GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

ONLINE RESOURCE ALLOCATION OPTIMIZATION AND WORKLOAD BALANCING SCHEDULING ON IAAS CLOUD SYSTEM

Deepika Saxena1, Dr. R.K. Chauhan2 and Shilpi Saxena3

1Assistant Professor, Department Of Computer Science And Applications, Kurukshetra University, Kurukshetra
2Professor, Department Of Computer Science And Applications, Kurukshetra University, Kurukshetra
3Assistant Professor, Department Of Computer Science And Applications, Graphic Era University, Dehradun

ABSTRACT

Cloud computing is a rapidly emerging paradigm in this very new era of technology. Basically, cloud is a cluster of distributed and interlinked servers providing on-demand services to customers. Broadly, it offers software-as-a-service (SAAS), platform-as-a-service (PAAS), infrastructure-as-a-service (IAAS). Here we are focusing on IAAS cloud system which offers computational resources to remote customers in the form of leases. Here we are defining real-time or online optimized scheduling of requests of various resources arriving simultaneously at data center of IAAS cloud service provider. Being more practical, our algorithm is providing best resource utilization and better results in terms of execution time as compared to DFPT algorithm for task scheduling in cloud computing which do not considers dependency between tasks (requests). Our algorithm proposes dynamic task allocation on IAAS clouds and resource scheduling by utilizing the updated status of various Virtual Machines available at real time. We have simulated this experiment using CloudSim toolkit. Surely, there is very beneficial improvement in results as compared to default FCFS scheduling and other available scheduling algorithms.

Keywords- Cloud, CloudSim, Dependency, Resource allocation, Task, Virtual Machine.

1. INTRODUCTION

In this new digital world of Information Technology and Advancement, the cloud computing has broaden the world of internet to all most everyone in this world. It has transformed the complex internet based services to the simplest laymen requirement of computation. The term `Cloud` means a cluster of interconnected distributed datacenters which are expanded worldwide for example, Google, Amazon etc. At each datacenter, there are several computer systems acting as servers and providing enormous variety of services to the customers. Briefly describing, cloud computing is a model for enabling convenient, on-demand network access to shared pool of computing resources for example, networks, servers, storage, applications and several other services that can be rapidly provisioned and released with minimal management effort or cloud service provider interaction[4]. The basic idea of cloud computing is based on a very fundamental principal of reusability of IT capabilities. In this paper, specifically considering IaaS clouds i.e. Infrastructure as a Service which represents lowest level of cloud service providing resources and services on pay-as-per use to its customers. The ultimate goal of this Cloud System is to instantly provide resources whenever customer requires it on their laptops, PCs, mobile phones etc wherever possible. In an IaaS model of cloud computing, Cloud Service Provider provides hardware, software, servers, storage and other infrastructure components to its customers. IaaS clouds also provides various applications to its users and handle task including system maintenance, backup and resiliency planning. IaaS customers pay on a per-use basis typically by hour, week or month. Some Cloud Service Provider also charge customers based on amount of storage space used or VM space used [5]. In recent trends of IT world, cloud computing is emerging with growing popularity of its capability to provide unlimited VM space, storage space and other resources with extreme flexibility so that no adjustment is required at customer end. But practically, there is no datacenter which has unlimited capacity. In fact, to meet real time significant demand of resources, Cloud Service Provider has to manage several arrangements and adjustments at their datacenters to concurrently schedule the thousand of requests and demand of resources arriving parallel at datacenter. In case of overflow of workload at datacenter, it can be shared among different datacenters or workload can be share in between public and private clouds with applying extreme security checks [2]. So, this
becomes a very challenging issue to schedule several requests simultaneously at datacenter satisfying all the customers, workload balancing, reduction of energy consumption, faster execution of requests at minimum cost etc. Clearly, default FCFS scheduling cannot provide satisfactory results in all the above mentioned respect. Therefore, there is rigorous demand of an optimized real-time scheduling technique. Though, many researchers have proposed different algorithms for scheduling in cloud computing which are providing good solutions in one or other respect, but none of them is able to completely satisfy all the different criteria of scheduling in cloud system. Broadly, in IaaS cloud system there is requirement of a scheduling technique which can satisfy all the different terms and conditions of real-time cloud computation. The scheduling technique should consider dependency and non-dependency among tasks or requests, bandwidth utility, data transfer rate, data transfer cost, response-time of VMs, workload balancing, resource sharing, CPU utilization, faster execution-time at minimum cost, how many VMs to be set up at one host computer system and many more aspects like that. Here, in our proposed work we have surely covered all these aspects and designed a satisfactory algorithm that is providing improved results in all respect.

II. LITRETURE SURVEY
Combined resource provisioning and scheduling strategy for executing scientific workflows on IaaS Clouds was presented which aims to minimize overall execution cost while meeting a user defined deadline [6]. This paper provides an optimized solution to scheduling problem of scientific workflows in cloud system. But it is considering single workflow execution at a time and not the multiple workflows. Various energy efficient strategies were studied which are based on approaches that specific plug-ins and energy control centers for networked large-scale hardware and software can be implemented and that they can reduce software and hardware related energy cost, improve load-balancing, reduce energy consumption due to communications, save CO2 emissions etc [1]. This paper has considered energy saving aspects only and not noticed the execution time, execution cost and dependency between tasks in cloud computing system. In case of significant client demands, it is necessary to share workload among multiple datacenters, this would provide cheaper resources and would enlarge the capacity of resource pool [2]. This paper is very flexible and is efficient in providing improved results for task scheduling in IaaS clouds. Also, they have designed two online dynamic scheduling algorithms i.e. DCLS and DCMSS for resource allocation mechanism. In DFPT algorithm [3] weighted fair priority queue is applied to schedule task according to deadline based criteria and minimum cost-based criteria. Though results are optimized and improved in this algorithm but it has limitations like it is not suitable dependent tasks, energy saving and load balancing criteria are not defined.

III. PROPOSED MODEL CONCEPTS
Our proposed model includes the following concepts: Here we assume that several requests are arriving at datacenter simultaneously, these requests are termed as tasks in cloud computing. In reality, most of the request shows dependencies upon one another, so here we have schedule the tasks according to their dependencies, also complexity of computation increases along with increasing dependencies. The set of tasks which includes dependencies represents applications as workflows.

Following steps are followed in our proposed model:

1. **Preparation of list of workflows** (or set of related tasks) submitted by several users at data center.
2. **Independent Tasks**: All independent task at same level can execute in parallel separately on selected VMs.
3. **Dependent Tasks**: All tasks which have output dependency on another task, their transfer time and transfer cost is calculated as follows:
   
   \[
   \text{Transfer Time} = \frac{\text{Data output}}{\text{bandwidth utilized for data transfer}}
   \]
   
   \[
   \text{Transfer Cost} = \frac{\text{Data output}}{\text{cost of bandwidth used}}
   \]
4. **Bandwidth** at same datacenter will cost same for data transfer, but if data is transferred to some external datacenter it will cost different and it will be high.
5. **Whenever a task is selected from the workflow for execution, it is mandatory to execute all its predecessor task first to resolve the dependency between tasks.**
6. **Execution Time and Execution Cost** of a single task is calculated as follows:
   
   \[
   \text{Execution Time} = \frac{\text{Task length}}{\text{Processing power of selected VM}}
   \]
Execution Cost = Cost of selected VM * Task length
Here task length is number of instructions of a task to be executed.

7. **Finally, Total Processing time and cost of each workflow is calculated as:**
   Total Processing Time = sum of execution time of each task in the task set + sum of all transfer time related to a task set.
   Total Processing Time = sum of execution cost of each task in the task set + sum of transfer cost of the task set.

8. **Resource allocation** is done using greedy approach of modified Cloud min-min algorithm, which selects suitable resource i.e. VM that execute the selected task at minimum cost and minimum time. Separate list of resource is maintained which keep track of the status of the VM by continuous VM monitoring at each resource allocation.

9. **Workload balancing is done as follows:** First of all, workload is calculated which depends upon the complexity and amount of computation involved in the task set. The complexity can be measured in terms of dependencies and communication among the tasks, amount of expected execution time based upon number of instructions in the task, memory space required for execution of task etc. If the calculated workload is higher than the capacity of single server, then workload is distributed among other servers at same datacenter, or if required among servers at different datacenter to balance the workload. Hence, workload is shared among distributed servers at same or different data center.

![DAG Diagrams](image-url)

**Figure 1.** Applications represented as DAG
Figure 1 and 2 are showing applications as submitted by the user, each application is a complex, so it is partitioned into set of task which are interconnected to each other in various respects, to show their relationship we have taken each application as DAG direct acyclic graph which clearly shows the dependency among tasks as possess by the real request at the cloud server at datacenter.

IV. METHODOLOGY
User submits applications to be executed on cloud server at datacenter as shown in figure 3. Each application is first partitioned into tasks in the form of DAG (Direct Acyclic Graph) based on dependency among tasks. Then these DAGs are placed in the list based on priorities (like deadline, cost, complexity etc) of execution using algorithm 1.

These priorities based list is then submitted at different cloud servers where they are scheduled and assigned to different VMs or resources using real-time resource allocation procedure as described in algorithm 2.
ALGORITHM 1: Preparation of task list based on priorities

Requirement – A DAG, expected execution time of each task in DAG
1. Deadline of every task is randomly generated.
2. Initially, all tasks are in list U.
3. Take task one by one from U and push current task node into stack S in decreasing order of their deadline.
4. While stack S is not empty do
5. If top(S) has unstacked immediate predecessor task then
6. Push immediate predecessor task node with least deadline into stack S.
7. Else push top of stack S into list P.
8. Pop top(S).
9. EndIf.
10. EndWhile.

Result - A list of task of P is prepared based on priorities.

ALGORITHM 2: Real-time resource allocation minimum cloud computing scheduling (RMCC)

Requirement – Priority based list of task P, n different VMs, execution-time matrix and data transfer-time matrix
2. While tasks not assigned do
3. T = top(P).
4. Get resource status information from all other DataCenterBroker(DCB) or manager servers.
5. Find resource or VM with minimum (VM_{min}) turnaround time (= waiting time + response time + execution time).
6. Assign task T to VM_{min}.
7. Remove T from P.
8. EndWhile.

Result – Selected task T is assigned for execution to VM with earliest finish time.

V. PERFORMANCE EVALUATION

In this section we have presented the experiments conducted in order to evaluate the performance of the proposed work. We utilized the CloudSim 3.1 tool to simulate the cloud environment and task sets are generated randomly using rand() function. The proposed work is compared with sequential, DFPT, EWSA algorithm for cloudlet scheduling. The configuration of the datacenter are as follows:

Table 1 shows the configuration of host and table 2 shows the configuration of VMs used in this proposed work. Here, we have used:

Number of processing element – 1

Number of host on each processor – 8 on each processor.
Table 1: Configuration of host

<table>
<thead>
<tr>
<th>RAM (MB)</th>
<th>2080</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing Power (MB)</td>
<td>220000</td>
</tr>
<tr>
<td>VM Scheduling</td>
<td>Time-shared</td>
</tr>
</tbody>
</table>

Table 2: Configuration of VMs

<table>
<thead>
<tr>
<th>VM</th>
<th>RAM</th>
<th>Processing Power (MIPS)</th>
<th>Processing Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM1</td>
<td>5024</td>
<td>22000</td>
<td>1</td>
</tr>
<tr>
<td>VM2</td>
<td>1048</td>
<td>11000</td>
<td>1</td>
</tr>
<tr>
<td>VM3</td>
<td>3308</td>
<td>22000</td>
<td>1</td>
</tr>
<tr>
<td>VM4</td>
<td>4604</td>
<td>32000</td>
<td>1</td>
</tr>
<tr>
<td>VM5</td>
<td>8028</td>
<td>55000</td>
<td>1</td>
</tr>
<tr>
<td>VM6</td>
<td>4000</td>
<td>41000</td>
<td>1</td>
</tr>
<tr>
<td>VM7</td>
<td>6024</td>
<td>44000</td>
<td>1</td>
</tr>
<tr>
<td>VM8</td>
<td>7028</td>
<td>62000</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3: System configuration

<table>
<thead>
<tr>
<th>Processor</th>
<th>Pentium Dual-core CPU <a href="mailto:T4400@2.20GHz">T4400@2.20GHz</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>3.00GB</td>
</tr>
<tr>
<td>System Types</td>
<td>64 bit operating system</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows 7</td>
</tr>
</tbody>
</table>

Performance with respect to time: The experimental results shows the remarkable improvement in time over sequential, DFPT and EWSA algorithms. The results have shown that proposed work is approximately 65.6% improved than FCFS, about 34.22% improved than DFPT and up to 22.68% improved than EWSA algorithm for task scheduling in cloud computing.

Table 4. Performance of proposed methodology with respect to time

<table>
<thead>
<tr>
<th>No. of task sets</th>
<th>FCFS</th>
<th>DFPT</th>
<th>EWSA</th>
<th>PROPOSED METHODOLOGY</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>803.649</td>
<td>366.281</td>
<td>359.441</td>
<td>280.6669</td>
</tr>
<tr>
<td>20</td>
<td>3202.96</td>
<td>1188.653</td>
<td>1146.111</td>
<td>660.5355</td>
</tr>
<tr>
<td>30</td>
<td>5564.02</td>
<td>3766.855</td>
<td>2881.666</td>
<td>1044.6681</td>
</tr>
<tr>
<td>40</td>
<td>7004.39</td>
<td>5044.666</td>
<td>3331.116</td>
<td>1846.4401</td>
</tr>
<tr>
<td>50</td>
<td>8963.38</td>
<td>6887.433</td>
<td>5842.556</td>
<td>3566.6664</td>
</tr>
<tr>
<td>60</td>
<td>10468.2</td>
<td>8113.998</td>
<td>7046.113</td>
<td>5442.1448</td>
</tr>
</tbody>
</table>
Figure 4. Graph showing comparison of execution time of proposed work RMCC with existing algorithms

Performance with respect to cost: The experimental results show the immense improvement in cost over sequential, DFPT and EWSA algorithms. The results have shown that proposed work is approximately 64% improved than FCFS, about 33% improved than DFPT and up to 23% improved than EWSA algorithm for task scheduling in cloud computing.

Table 5. Comparison of Cost of Execution of Proposed methodology V/s existing algorithms

<table>
<thead>
<tr>
<th>No. of task set</th>
<th>FCFS</th>
<th>DFPT</th>
<th>EWSA</th>
<th>PROPOSED METHODOLOGY</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>363.44</td>
<td>200.448</td>
<td>197.576</td>
<td>160.001</td>
</tr>
<tr>
<td>20</td>
<td>539.61</td>
<td>369.866</td>
<td>311.667</td>
<td>271.904</td>
</tr>
<tr>
<td>30</td>
<td>944.68</td>
<td>677.007</td>
<td>661.112</td>
<td>511.008</td>
</tr>
<tr>
<td>40</td>
<td>1248.5</td>
<td>721.481</td>
<td>624.081</td>
<td>441.169</td>
</tr>
<tr>
<td>50</td>
<td>1554.8</td>
<td>1098.22</td>
<td>976.116</td>
<td>771.816</td>
</tr>
<tr>
<td>60</td>
<td>1768.6</td>
<td>1544.61</td>
<td>1496.67</td>
<td>981.197</td>
</tr>
</tbody>
</table>

Figure 5. Graph showing comparison of cost of execution of proposed RMCC method with existing algorithms

Table 4 and Table 5 are showing performance evaluation of various number of task set against execution time and execution cost. Here, a task set can have one task, two tasks or up to five tasks having interdependency. A task set is generated at run time and dependency is also decided at execution time randomly. So we have calculated, the results
of various execution time of task set as the arithmetic mean of execution time obtained on running the task set twenty times. Each time result may somewhat vary depending upon dependency and number of task in task set.

But our proposed work will show significant advance results every time as compared to existing algorithms.

VI. CONCLUSION AND FUTURE WORK
IAAS cloud computing offers immense research work to do like data security, virtualization license management etc. But the scheduling is the topic of research in this field. Many well defined algorithms are proposed by many researchers, but none of them is suitable in all the way. In contrast, our proposed work is more dynamic and pragmatic that meets all the demands of perfect scheduling in IAAS. We have proposed RCMMS and workload balancing concepts and resolved dependency among task sets while minimizing execution time and execution cost. We have provided complete scenario of real time IAAS cloud scheduling and implemented it using CloudSim. The outcome results are showing tremendous improvement as compared to FCFS, DFPT[3] and EWSA[6] cloudlet scheduling algorithms.

In future more sophisticated and real algorithms can be research and work out for IAAS cloud system including better techniques for workload balancing, energy-saving techniques and considering contention problem at datacenters.

REFERENCES


