific services when needed
- Leverage public cloud cost benefits
- Leverage existing investments: private clouds
- Accuracy
- Scalability
- Fast Response time

B .Many task computing

Many Task Computing (MTC) defines different types of high-performance applications involving many different tasks, and requiring large number of computational resources over short periods of time. These tasks can be of very different nature, with sizes from small to large, loosely coupled or tightly coupled, or compute-intensive or data-intensive. Cloud computing technologies can offer important benefits for IT organizations and data centers running MTC applications with elasticity and rapid provisioning, enabling the organization to increase or decrease its infrastructure capacity within minutes and it can reduce or even eliminate their in-house infrastructures, resulting on a reduction in capital investment and personnel costs with different hardware configurations, operating systems, and software packages.

A. Objective

The scope of the proposed method is to analyze the performance of the multi cloud access in the many task applications with optimized scheduling algorithms allocated in the virtual machine.

B. Problem Statement

Cloud has enormous advantages, but single cloud is not efficient for Many-task Computing. Single-Cloud architecture is not capable and minimizes the sharing of hardware, memory and other resources.

II LITERATURE REVIEW

One of the main unresolved problems that arise during the access the of cloud information for many task applications. In this case, a complete understanding of the entire phenomenon requires that the cloud access should be viewed as a sequence of events. MTC applications are typically loosely coupled that are communication-intensive but not naturally expressed using standard message passing interface commonly found in high performance computing, drawing attention to the many computations that are heterogeneous but not parallel in manner.

C .Existing System

Single-Cloud architecture is not capable for scientific applications and does not minimize the sharing of hardware, memory, resources and I/O Response and Cost optimizations. The main aim of the MTC Application is to implement the High Performance Computing characters in the parallel processing. Existing system concentrates on the deployment of multi-cloud architecture. For having multi-cloud architecture, we need to process more than one task at a time.

Multi-processing needs more accuracy, scalability and efficiency. It lacks some performance measures.

D. Limitation of the Existing System

- Allocation of the jobs is irrelevant to the scheduling mechanisms
- Lack of accuracy, scalability and efficiency

E. Proposed System

Cloud services are entirely based on distributed computing, in order to achieve high performance computing (HPC), many Task computing will be implemented in the Multi-cloud Environment. Performance analysis will be carried over in the multi cloud services in order to complete multiple tasks in the MTC Applications. Performance analysis consists of CPU scheduling, Memory utilization, I/O tasks, resource time-sharing and Cost Benefit analysis. Here, the list of CPU jobs will be considered as jobs.

III METHODOLOGY
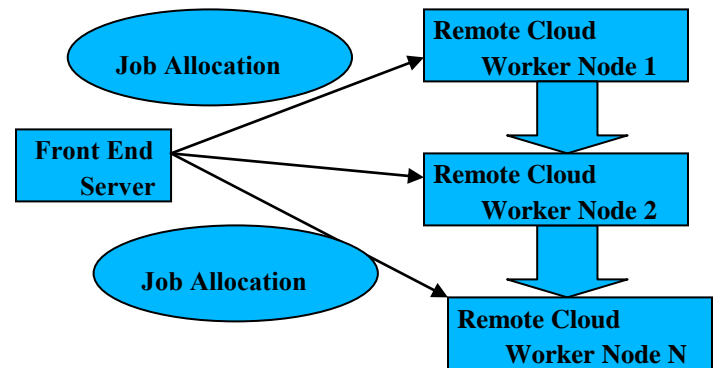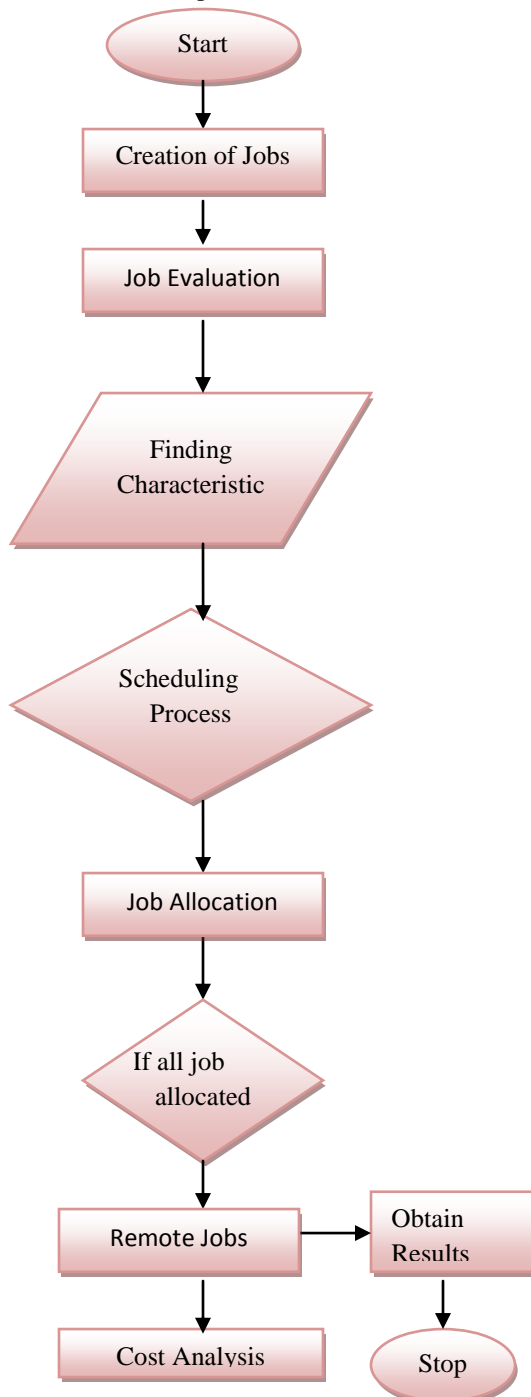
F: Architecture of the proposed System

Figure 1 Architecture of the Proposed System

In this methodology, Jobs Allocation by Front end server and Service LAN are used. A job means CPU tasks that will be allocated by front end server. Jobs are given in the form of request to the front end server. Front end server allocates the jobs to the remote cloud worker nodes which handles multitask processing of several jobs. Front end server and remote cloud worker nodes are connected by Service LAN. Remote worker cloud nodes are actually the cloud providers who provides and process the services. Each remote cloud worker starts from 1,2, 3 up to N no of the job allocations. So, result we are going to measure the performance of overall configurations of computing clusters.

Modules

- Job Creation
- VM allocation
- Job Scheduling
- Job Allocation
- Execution parallelization

```
              Start

         Creation of Jobs

          Job Evaluation

            Finding
          Characteristic

           Scheduling
            Process

           Job Allocation

           If all job
           allocated

    Remote Jobs -----> Obtain
                        Results

    Cost Analysis        Stop
```

The Multi-cloud Deployment have processed in the parallel fashion in loosely coupled system for executing the Many Task Applications and based on that process of the optimized performance analysis to be studied. The Process have been started with creation of Jobs, after that all the jobs will be evaluated based on many task applications. In the second part the Scheduling process is initiated by the scheduler and jobs to be allocated to the remote server for execution. After all process to be evaluated and executed the cost benefits analysis to be carried for each job performance in the Multi-cloud providers. Figure 2 Process flow of Job Allocation shows the process flow of jobs allocation. Those processes are Job Creation, VM allocation, Job Scheduling, Job Allocation, Execution Parallelization.

Job Creation: In the Job Creation, job will to be assigned to the cloud servers. All Jobs will be executed on the remote server and all jobs are selected by the clients dynamically. The Client should specify the execution time for every job.
- Jobs are referred to as CPU tasks.
- CPU tasks such as "Reading or counting no of lines, words and characters in a file", "Segmenting a image and calculating a time for the image segmentation".

Input

- Selecting a file

Output

- Show the properties the file (Memory, file path).
- Assign these tasks to server

VM allocation: Virtual machines are created on the remote machine. These virtual machines act as the virtual cloud server. The jobs allocated by the clients will be executed in the cloud servers. These virtual machines execute the jobs parallel manner.

Input

- Getting those files from client.

Output

- Executing the jobs (File Reading, Image segmentation) on the server.
- Send execution time of the Jobs executed on the server to client.

Job Scheduling: Jobs are allocated from the client machine. The jobs have to be schedule before allocation of the virtual machines. So, scheduling algorithm will be implemented in the client machines. The execution time of every job will be obtained based on their execution time scheduling process will be done.

Shortest-Job-First Scheduling

- A different approach to CPU scheduling is the shortest-job-first (SJF) scheduling algorithm. This algorithm associates with each process the length of the process's next CPU burst (CPU Execution Time).
- When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If the next CPU bursts of two processes are the same, FCFS scheduling is used.

Job Allocation: The properties of every VM will be obtained dynamically. Based on the loads of every VM the scheduled jobs will be allocated to every VMs. The jobs which are allocated in the VMs will be executed parallel and obtain the results

Properties of the VMs

- Memory Size
- CPU Time
- Before Execution
- After Execution
- Response Time
- I/O Response
- Cost Analysis

Execution Parallisation: Virtualized enterprise servers calls for more advanced Multi-Input Multi-Output (MIMO) control solutions. The jobs are allocated from each and every virtual machine. So, the jobs are executed in parallel manner

H: Algorithm Evulation

In the Jobs Scheduling process jobs are allocated from the client machine. The jobs have to be schedule before allocation of the virtual machines. So, scheduling algorithm will be implemented in the client machines. The execution time of every job will be obtained based on their execution time scheduling process will be done. Instead of guaranteeing optimal solution, these techniques aim to find reasonable solutions in a relatively short time. In this study, we used three rules of algorithms.

First Come First Serve (FCFS)

First Come First Serve (FCFS) or also known as First in First out (FIFO) is the simplest and the most fundamental of cloud scheduling that involves client-server interaction. In cloud scheduling, FCFS policy manages the jobs based on their arrival time, which means that the first job will be processed first without other biases or preferences.

Earliest Deadline First (EDF)

Earliest Deadline First (EDF) is a policy that schedule all the incoming jobs according to the specified due date or deadline. Incoming jobs will be processed or put in the queue based on the chronology indicate by the deadline. The job with the earliest deadline will be placed first in the processing queue.

Shortest Job First (SJF)

Shortest Job First (SJF) also known as Shortest Job Next (SJN) or Shortest Process Next (SPN) is a scheduling technique that selects the job with the smallest execution time. The jobs are queued with the smallest execution time placed first and the job with the longest execution time placed last and given the lowest priority. In theory, the best strategy to minimize the response time is to schedule the shortest job on the fastest resource. Since this policy gives preference to some groups of jobs over other group of jobs, this policy is unfair when compared to FCFS policy. Using SJF Optimal Scheduling should be reached easily and that maintains minimum average time for a given set of processes. SJF always gives minimum waiting time and turnaround time

Algorithm 1. FCFS

0 Add job(i) last in the main queue

Algorithm 2. SJF

0 for i=0 to i<main queue.size

1 if jobi+1 length < jobi length then

2 add job(i)+1 in front of jobi in the queue

3 end if

4 if main queue.size = 0 then

5 add jobi last in the main queue

6 end i

Algorithm 3. EDF

0 for i=0 to i<main queue.size

1 if jobi+1 due date < jobi due date then

2 add jobi+1 in front of jobi in the queue

3 end if

4 if main queue.size = 0 then

5 add jobi last in the main queue

6 end i

PERFORMANCE ANALYSIS

VMs is the way they let maximize resources. But with that comes increased tracking and monitoring to keep resource utilization in the "Goldilocks" zone. Part of VMs flexibility is the ability to grab resources such as memory form adjacent machines to temporarily meet increased demand. In a best practices model you would have your VMs set up with optimal sharing of resource pools to avoid those temporary events.
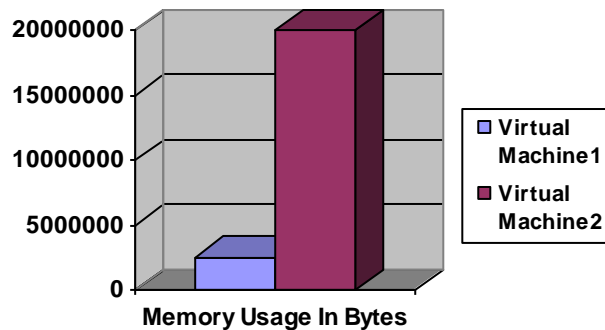

Figure:3 Performance of Memory Usage

Figure 3 described the memory usage of each job allocated in the virtual machine and which shows bytes allocated for job executed in the machine. The job allocated by client machine is "Reading or counting no of lines, words and characters in a file". The required bytes allocated for the reading the file and counting the file is 11,620,352.similary all the no of tasks will be allocated to the server by the client machine and performance to be compared while accessing the multi-cloud providers. Figure 4 shows the performance of execution time of process of the job. The job allocated by client machine is "Reading or counting no of lines, words and characters in a file" and this shows the performance between the jobs and milliseconds and which requires 1,406 milliseconds.
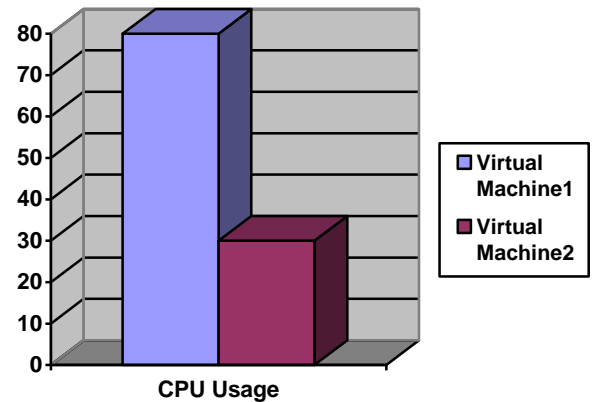

Figure:4 Performance of CPU Usage

II. CONCLUSION & FUTURE ENHANCEMENT

Cloud storage enables network online storage where data is stored on multiple virtual servers, generally hosted by third parties, rather than being hosted on dedicated servers. In order to carry out huge tasks in Cloud environment, single cloud provider is not sufficient to perform the many task Applications and services. For handling Intensive task, we need to have multi-cloud environment to improve the cost-effectiveness of the deployment and increase availability. Larger tasks are carried out by processing of many tasks at a time in a cloud computing environment. In order to handle intensive tasks in multi-cloud environment, we analyzed the performance of Multi-cloud computing services. Performance analyses consist of CPU scheduling, Memory utilization; I/O tasks, resource time-sharing and cost benefits. In the methodology, Jobs Allocation by Front end server and Service LAN are used. Future process concentrates on the secured processing of cloud computing data. We will achieve the process of multi-tasking environment in multi-cloud Infrastructure by having some effective tools for measuring over all performance and optimization of multi-cloud computing services. Earlier systems not focused on the security issues. Future process concentrates on the secured processing of cloud computing data. It also includes some effective performance analysis tools for cloud computing services.

REFERENCES

[1] I. Raicu, I. Foster, and Y. Zhao, "Many-Task Computing for Grids and Supercomputers," Proc. Workshop Many-Task Computing on Grids and Supercomputers, pp. 1-11, 2008.

[2] BioTeam "Howto: Unicluster and Amazon EC2," technical report, BioTeam Lab Summary, 2008.

[3] I. Llorente, R. Moreno-Vozmediano, and R. Montero, "Cloud Computing for On-Demand Grid Resource Provisioning," Advances in Parallel Computing, vol. 18, pp. 177-191, IOS Press, 2009.

[4] Amazon Elastic Compute Cloud, http://aws.amazon.com/ec2, 2010.

[5] ElasticHosts, http://www.elastichosts.com/, 2010.

[6] E. Walker, "The Real Cost of a CPU Hour," Computer, vol. 42, no. 4, pp. 35-41, Apr. 2009.

[7] M.A. Frumkin and R.F. Van der Wijngaart, "NAS Grid Benchmarks: A Tool for Grid Space Exploration," J. Cluster Computing, vol. 5, no. 3, pp. 247-255, 2002.

[8] R.S. Montero, R. Moreno-Vozmediano, and I.M. Llorente, "An Elasticity Model for High Throughput Computing Clusters," to be published in J. Parallel and Distributed Computing, doi: 10.1016/ j.jpdc.2010.05.005, 2010.

[9] E. Walker, J. Gardner, V. Litvin, and E. Turner, "Creating Personal Adaptive Clusters for Managing Scientific Jobs in a Distributed Computing Environment," Proc. IEEE Second Int'l Workshop Challenges of Large Applications in Distributed Environments (CLADE '06).

[10] I. Raicu, Y. Zhao, C. Dumitrescu, I. Foster, and M. Wilde, "Falkon: A Fast and Light-Weight TasK ExecutiON Farmework," Proc. IEEE/ACM Conf. SuperComputing, 2007.

[11] E. Huedo, R.S. Montero, and I.M. Llorente, "The GridWay Framework for Adaptive Scheduling and Execution on Grids," Scalable Computing—Practice and Experience, vol. 6, pp. 1-8, 2006.

[12] J. Chase, D. Irwin, L. Grit, J. Moore, and S. Sprenkle, "Dynamic Virtual Clusters in a Grid Site Manager," Proc. 12th IEEE Symp. High Performance Distributed Computing, 2003.

[13] P. Ruth, P. McGachey, and D. Xu, "VioCluster: Virtualization for Dynamic Computational Domains," Proc. IEEE Int'l Conf. Cluster Computing, 2005.

[14] I. Foster, T. Freeman, K. Keahey, D. Scheftner, B. Sotomayor, and X. Zhang, "Virtual Clusters for Grid Communities," Proc. Sixth IEEE Int'l Symp. Cluster Computing and the Grid, 2006.

[15] M. Murphy, B. Kagey, M. Fenn, and S. Goasguen, "Dynamic Provisioning of Virtual Organization Clusters," Proc. Ninth IEEE Int'l Symp. Cluster Computing and the Grid, 2009.