RESEARCH ARTICLE OPEN ACCESS

PRIORITY BASED SCHEDULING IN CLOUD COMPUTING BASED ON TASK – AWARE TECHNIQUE

Jeevithra.R *, Karthikeyan.T **
* (M.Phil Computer Science Student Department of Computer Science)
** (Associate Professor Department of Computer Science)
(Dr.M.G.R Chockalingam Arts College Arni-632317, Tamilnadu, India)

Abstract:

Cloud computing is an internet-based computing where resources, software and information are provided to computers on-demand, like a public utility. It is emerging as a platform for sharing resources like infrastructure, software and various applications. The majority of cloud computing infrastructure consists of reliable services delivered through data centers. Modern data centers, operating under the cloud computing model are hosting a variety of applications ranging from those that run for a few seconds to those that run for longer periods of time. When a job is submitted to the clouds, it is partitioned into several tasks. Scheduling a set of tasks in is one of the traditional challenges in parallel and distributed computing. It is very important to decide the order in which these tasks should be executed in order to increase the overall efficiency. Task scheduling is a key process for assigning the requests to resources in an efficient way considering cloud characteristics. Proper scheduling can have significant impact on the performance of the system. In this thesis work a task level scheduling approach has been proposed which prioritize the tasks to reduce the execution time, waiting time and response time. The simulator CloudSim has been used to validate the Proposed algorithm.

I. Cloud Computing

Cloud Computing is gaining in popularity, both in the academic world and in software industry. Cloud Computing include on-demand self service and dynamic scalability. As Cloud Computing utilizes pay-as-you-go payment solution, customers are offered fine grained costs for rented services. These advantages can be utilized to prevent web-applications from breaking during peak loads supporting end users. According to Amazon, provider of Amazon Web Service (AWS), a major Cloud Computing host: “much like plugging in a microwave in order to power it doesn’t require any knowledge of electricity, one should be able to plug in an application to the cloud in order to receive the power it needs to run, just like a utility”. Cloud Computing refers to both the applications delivered as services over the internet and systems software in the datacenters that provide those services. These datacenter hardware and software called as a Cloud.

II. Cloud Computing Service Models

Software-as-a-Service (SaaS). The SaaS service model offers the services as applications to the consumer, using standardized interfaces. The cloud provider is responsible for the management the application, operating systems and underlying infrastructure. The services run on top of a cloud infrastructure and are invisible for the consumer. The consumer can only control some of the user-specific application configuration settings.

Platform-as-a-Service (PaaS). The PaaS service model offers the services as operation and development platforms to the consumer. The consumer does not manage the underlying cloud infrastructure such as network, operating systems, servers, or storage, but only has control over the deployed applications and possibly application hosting environment configurations. The consumer can use the platform to develop and run his own applications, supported by a cloud-based infrastructure Examples: Microsoft Windows Azure, Google App Engine.

Infrastructure-as-a-Service (IaaS). The IaaS service model is the lowest service model in the technology stack, offering infrastructure resources as a service which includes raw data storage, processing power and network capacity. The consumer can the use IaaS based services to deploy his own operating systems and applications hence offer a wider variety of deployment possibilites for a consumer than the PaaS and SaaS models. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems; storage, deployed applications, and has limited control of select networking components (e.g., host firewalls). Example: Amazon EC2.

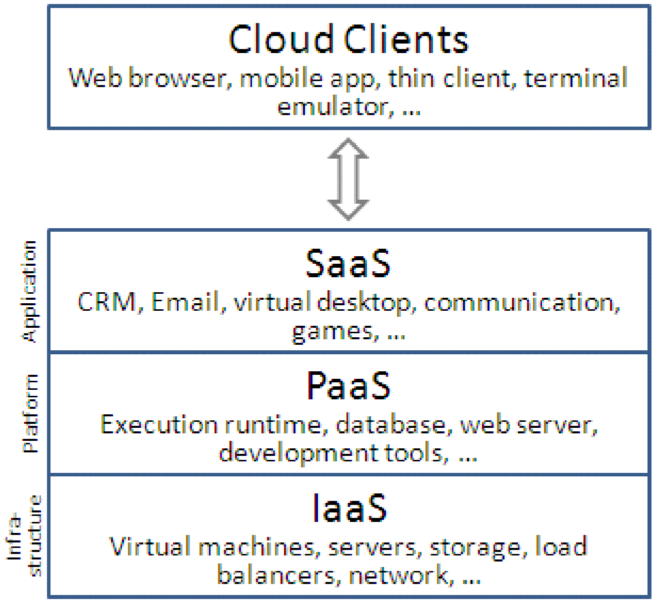

Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
Application
SaaS
CRM, Email, virtual desktop, communication, games, ...
Platform
PaaS
Execution runtime, database, web server, development tools, ...
Infra-structure
IaaS
Virtual machines, servers, storage, load balancers, network, ...

Figure 1.2 Cloud Computing Service Models

III. LITERATURE SURVEY

Virtualization

Virtual machine (VM) technology has recently emerged as an essential building- block for data centers and cluster systems, largely due to its capabilities of migrating and consolidating workload. Altogether, these features allow a data center to serve multiple users in a secure and efficient way. These virtualized infrastructures are considering a key component to drive the emerging Cloud Computing paradigm. Migration of virtual machines seeks to improve efficiency and fault tolerance of systems. More specially, the reasons that justify VM migration in a production system which includes: the need to balance system load, done by migrating VMs out of overloaded/overheated servers; and the need of selectively bringing servers down for maintenance after migrating their workload to other servers. The facility to migrate an entire operating system overcomes most difficulties that traditionally have made process-level migration a complex operation. The applications themselves and their corresponding processes do not need to be aware that a migration is occurring. Virtualization the Cloud Computing paradigm allows workloads to be deployed and scaled-out quickly through the rapid provisioning of virtual machines or physical machines. Any request of resources will be delivered by Cloud in terms of Virtual Machine. So placement of Virtual machine is most important part in Cloud Computing.

VM Migration

Cloud computing infrastructure consists of resource virtualization and resource scheduling based on service level agreement. Resource virtualization forms resource set by collecting resources that can be virtualized. This resource set is in turn converted to dynamic resource set that is utilized dynamically as demand arises. Scheduling of dynamic resources assigns resources to demands according to the service level agreement. The underlying technologies and issues for cloud computing, therefore, include resource virtualization, scalable resource management, and load balancing of resources across time and location, and quality of service. Efficient assignment and scheduling of resources is yet to be developed to dynamically match workload demands in the absence of a clear picture of the future usage. Human intervention, therefore, is indispensable and infrastructure administrators manually schedule and/or move virtualized resources to sustain the fluctuating demands, resulting in added burden on the already complex operation. Solutions to these pressing needs for cloud computing infrastructure hinge on virtual machine migration. VM migration is designed to move VMs from an overloaded physical machine to a lightly loaded machine. Moving VMs will lessen the burden on the overloaded machine while utilizing the idling physical machine. Numerous critical issues need to be addressed to make VM migration approach successful. Among the issues is an important parameter threshold that dictates what constitutes a machine underutilized or overloaded. If the overall resource utilization of a physical machine is over a certain fixed threshold, the machine is deemed overloaded and one or more of the VMs may be selected for migration.

IV. PROBLEM STATEMENT

Gap Analysis

Cloud computing is recently a booming area and has been emerging as a commercial reality in the information technology domain. However the technology is still not fully developed. There are still some areas which need improvements. One major area of concern is the Task scheduling. As large scale data processing is increasingly common in cloud computing systems, so it should be able to process the data efficiently in a given frame of time. The scheduling of tasks to the adaptable resources in accordance with adaptable time involves finding out a proper sequence in which tasks can be executed is needed. In such a scenario, tasks should be scheduled efficiently such that the response time, waiting time and execution time can be reduced. For applications such as message passing, parallel applications or multitier web applications, modeling in what order of priority tasks must be executed is very important. So to make effective use of the remarkable abilities of the cloud, efficient scheduling algorithms are mandatory. These scheduling algorithms are generally applied by cloud resource manager to optimally dispatch workloads to the cloud resources. Therefore extension of cloud simulation framework, NetworkCloudsim has been implemented which supports modeling of real cloud datacenter and applications like E-commerce and workflow. The problem of task scheduling is critical not only to achieve high Cloud performance, but also to satisfy various cloud consumers' demands in an equitable fashion thereby enhancing the overall performance of the cloud computing environment.

Need and Significance of Research

Workflow applications often require very complex execution environments. However, NetworkCloudsim uses only simple four stages in which NetworkCloudlet can be: Send, Receive, Executed and Finished. There is no level of priority used. But for scheduling multitier applications the currently implemented algorithm requires modifications as multitier applications are more event based and synchronized. So when workflow application are introduced in datacenter prioritization of tasks is needed to be done in such a way that datacenter works smoothly without the bottleneck of load, bandwidth and dependencies. So it is important to design a scheduling algorithm for scheduling of workloads in the Cloud environment so as to simultaneously minimize the execution time, response time and waiting time. Secondly task level scheduling, prioritization of tasks and provision of resources are important issues in both Grid and Cloud Computing, so there is a need to develop a new approach for task level scheduling to increase the overall efficiency.

V. PROPOSED STATEMENT

Solution to the Problem

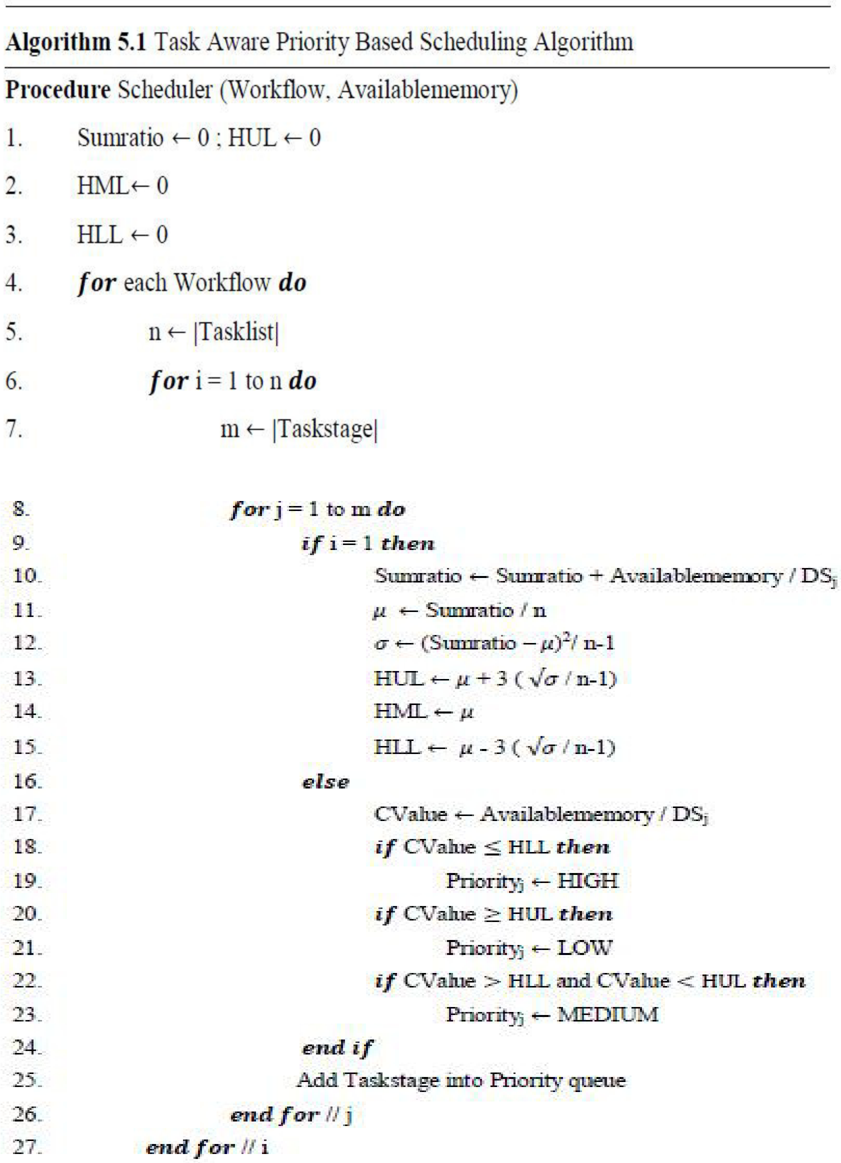
Workflows have been used to represent a variety of applications involving high processing. As mentioned in Gap Analysis and Literature Survey there is a need to enhance the tasks of a workload which must incorporate the concept of finding priority in which these tasks must be executed. The Cloud workflow applications often consists of several tasks and each task is implemented by several substitute Cloud services. QoS and resource utilization are necessary and play more important roles in Cloud service workflow applications. In order to efficiently and effectively schedule the tasks and data of applications among cloud services, end user QoS- based scheduling schemas should be implemented, such as those for minimizing total execution time, waiting time, response time and balancing the load of resources. So scheduler needs to take into account the computation stages of applications. Therefore, a new scheduler is formulated to schedule tasks.

- Priority scheduling approaches can be broadly classified into three categories:
- Task-level Fixed-Priority (TFP) scheduler assigns a fixed priority to all the jobs of each single task.
- Job-level Fixed-Priority (JFP) scheduler assigns a fixed priority to each single job, and a Job-level
- Dynamic-Priority (JDP) scheduler assigns a priority to each single job that can dynamically change over time.

The main advantage of task level scheduling is high performance computing and the best system throughput. A simulation framework which supports the modeling of essential data center inputs has been presented. The goal of workflow scheduling is spreading the load on processors and maximizing their utilization while minimizing the total task execution time. Its main purpose is to schedule tasks to the adaptable resources in accordance with adaptable time, which involves finding out a proper sequence in which tasks can be executed. The main focuses of this thesis is to understand the fundamentals of task level scheduling and based on that, improved priority assignment algorithm is proposed. A new approach is developed in which each tasklist is assigned a three level priority and then according to this priority tasklists are allowed to run.

Proposed Algorithm

In algorithm 5.1 HUL is historical upper limit, a real integer that stores the upper limit calculated based on historical data using X- bar chart upper limit formula. HML is the historical middle limit and HLL is historical lower limit calculated from the middle level and lower level X-bar chart formulas respectively. Workflow is divided into n number of task list. Each tasklist is divided in m task stages. Available memory is the RAM available for execution of task and DS is the size of data that is being executed.



Algorithm 5.1 Task Aware Priority Based Scheduling Algorithm

```
Procedure Scheduler (Workflow, Availablememory)
Sumratio ← 0 ; HUL ← 0
HML← 0
HLL ← 0
for each Workflow do
        n ← |Tasklist|
        for i = 1 to n do
                m ← |Taskstage|
                for j = 1 to m do
                        if i = 1 then
                                Sumratio ← Sumratio + Availablememory / $DS_j$
                                $\mu$ ← Sumratio / n
                                $\sigma$ ← $(\text{Sumratio} - \mu)^2$/ n-1
                                HUL ← $\mu$ + 3 ( $\sqrt{\sigma}$ / n-1)
                                HML ← $\mu$
                                HLL ← $\mu$ - 3 ( $\sqrt{\sigma}$ / n-1)
                        else
                                CValue ← Availablememory / $DS_j$
                                if CValue ≤ HLL then
                                        $\text{Priority}_j$ ← HIGH
                                if CValue ≥ HUL then
                                        $\text{Priority}_j$ ← LOW
                                if CValue > HLL and CValue < HUL then
                                        $\text{Priority}_j$ ← MEDIUM
                        end if
                        Add Taskstage into Priority queue
                end for // j
        end for // i
```

Algorithm takes workflows and available memory as input and gives priority queue as output. Historical upper level, historical middle level and historical lower level are calculated only for the first time. After calculation of these levels, the new values of memory/data ratios of other task stages are compared with these limits. If the values are less than equal to lower limit, they are assigned high priority as small tasks gets executed fast. If the values are greater than lower limit and lower than higher limit, they are assigned middle level priority and if the values are greater than higher limits, they are assigned low priority as they will take more time to execute, increasing the average waiting time and processor overhead which will decreasing the overall efficiency. After the calculation of priorities all the subtasks are inserted in the priority queue according to their priorities. By prioritizing the task stages the whole task is optimized.

VI. CONCLUSION AND FUTURE SCOPE

Task scheduling is a major issue in both large-scale parallel and distributed systems that which affects the system performance. In this thesis a scheduling algorithm for prioritization of tasks has been proposed. By prioritizing the tasks of a workflow the broker can have better control on which task should be processed first. X-bar chart is used to find the order priority for tasks execution. The comparative analysis has been done with previous random overlap algorithm. The comparative analysis depicted that there is a decrease in the execution time, waiting time and response time.

Future Scope

- Other factors should be taken into consideration such as bandwidth and cpu utilization.
- This concept can be extended for handling fault tolerance.
- This approach can also be extended for optimization of energy efficiency and to reduce power consumption.

VII. REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. d. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson and M. Zaharai, "A view of Cloud Computing," *ACM,* vol. 53, no. 4, pp. 50-58, April 2010.

[2] J. Varia, "Architecting for Cloud Computing:Best Practices," Amazon, January 2010.

[3] X. Wang, B. Wang and J. Wang, "Cloud Computing and its Key Techniques in Science and Automation Engineering (CSAE)," *IEEE,* vol. 2, pp. 404-410, 2010

[4] R. Clark, "A Break in the Clouds: Towards a Cloud Definition," *ACM,* vol. 39, no. 1, pp. 50-55, January 2009.

[5] G.Heiner, R.Sasse, E.Fuchs and H.Kopetz,"Time-Tiggered Architecture(TTA) Advances in Information Technology," 2011.

[6] B. Hay, K. Nance and M. Bishop, "Strom Cloud rising:Security Challenges for IaaS Cloud Computing," in *System Science*, Hawaii, 2011.

[7] Amazon SLA, 2010. [Online]. Available: aws.amazon.com/ec2-sla/.

[8] H. U, Rehm, Rueter and Wittmann, in *Grid and Cloud Computing*, August 2009, pp. 249-265.

[9] K. E, "Grid Compiting: Past, Present and Future- An Innovation Prespective," IBM white paper, 2006.

[10] L. v. Doorn, "Hardware Virtualization Trends," in *ACM*, Newyork, USA, 2006.

[11] R. Bhuyya, D. Abramson and J. Giddy, "A case for Economy Grid Architecture for Service-Oriented Grid Computing," in *10th IEEE International Heterogeneous Computing Wokshop*, California USA, 2001.

[12] A. Goscinski and M. Brock, "Towards Dynamic and Attribute based Publication, Discovery and Selection for Clouc Computing," *Future Generation Computer Systems,* vol. 26, no. 7, pp. 947-970, July 2010.

[13] Bardin, J. Callas, J. Chaput, S. Fusco and P. Gilbert, "Security Guidance for Critical Areas of Focus in Cloud Computing," in *Cloud Security Alliance*, January 2010.

[14] "Amazon Virtual Private Cloud (VPC)," Amazon, [Online]. Available: aws.amazon.com/vcp/. [Accessed 11 January 2013].

[15] S. Crosby and D. Brown, "The Virtualization Reality," *Queue,* vol. 4, no. 10, pp. 34-41, 2007.