Impact Of Parallelism And Virtualization On Task Scheduling In Cloud Environment

Sanjay Kumar Sharma, Nagresh Kumar

Abstract: Today’s, cloud computing has placed itself in every field of the IT industry. It provides infrastructure, platform, and software as an amenity to users which are effortlessly available via the internet. It has a large number of users and has to deal with a large number of task executions which needs a suitable task scheduling algorithm. Virtualization and parallelism are the core components of cloud computing for optimization of resource utilization and to increase system performance. Virtualization is used to provide IT infrastructure on demand. To take more advantages of virtualization, parallel processing plays an important role to improve the system performance. But it is important to understand the mutual effect of parallel processing and virtualization on system performance. This research helps to better understand the effect of virtualization and parallel processing. In this paper, we studied the effect of parallelization and virtualization for the scheduling of tasks to optimize the time and cost. We found that virtualization along with parallelization boosts the system performance. The experimental study has been done using the CloudSim simulator.

Keywords: virtualization; parallel programming; distributed processing

1. INTRODUCTION

In the present world of information technology, cloud computing emerges as a new computing technology due to its economical and operational benefits. Cloud computing is able to perform the processing of an enormous amount of data using high computing capacity and distributed servers. Clients are facilitated to avail of this facility on the basis of pay-per-use policy. When the users need changes, the cloud server’s capacity scales up, and down to meet the user requirements. It is highly flexible, reduces capital expenditure, robust disaster recovery and can operate from anywhere through the internet. Users can avail these services by just submitting the request to the environment provided by the service provider. Parallel processing is a computing technique in which more than one process is executed simultaneously on different processors. Multiple processes solve a given problem efficiently by executing on multiple processors. The divide and conquer technique is used to divide a task into multiple subtasks. A parallel program written on the basis of the divide and conquer technique execute subtask on multiple processors. The requirement of high computation power cannot be full-fill by a single CPU. So parallel processing can improve the system computation power by increasing the number of CPUs. It is the best cost-effective technique to enhance system computation power. This technique can also be used in load balancing in cloud computing [1]. Virtualization is a key component of cloud computing. Virtualization is a mechanism that is used to create interactive environments like server, storage, operating system, desktop, etc, which is expected by the user. Hardware virtualization based on a set of software is also called a hypervisor or virtual machine manager (VMM). This software operates on each VM instance in complete isolation. A high-performance server, based on several machines needs suitable and customized software on demand. This approach helps to deploy tasks parallelly to the available resources on-demand, such as VMware, AmazonEC2, vCloud, RightScale, etc. [1][2][3][4]

- Sanjay Kumar Sharma, Department of Computer Science, Banasthali Vidyapith, Rajasthan, India. E-mail: skumar2.sharma@gmail.com
- Nagresh Kumar, Department of Computer Science, Banasthali Vidyapith, Rajasthan, India. Email: nagresh@gmail.com

Parallelism and virtualization are the key components of cloud computing. Now, the question is how parallelism and virtualization affect each other to improve system performance? Does it prevent or boost system performance? What are the other parameters which also affect the parallelism and virtualization? This research paper is organized as follows: the literature review of previous work is described in section-2. Section-3 contains the details about the proposed methodology for parallelization and virtualization. Section-4 describes the experimental setting and simulation result. Section-5 describes future work and conclusion.

2. RELATED WORK

Parallel architecture is an organization of resources to maximize resource utilization and improve system performance. The organisation of resources is categorised into three parts; shared memory processor (SMP), cluster parallel computers and hybrid. SMP is also called multiple processors in which each processor access a single shared memory. So, a task divided into subtask must be shared the same memory and communication between subtasks take place via shared variables. In a cluster parallel computers, more than one computing node with its own memory and processor are connected to each other. Hence two processor needs to send a message to each other node in order to exchange variable values. A hybrid architecture is just an interconnected SMP. Communication with SMP takes place through a variable value in shared memory but the SMP node communicates via message passing. Cluster architecture is highly scalable [5]. OpenMP is a standard application program interface (API) that consists of a set of directives and library subroutines [6]. OpenMPI is a shared memory multiprocessing programming environment that supports C/C++ and Fortran programming language. A program can write and execute a parallel program in the OpenMP environment by dividing the task into an independent task [7]. Virtualization is a technology that is used to provide better resource utilization. A different operating system can be installed on different virtual machines (VMs) to increase system flexibility. A hypervisor runs on the top of hardware. A hypervisor is also called a virtual machine manager (VMM). Hypervisor hides the actual physical hardware and provides virtual resources as per user expectations. The hypervisor is also responsible to protect hardware from...
malware unauthorised access. All the external requests are managed by the hypervisor. Virtualization and parallelization are the key components in cloud computing to improve system performance; it also placed some overhead on the complete system. The overhead percentage depends upon the host configuration. Perera et al. [8] compare the hypervisors VMware [9] and Xen [8]. Authors conclude that Xen is much better than VMware concern to paravirtualized and Xen is much better than Xen concern to fully-virtualized [10]. Tafa et al. [11] study to improve the system performance; it also placed some overhead on the complete system. The overhead percentage depends upon the host configuration. Perera et al. [8] compare the hypervisors VMware [9] and Xen [8]. Authors conclude that Xen is much better than VMware concern to paravirtualized and Xen is much better than Xen concern to fully-virtualized [10].

3. METHODOLOGY
For the experimental study, we use the CloudSim simulator and configure the hypervisor to analyse the parameters of parallelism and virtualization to improve the system performance.

VM Allocation Model
To exhibits, the simulation result on CloudSim simulator, space-shared and time-shared allocation policy for both cloudlets and VMs are provisioned [14]. The execution policy of cloudlets and VMs with time-shared and space-shared provisioning is clearly shown in fig.-1. In this figure, the execution of 8 tasks (t1, t2, ..., t8) is shown with a host with 2 CPU cores and two VMs per core.

In fig-1(a) space-shared provisioning is taken for cloudlets and VMs. In space shared only one VM can run at a time. Since each VM has two cores, so at a time two tasks can be executed simultaneously as shown in fig-1(a). According to space shared policy of VMs and cloudlets estimated finish time (eft) of a task p managed by the virtual machine i is given by

\[
eft(p) = est(p) + \frac{rl}{capacity \times cores(p)}
\]

Where est(p) is task execution start time and rl is a total number of instructions executed on a processor. The result eft(p) depends on the status of a task in the queue because tasks are placed in a ready queue if and only if the required number of free PEs are available in VMs. The total capacity of a host with np processing elements (PEs) is given by:

\[Capacity = \sum_{i=1}^{np} \frac{cap(i)}{np}\]

Where cap(i) is the processing strength of individual elements.

Fig.1: Provisioning policies for cloudlets and VMs
(a) Cloudlets: Space-shared, VMs: Space-shared (b) Cloudlets: Time-shared, VMs: Space-shared (c) Cloudlets: Space-shared, VMs: Time-shared (d) Cloudlets: Time-shared

VM Allocation Model
To exhibit, the simulation result on CloudSim simulator, space-shared and time-shared allocation policy for both cloudlets and VMs are provisioned [14]. The execution policy of cloudlets and VMs with time-shared and space-shared provisioning is clearly shown in fig.-1. In this figure, the execution of 8 tasks (t1, t2, ..., t8) is shown with a host with 2 CPU cores and two VMs per core.

In fig-1(a) space-shared provisioning is taken for cloudlets and VMs. In space shared only one VM can run at a time. Since each VM has two cores, so at a time two tasks can be executed simultaneously as shown in fig-1(a). According to space shared policy of VMs and cloudlets estimated finish time (eft) of a task p managed by the virtual machine i is given by

\[
eft(p) = est(p) + \frac{rl}{capacity \times cores(p)}
\]

Where est(p) is task execution start time and rl is a total number of instructions executed on a processor. The result eft(p) depends on the status of a task in the queue because tasks are placed in a ready queue if and only if the required number of free PEs are available in VMs. The total capacity of a host with np processing elements (PEs) is given by:

\[Capacity = \sum_{i=1}^{np} \frac{cap(i)}{np}\]

Where cap(i) is the processing strength of individual elements.
Datacentre Parameters
String arch = "x86"; // system architecture
String os = "Linux"; // operating system
String vmm = "Xen";
Number of host = 2

Host Parameters
int ram = 65535; //host memory (MB)
long storage = 1000000; //host storage
int bw = 10000;
core = each host is quad core

VM Parameters
long size = 10000; //image size (MB)
int ram = 512; //vm memory (MB)
in
nt mips = 1000;
long bw = 1000;
int pesNumber = n; //n number of cpus
String vmm = "Xen"; //VMM name

VM Scheduling Policy: Time shared and space shared.
Cloudlet Scheduling Policy: Time shared and space shared.

Simulation Results
Simulation results are recorded as average turnaround time (in simulator time unit) in four different tables as given below. Four different abbreviations are used as-

(i) SS-Space share for VMs and Space shared for cloudlets.
(ii) ST-Space share for VMs and Time shared for cloudlets.
(iii) TS-Time share for VMs and Space shared for cloudlets.
(iv) TT-Time share for VMs and Time shared for cloudlets.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Cloudlets=20,PE's/VM=1</th>
<th>Table 2</th>
<th>Cloudlets=20,PE's/VM=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prov VM</td>
<td>SS</td>
<td>ST</td>
<td>TS</td>
</tr>
<tr>
<td>2</td>
<td>5.6</td>
<td>5.6</td>
<td>10.1</td>
</tr>
<tr>
<td>4</td>
<td>3.1</td>
<td>3.1</td>
<td>5.1</td>
</tr>
<tr>
<td>8</td>
<td>1.9</td>
<td>1.9</td>
<td>2.7</td>
</tr>
<tr>
<td>16</td>
<td>1.9</td>
<td>1.9</td>
<td>2.7</td>
</tr>
<tr>
<td>32</td>
<td>1.9</td>
<td>1.9</td>
<td>2.7</td>
</tr>
<tr>
<td>64</td>
<td>1.9</td>
<td>1.9</td>
<td>2.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Cloudlets=40,PE's/VM=1</th>
<th>Table 4</th>
<th>Cloudlets=40,PE's/VM=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prov VM</td>
<td>SS</td>
<td>ST</td>
<td>TS</td>
</tr>
<tr>
<td>2</td>
<td>5.6</td>
<td>5.6</td>
<td>10.1</td>
</tr>
<tr>
<td>4</td>
<td>3.1</td>
<td>3.1</td>
<td>5.1</td>
</tr>
<tr>
<td>8</td>
<td>1.9</td>
<td>1.9</td>
<td>2.6</td>
</tr>
<tr>
<td>16</td>
<td>1.9</td>
<td>1.9</td>
<td>2.6</td>
</tr>
<tr>
<td>32</td>
<td>1.9</td>
<td>1.9</td>
<td>2.6</td>
</tr>
<tr>
<td>64</td>
<td>1.9</td>
<td>1.9</td>
<td>2.6</td>
</tr>
</tbody>
</table>

Note: We have used ‘Prov’ as a abbreviation for provision.
Space shared provision for cloudlets and VMs are more energy-efficient than time shared because there is a less number of switching occurs.

6. The divergence rate of time shared provisioning of VMs is faster than space shared provisioning as shown in all four graphs.

Now, it is clear from the graphs, recorded data in four different tables and result analysis that parallel computing reduces the average turnaround time and increases the overall system performance. The availability of multicore VMs results in independent execution of tasks and hence more speedup can be achieved. More virtualization generates extra overheads that should be minimized. A hypervisor is responsible to schedule the available physical resources and allocates them to different installed virtual machines. So a hypervisor must consider the points raised above. Selection of scheduling algorithm plays a key role in the hypervisor to increase system performance and better resource utilization. A suitable scheduling algorithm minimizes the waiting time of VM. This policy makes VM unaware of the existence of any competition from other VMs on the available resources. To schedule the task on the best resources, an optimization algorithm should be implemented in the hypervisor. Due to the heterogeneous nature of a large number of VMs and cloudlets an intelligent program is also needed for parallel execution.

## 5. CONCLUSION

Cloud computing is leading technology in the field of information technology. Virtualization and parallelization are the key components of cloud computing. Parallelization and virtualization definitely improve system performance. The maximum benefits of virtualization are taken by parallel execution of tasks on virtualized VM and parallelization is possible only through feasible virtualization. Parallelization and virtualization create extra overhead in cloud computing. In this paper, the mutual effect of virtualization and parallelization are properly analysed so that a hypervisor can take maximum benefits concern to average turnaround time. The next step is to overcome the extra overhead of virtualization and parallelization.

## REFERENCES


