M25COM

Cloud Computing and Distributed Technologies

Due date: 14th Dec 2018

Name: Madhurya Manjunath Mamulpet
ID: 8369749

Contents

CloudSim

1. Introduction

It is a new extensible simulation framework that enables modeling, simulation and experimentation of emerging Cloud Computing infrastructure(Calheiros *et al.*, 2010) and application service that enables developers to perform a test of their provisioning policies in a control and repeatable environment It helps to analyse the worst-case scenarios before the real-world deploys it.

2. Capabilities of CloudSim

Capabilities of Cloud computing supports simulation and modeling of cloud computing in a large-scale environment, that includes data centers in a single computing node physically.
It is a platform that is self-contained for modeling cloud, provisioning **self-service** and allocating policies. It supports simulation network connections compared to all other system elements.

Offers facility for simulation of federated cloud environment of resources that is inter-networked from public and private domains.
The virtualization engine that is available to help create and manage multiple virtualized services in a data center node.
The allocation of processing cores is flexible to switch to space -shared or time shared in virtualized services.
CloudSim is a stimulating tool that enables developers to perform a test of their provisioning policies in a control and repeatable environment It helps to analyse the worst-case scenarios before the real- world deploys it.

- Pooling resources
- Broad Network access on demand
- Self-service

3. CloudSim Architecture

CloudSim Architecture(Calheiros *et al.*, 2010) contains different computing levels
User level where applications are directly accessible by the end-users Example: Enterprise and Scientific.
middleware **SaaS** it contains software frameworks, like Web2.0 Interfaces, That help developer in creating cost-effective and user-interfaces.
Core middleware (**PaaS**) (Haider and Wahab, 2011)develops the platform services that includes run-time environment which helps in hosting and managing the user level application services
IaaS System level It is an existing large physical resource that powers the data centers
User code is the basic entities that include their specification, application requirements, number of users, VMs, type, and Scheduling policies.
Simulation layer of CloudSim I includes dedicated interfaces for storage, bandwidth, and VMs.
Related Classes(Kumar and Goudar, 2012)
CloudSim is the main class, that is responsible for event managing queues and step by step controlling e simulation events.
the future queue is a class that develops the future event queue used by CloudSim.
Deferred Queues the class that develops the deferred event queue accessible by CloudSim.

4. Comparison of various CloudSim

Features Compared	Package org.CloudSim. or bus.examples	Package org.CloudSim.orgbus. examples.network	Package org.CloudSim.orgbus. examples.network. datacenter
Data Centers	A datacentre is developed with one or two hosts, It is a resource class whose host List are visualized	A datacentre is developed that must run within cloudlets	Requires data center
Hosts		Contains a host associated with VM scheduler with cloudlets	Host packet that travels through the virtual network with a host
VM & CLOUDLETS	VM's are allocated scheduling policy as per provision.		-
PE (Processing Elements)	There is two type of time shared allows sharing of processing elesments by multiple VMs and spaces shared doesn't pallow sharing of PEs.	-	-
Scheduling Policy		-	Network Cloudlet Space Shared Scheduler develops a scheduling policy performed by a VM
Network & Topologies	It includes structured network packets	It contains a network topology Topological node is a network node that restores information from a topological-generated file. It contains delay logical information for the network	Edge switch enables to simulate edge switch for data center network

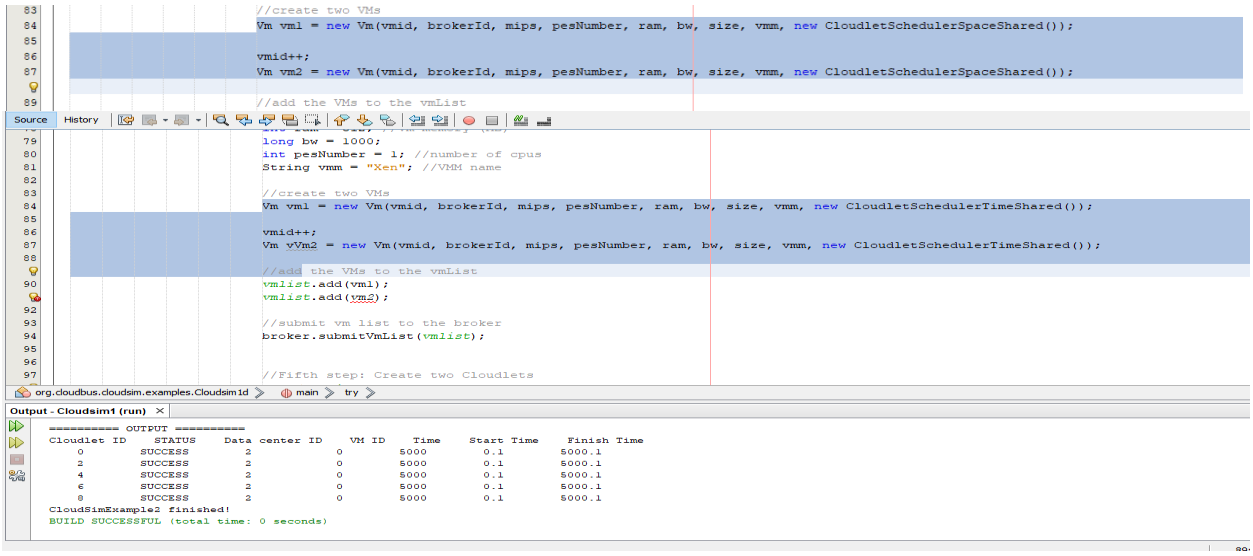
Table 1 – Comparison of various Cloudsim

Package org.CloudSim.orgbus.examples
This class includes a structured **network packet**(Calheiros *et al.*, 2010)
The data center is a resource class whose **host List** are visualized
Characteristics of data center represent static properties such as architecture of a resources, Operating System (OS), **management policy** , time zone along which the resource is located along resource configuration.
There is two type of time shared **allows sharing of Processing elements by multiple Virtual machines**.and spaces shared doesn't allow sharing of PEs.
Package org.CloudSim.orgbus.examples.network
The **topological graph** is that class that represents a graph containing nodes and edges, that are used for input with a network layer.
Topological link is a class that s and **link (edge**) from a graph
Topological node is a network node that revives information from a generated file.
Package org.CloudSim.orgbus.examples.network.datacenter
Edge switch enables to simulate **edge switch** for data center network
Host packet a data packet which passes through the **virtual network** with a host
Network **Cloudlet Space Shared Scheduler** develops a scheduling policy is performed by VM machine

5. 2a -simulation

---

a) Space-Space



b) Space-Time

```
Source  History
74  //VM description
75  int vmid = 0;
76  int mips = 250;
77  long size = 10000; //image size (MB)
78  int ram = 512; //vm memory (MB)
79  long bw = 1000;
80  int pesNumber = 1; //number of cpus
81  String vmm = "Xen"; //VMM name
82
83  //create two VMs
84  Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerSpaceShared());
85
86  vmid++;
87  Vm vm2 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerSpaceShared());
88
89  //add the VMs to the vmList
90  vmlist.add(vm1);
91  vmlist.add(vm2);
92
```

```
                new Host(
                        hostId,
                        new RamProvisionerSimple(ram),
                        new BwProvisionerSimple(bw),
                        storage,
                        peList,
                        new VmSchedulerSpaceShared(peList)
                )
```

```
========== OUTPUT ==========
Cloudlet ID     STATUS      Data center ID     VM ID     Time     Start Time     Finish Time
     0          SUCCESS          2              0        1000        0.1          1000.1
     1          SUCCESS          2              1        1000        0.1          1000.1
     2          SUCCESS          2              0        1000        1000.1          2000.1
     3          SUCCESS          2              1        1000        1000.1          2000.1
     4          SUCCESS          2              0        1000        2000.1          3000.1
     5          SUCCESS          2              1        1000        2000.1          3000.1
     6          SUCCESS          2              0        1000        3000.1          4000.1
     7          SUCCESS          2              1        1000        3000.1          4000.1
     8          SUCCESS          2              0        1000        4000.1          5000.1
     9          SUCCESS          2              1        1000        4000.1          5000.1
CloudSim2c finished!
BUILD SUCCESSFUL (total time: 0 seconds)
```

c) Time-Space

```
Source  History
215             int bw = 10000;
216
217             hostList.add(
218                     new Host(
219                             hostId,
220                             new RamProvisionerSimple(ram),
221                             new BwProvisionerSimple(bw),
222                             storage,
223                             peList,
224                             new VmSchedulerTimeShared(peList)
225                     )
226             ); // This is our machine
227
228
229             // 5. Create a DatacenterCharacteristics object that stores the
230             //    properties of a data center: architecture, OS, list of
231             //    Machines, allocation policy: time- or space-shared, time zone
232             //    and its price (G$/Pe time unit).
233             String arch = "x86";      // system architecture
```

```
org.cloudbus.cloudsim.examples.Cloudsim1b > createDatacenter >
Output - Cloudsim1 (run)
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
     0         SUCCESS         2             0      1000       0.1          1000.1
     1         SUCCESS         2             1      1000       0.1          1000.1
     2         SUCCESS         2             0      1000       1000.1          2000.1
     3         SUCCESS         2             1      1000       1000.1          2000.1
     4         SUCCESS         2             0      1000       2000.1          3000.1
     5         SUCCESS         2             1      1000       2000.1          3000.1
     6         SUCCESS         2             0      1000       3000.1          4000.1
```

d) Time-Time

```
Source  History
211             //4. Create Host with its id and list of PEs and add them to the list of machines
212             int hostId=0;
213             int ram = 2048; //host memory (MB)
214             long storage = 1000000; //host storage
215             int bw = 10000;
216
217             hostList.add(
218                     new Host(
219                             hostId,
220                             new RamProvisionerSimple(ram),
221                             new BwProvisionerSimple(bw),
222                             storage,
223                             peList,
224                             new VmSchedulerSpaceShared(peList)
225                     )
226             ); // This is our machine
227
228
```

```
org.cloudbus.cloudsim.examples.Cloudsim1d > createDatacenter >
Output - Cloudsim1 (run)
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
     0         SUCCESS         2             0      5000       0.1          5000.1
     2         SUCCESS         2             0      5000       0.1          5000.1
     4         SUCCESS         2             0      5000       0.1          5000.1
     6         SUCCESS         2             0      5000       0.1          5000.1
     8         SUCCESS         2             0      5000       0.1          5000.1
CloudSimExample2 finished!
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
Source  History
88                  String vmm = "Xen"; //VMM name
89
90                  //create two VMs
91                  Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared());
92
93                  vmid++;
94                  Vm vm2 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared());
95
96                  //add the VMs to the vmList
97                  vmlist.add(vm1);
98                  vmlist.add(vm2);
99
100                 //submit vm list to the broker
101                 broker.submitVmList(vmlist);
102
103
104                 //Fifth step: Create two Cloudlets
                    cloudletList = new ArrayList<Cloudlet>();
106
```

6. 2b simulation

---

a) Time-Time

```
                    Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared());
85
86                  vmid++;
87                  Vm vm2 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared());
88
89                  //add the VMs to the vmList
90                  vmlist.add(vm1);
91                  vmlist.add(vm2);
```

```
========== OUTPUT ==========
Cloudlet ID     STATUS     Data center ID     VM ID     Time     Start Time     Finish Time
     0          SUCCESS          2               0          1000          0.1          1000.1
     1          SUCCESS          2               1          1000          0.1          1000.1
     2          SUCCESS          2               0          1000          1000.1          2000.1
     3          SUCCESS          2               1          1000          1000.1          2000.1
     4          SUCCESS          2               0          1000          2000.1          3000.1
     5          SUCCESS          2               1          1000          2000.1          3000.1
     6          SUCCESS          2               0          1000          3000.1          4000.1
     7          SUCCESS          2               1          1000          3000.1          4000.1
     8          SUCCESS          2               0          1000          4000.1          5000.1
     9          SUCCESS          2               1          1000          4000.1          5000.1
CloudSim2c finished!
BUILD SUCCESSFUL (total time: 0 seconds)
```

b) Space-Space

```
                //create two VMs
84              Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerSpaceShared());
85
86              vmid++;
                Vm vm2 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerSpaceShared());
88
89              //add the VMs to the vmList
90              vmlist.add(vm1);
91              vmlist.add(vm2);
```

```
209                 hostList.add(
210                         new Host(
211                                 hostId,
212                                 new RamProvisionerSimple(ram),
213                                 new BwProvisionerSimple(bw),
214                                 storage,
215                                 peList,
216                                 new VmSchedulerSpaceShared(peList)
217                         )
```

```
========== OUTPUT ==========
Cloudlet ID     STATUS     Data center ID     VM ID     Time     Start Time     Finish Time
     0          SUCCESS          2               0         1000         0.1          1000.1
     1          SUCCESS          2               1         1000         0.1          1000.1
     2          SUCCESS          2               0         1000         1000.1          2000.1
     3          SUCCESS          2               1         1000         1000.1          2000.1
     4          SUCCESS          2               0         1000         2000.1          3000.1
     5          SUCCESS          2               1         1000         2000.1          3000.1
     6          SUCCESS          2               0         1000         3000.1          4000.1
     7          SUCCESS          2               1         1000         3000.1          4000.1
     8          SUCCESS          2               0         1000         4000.1          5000.1
     9          SUCCESS          2               1         1000         4000.1          5000.1
CloudSim2b finished!
BUILD SUCCESSFUL (total time: 0 seconds)
```

c) Space-Time

```
84                  Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerSpaceShared());
85
86                  vmid++;
87                  Vm vm2 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerSpaceShared());
88
89                  //add the VMs to the vmList
90                  vmlist.add(vm1);
91                  vmlist.add(vm2);
```

```
========== OUTPUT ==========
Cloudlet ID     STATUS     Data center ID     VM ID     Time     Start Time     Finish Time
     0          SUCCESS          2               0         5000         0.1          5000.1
     2          SUCCESS          2               0         5000         0.1          5000.1
     4          SUCCESS          2               0         5000         0.1          5000.1
     6          SUCCESS          2               0         5000         0.1          5000.1
     8          SUCCESS          2               0         5000         0.1          5000.1
     1          SUCCESS          2               1         5000         0.1          5000.1
     3          SUCCESS          2               1         5000         0.1          5000.1
     5          SUCCESS          2               1         5000         0.1          5000.1
     7          SUCCESS          2               1         5000         0.1          5000.1
     9          SUCCESS          2               1         5000         0.1          5000.1
CloudSim2a finished!
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
206                 // 3. Create PEs and add these into a list.
207                 peList.add(new Pe(0, new PeProvisionerSimple(mips)));
                    peList.add(new Pe(1, new PeProvisionerSimple(mips)));// need
209
210                 //4. Create Host with its id and list of PEs and add them to
211                 int hostId=0;
212                 int ram = 2048; //host memory (MB)
213                 long storage = 1000000; //host storage
214                 int bw = 10000;
215
216                 hostList.add(
217                                 new Host(
218                                         hostId,
219                                         new RamProvisionerSimple(ram),
220                                         new BwProvisionerSimple(bw),
221                                         storage,
222                                         peList,
223                                         new VmSchedulerTimeShared(peList)
224                                 )
225                         ); // This is our machine
```

d) Time-Space

```
91                  Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared());
92
93                  vmid++;
94                  Vm vm2 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared());
95
96                  //add the VMs to the vmList
97                  vmlist.add(vm1);
98                  vmlist.add(vm2);
```

```
========== OUTPUT ==========
Cloudlet ID     STATUS     Data center ID     VM ID     Time     Start Time     Finish Time
     0          SUCCESS          2              0        2500        0.1          2500.1
     1          SUCCESS          2              0        2500        0.1          2500.1
     2          SUCCESS          2              0        2500        0.1          2500.1
     3          SUCCESS          2              0        2500        0.1          2500.1
     4          SUCCESS          2              0        2500        0.1          2500.1
     5          SUCCESS          2              1        2500        0.1          2500.1
     6          SUCCESS          2              1        2500        0.1          2500.1
     7          SUCCESS          2              1        2500        0.1          2500.1
     8          SUCCESS          2              1        2500        0.1          2500.1
     9          SUCCESS          2              1        2500        0.1          2500.1
CloudSim2d finished!
BUILD SUCCESSFUL (total time: 0 seconds)
```

7. 2c Simulation

a) Time-Time

```
213                 hostList.add(
214                         new Host(
215                                 hostId,
216                                 new RamProvisionerSimple(ram),
217                                 new BwProvisionerSimple(bw),
218                                 storage,
219                                 peList,
220                                 new VmSchedulerTimeShared(peList)
221                         )
222                 ); // This is our first machine
223
224                 //create another machine in the Data center
                    List<Pe> peList2 = new ArrayList<Pe>();
226
227                 peList2.add(new Pe(0, new PeProvisionerSimple(mips)));
228
229                 hostId++;
230
231                 hostList.add(
232                         new Host(
233                                 hostId,
234                                 new RamProvisionerSimple(ram),
235                                 new BwProvisionerSimple(bw),
236                                 storage,
237                                 peList2,
238                                 new VmSchedulerTimeShared(peList2)
239                         )
                    ); // This is our second machine
241
```

```
88          //create two VMs
            Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared());
90
91          //the second VM will have twice the priority of VM1 and so will receive twice CPU time
92          vmid++;
93          Vm vm2 = new Vm(vmid, brokerId, mips * 2, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared())
94
95          //add the VMs to the vmList
96          vmlist.add(vm1);
```

```
========== OUTPUT ==========
Cloudlet ID     STATUS     Data center ID     VM ID     Time     Start Time     Finish Time
     5          SUCCESS          2              1        400         0.1          400.1
     6          SUCCESS          2              1        400         0.1          400.1
     7          SUCCESS          2              1        400         0.1          400.1
     8          SUCCESS          2              1        400         0.1          400.1
     9          SUCCESS          2              1        400         0.1          400.1
     0          SUCCESS          2              0        800         0.1          800.1
     1          SUCCESS          2              0        800         0.1          800.1
     2          SUCCESS          2              0        800         0.1          800.1
     3          SUCCESS          2              0        800         0.1          800.1
     4          SUCCESS          2              0        800         0.1          800.1
CloudSim3a finished!
BUILD SUCCESSFUL (total time: 0 seconds)
```

b) Space-Time

```
        Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerSpaceShared());

        //the second VM will have twice the priority of VM1 and so will receive twice CPU time
        vmid++;
        Vm vm2 = new Vm(vmid, brokerId, mips * 2, pesNumber, ram, bw, size, vmm, new CloudletSchedulerSpaceShared());
```

```
                hostList.add(
                        new Host(
                                hostId,
                                new RamProvisionerSimple(ram),
                                new BwProvisionerSimple(bw),
                                storage,
                                peList,
                                new VmSchedulerTimeShared(peList)
                        )
                ); // This is our first machine

                //create another machine in the Data center
                List<Pe> peList2 = new ArrayList<Pe>();

                peList2.add(new Pe(0, new PeProvisionerSimple(mips)));

                hostId++;

                hostList.add(
                        new Host(
                                hostId,
                                new RamProvisionerSimple(ram),
                                new BwProvisionerSimple(bw),
                                storage,
                                peList2,
                                new VmSchedulerTimeShared(peList2)
                        )
                ); // This is our second machine
```

```
========== OUTPUT ==========
Cloudlet ID     STATUS     Data center ID     VM ID     Time     Start Time     Finish Time
     5          SUCCESS          2               1         80         0.1          80.1
     0          SUCCESS          2               0        160         0.1         160.1
     6          SUCCESS          2               1         80        80.1         160.1
     7          SUCCESS          2               1         80       160.1         240.1
     1          SUCCESS          2               0        160       160.1         320.1
     8          SUCCESS          2               1         80       240.1         320.1
     9          SUCCESS          2               1         80       320.1         400.1
     2          SUCCESS          2               0        160       320.1         480.1
     3          SUCCESS          2               0        160       480.1         640.1
     4          SUCCESS          2               0        160       640.1         800.1
CloudSim3b finished!
BUILD SUCCESSFUL (total time: 0 seconds)
```

c) Space-Space

```
                hostList.add(
                        new Host(
                                hostId,
                                new RamProvisionerSimple(ram),
                                new BwProvisionerSimple(bw),
                                storage,
                                peList,
                                new VmSchedulerSpaceShared(peList)
                        )
                ); // This is our first machine

                //create another machine in the Data center
                List<Pe> peList2 = new ArrayList<Pe>();

                peList2.add(new Pe(0, new PeProvisionerSimple(mips)));

                hostId++;

                hostList.add(
                        new Host(
                                hostId,
                                new RamProvisionerSimple(ram),
                                new BwProvisionerSimple(bw),
                                storage,
                                peList2,
                                new VmSchedulerSpaceShared(peList2)
                        )
```

```
90          //create two VMs
91          Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerSpaceShared());
92
93          //the second VM will have twice the priority of VM1 and so will receive twice CPU time
94          vmid++;
95          Vm vm2 = new Vm(vmid, brokerId, mips * 2, pesNumber, ram, bw, size, vmm, new CloudletSchedulerSpaceShared());
96
```

```
========== OUTPUT ==========
Cloudlet ID     STATUS     Data center ID     VM ID     Time     Start Time     Finish Time
     5          SUCCESS          2              1         80          0.1          80.1
     0          SUCCESS          2              0         160          0.1          160.1
     6          SUCCESS          2              1         80          80.1          160.1
     7          SUCCESS          2              1         80          160.1          240.1
     1          SUCCESS          2              0         160          160.1          320.1
     8          SUCCESS          2              1         80          240.1          320.1
     9          SUCCESS          2              1         80          320.1          400.1
     2          SUCCESS          2              0         160          320.1          480.1
     3          SUCCESS          2              0         160          480.1          640.1
     4          SUCCESS          2              0         160          640.1          800.1
CloudSim3c finished!
BUILD SUCCESSFUL (total time: 0 seconds)
```

d) Time-Space

```
88          //create two VMs
            Vm vm1 = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared());
90
91          //the second VM will have twice the priority of VM1 and so will receive twice CPU time
92          vmid++;
93          Vm vm2 = new Vm(vmid, brokerId, mips * 2, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared())
94
95          //add the VMs to the vmList
96          vmlist.add(vm1);
215                     hostList.add(
216                             new Host(
217                                     hostId,
218                                     new RamProvisionerSimple(ram),
219                                     new BwProvisionerSimple(bw),
220                                     storage,
221                                     peList,
222                                     new VmSchedulerTimeShared(peList)
223                             )
224                     ); // This is our first machine
225
226                     //create another machine in the Data center
                        List<Pe> peList2 = new ArrayList<Pe>();
228
229                     peList2.add(new Pe(0, new PeProvisionerSimple(mips)));
230
231                     hostId++;
232
233                     hostList.add(
234                             new Host(
235                                     hostId,
236                                     new RamProvisionerSimple(ram),
237                                     new BwProvisionerSimple(bw),
238                                     storage,
239                                     peList2,
240                                     new VmSchedulerTimeShared(peList2)
241                             )
```

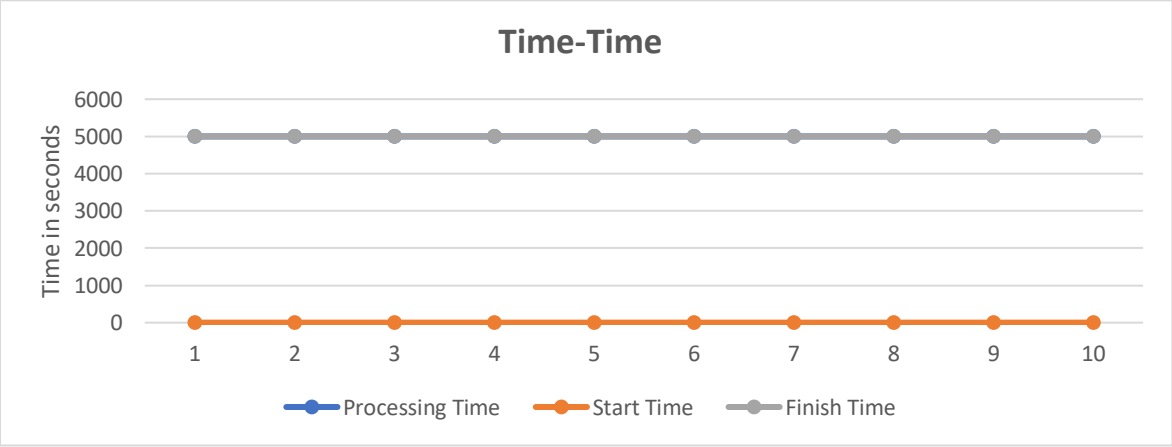
8. Result Analysis

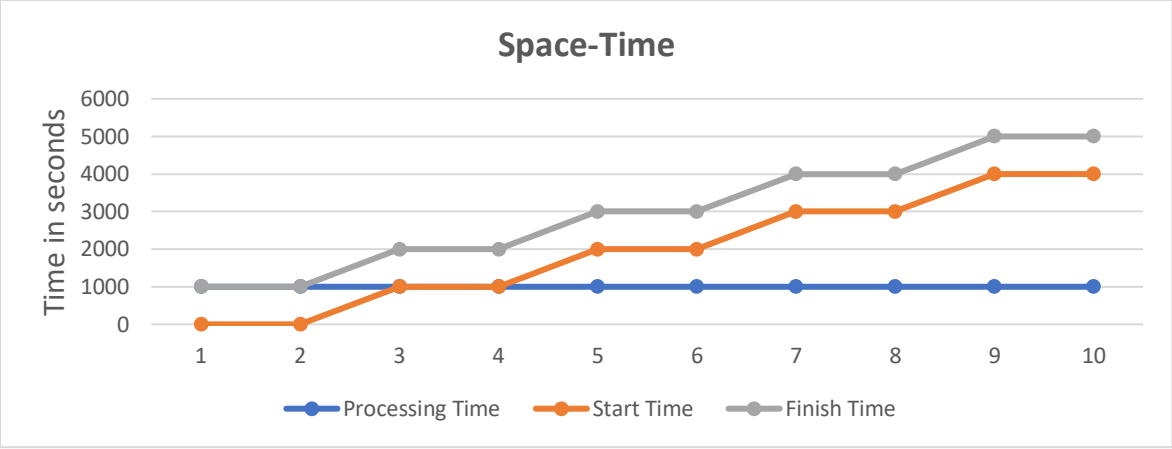
Task1-(3) Results Presentation and Discussion

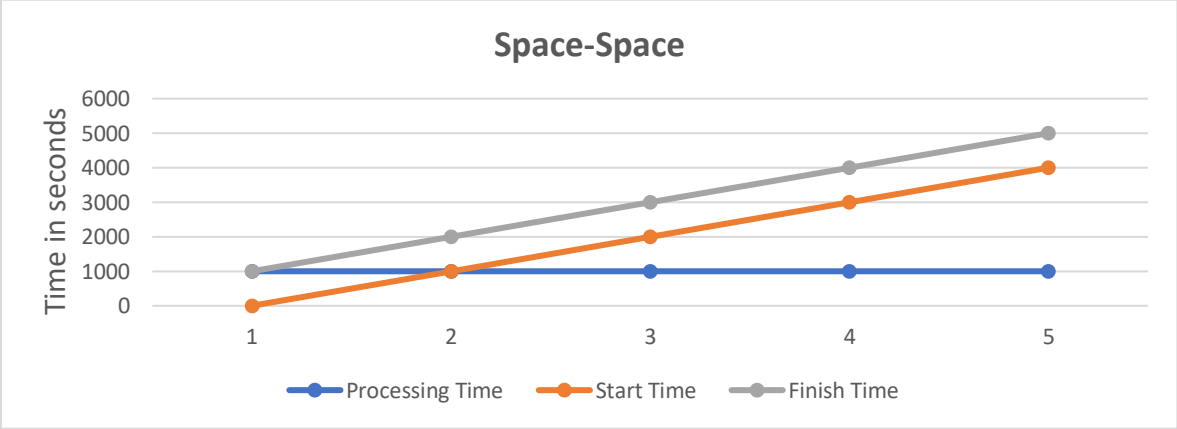
Comparison results of CloudSim simulation (Goyal, Singh and Agrawa, 2012)
In all the simulation 5 cloudlets are assigned to one virtual machine (vmid-0)and 5 cloudlets to the other virtual machine(vmid-1)

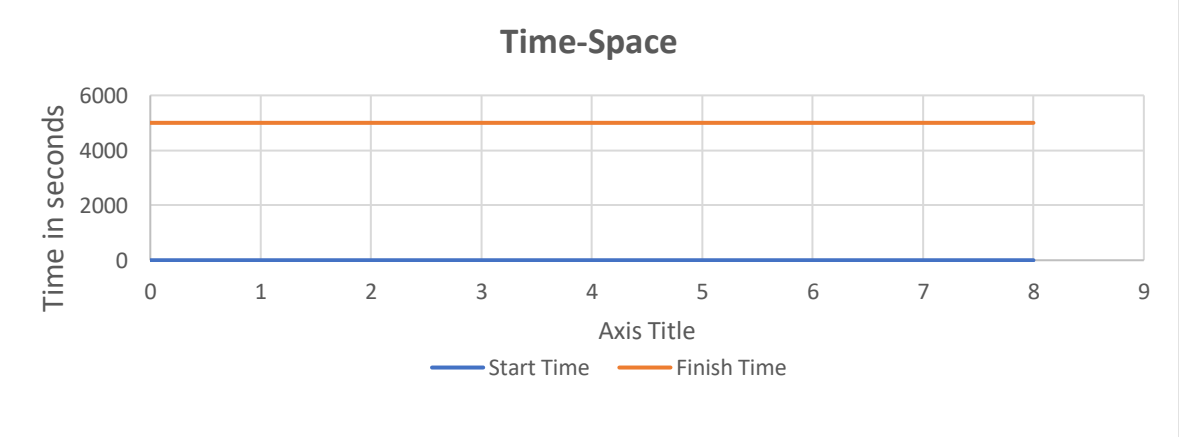
a) 2a Simulation

Scheduling policies	Cloudsim simulation of **Time-Time** shared policy	Cloudsim simulation of **Space-Time** shared policy	Cloudsim simulation of **Space-Space** shared policy	Cloudsim simulation of **Time-Space** shared policy
Total Processing Time	5000 s	1000 s	1000 s	5000 s
Total number of cloudlets	10	10	5	5
Time taken per cloudlet	5000	1000	1000	5000
Processing Elements	1	1	1	1
Virtual Machines	2	2	1	1
Data Centers	1	1	1	1
Host	1	1	1	1
	Remarks: All 10 cloudlets are simulated simultaneously, it takes more time for execution due to a single processor	Remarks: All 10 cloudlets run simultaneously the execution time is faster as space is first allocated	Remarks: Only 5 cloudlets have been executed as there is only one processing element running on both space shared policies	Remarks: 5 cloudlets are executed which are assigned to only one VM id where the



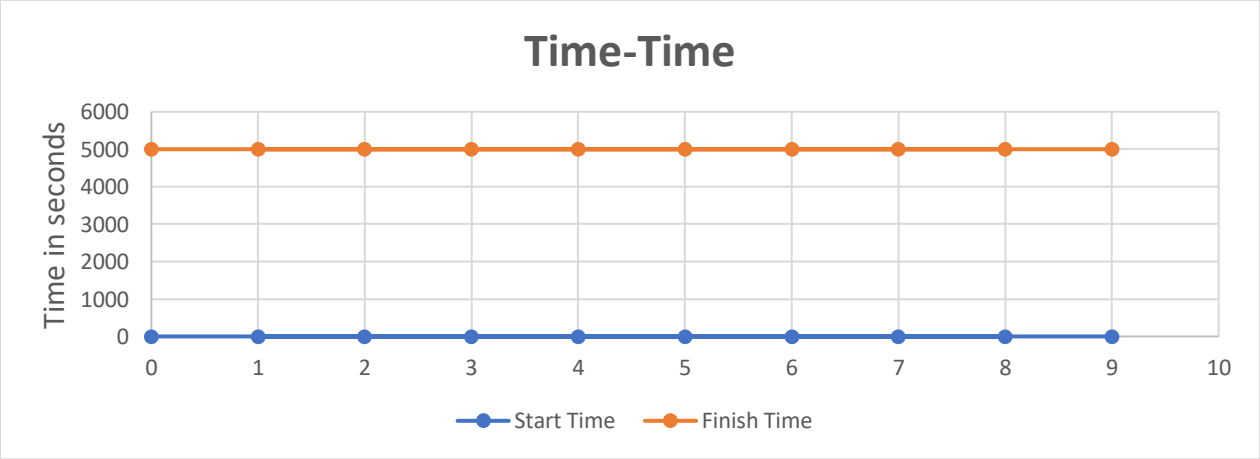

Space-Time
Time in seconds
6000
5000
4000
3000
2000
1000
0
1
2
3
4
5
6
7
8
9
10
Processing Time
Start Time
Finish Time

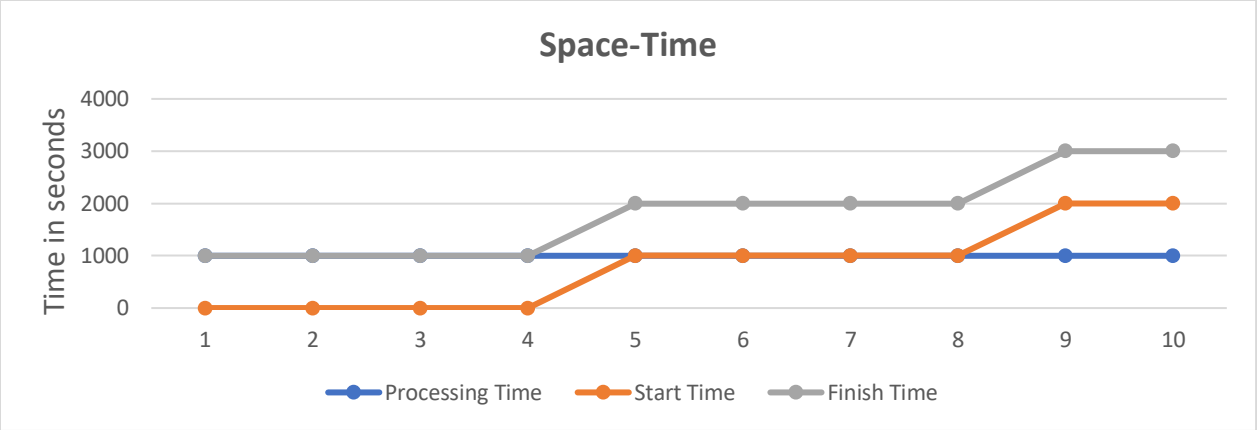

Space-Space
Time in seconds
6000
5000
4000
3000
2000
1000
0
1
2
3
4
5
Processing Time
Start Time
Finish Time

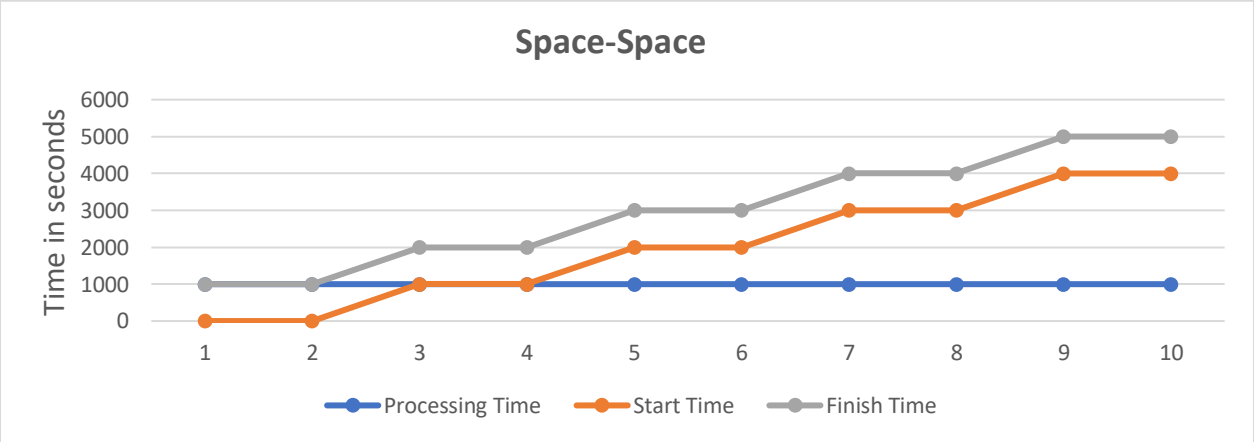

Time-Space
Time in seconds
6000
4000
2000
0
0
1
2
3
4
5
6
7
8
9
Axis Title
Start Time
Finish Time

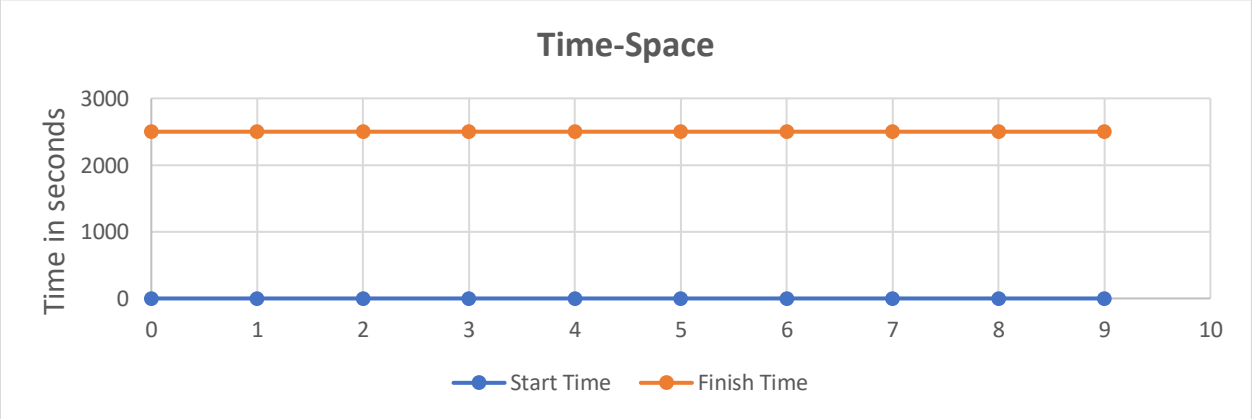
b) 2bSimulation

Scheduling policies	Cloudsim simulation of **Time-Time shared policy**	Cloudsim simulation of **Space-Time shared policy**	Cloudsim simulation of **Space-Space shared policy**	Cloudsim simulation of **Time-Space shared policy**
Total Processing Time	5000 s	1000 s	1000 s	2500 s
Total number of cloudlets	10	10	10	10
Time taken per cloudlet	5000	1000	1000	2500
Processing Elements	2	2	2	2
Virtual Machines	2	2	2	2
Data Centers	1	1	1	1
Host	1	1	1	1
Remarks:	It executes 10 cloudlets simultaneously since there are two processing elements each processor simulates 5 cloudlets at a time.	It simulates 4 cloudlets at a time and two cloudlets assigned to each of the two virtual machines hence the processing time is fast	It simulates 2 cloudlets at a time and one cloudlet assigned to each of the two virtual machine processing time is faster for individual cloudlets	It simulates 10 cloudlets at a time 5 assigned to each of the two virtual machines due to which the processing time is faster compared to all other policies in this simulation model



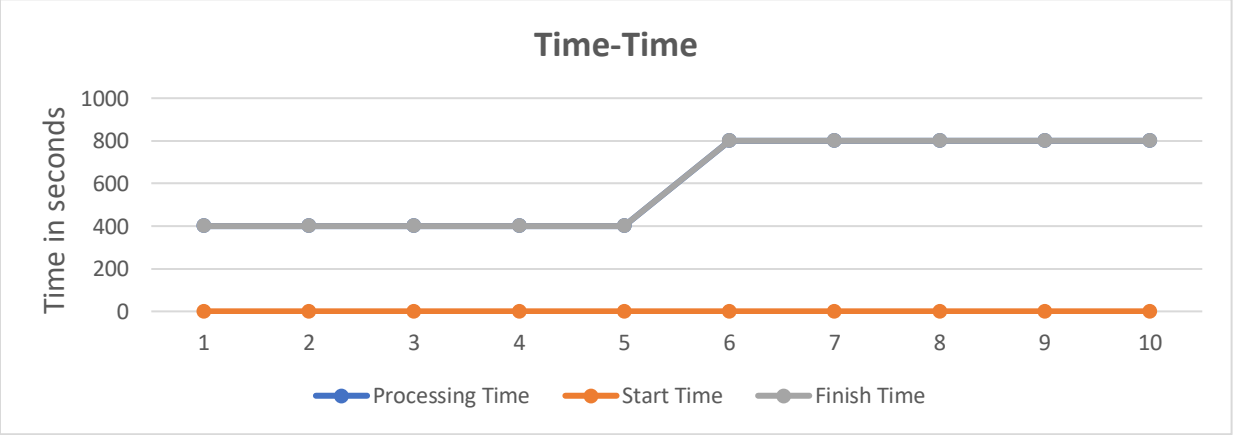

Space-Time
Time in seconds
4000
3000
2000
1000
0
1
2
3
4
5
6
7
8
9
10
Processing Time
Start Time
Finish Time

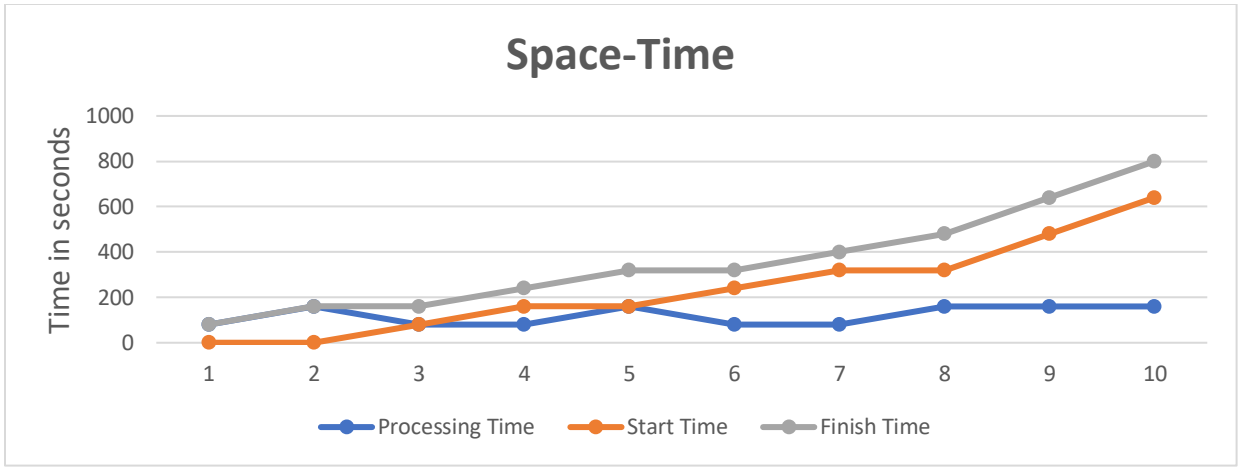

Space-Space
Time in seconds
6000
5000
4000
3000
2000
1000
0
1
2
3
4
5
6
7
8
9
10
Processing Time
Start Time
Finish Time

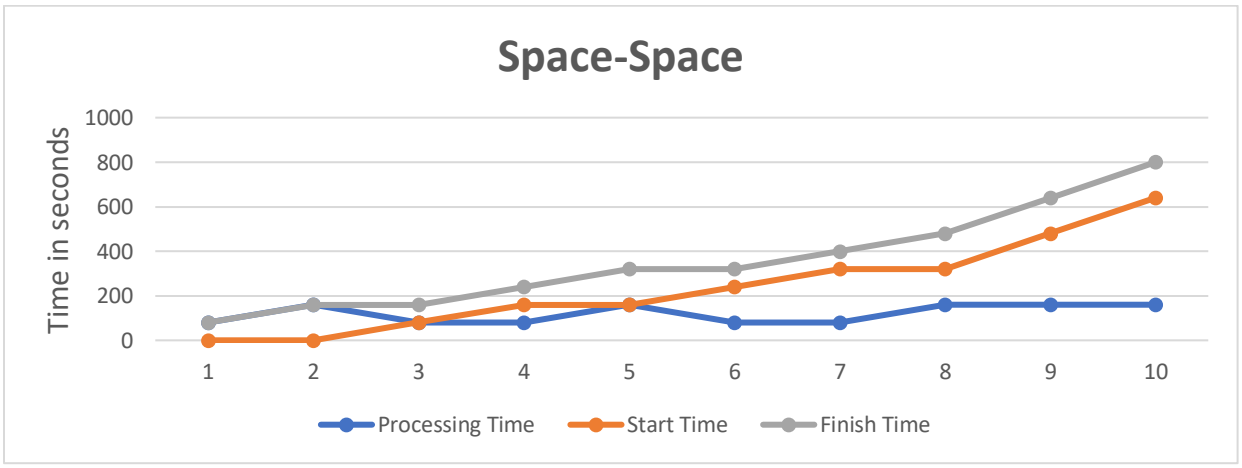

Time-Space
Time in seconds
3000
2000
1000
0
0
1
2
3
4
5
6
7
8
9
10
Start Time
Finish Time

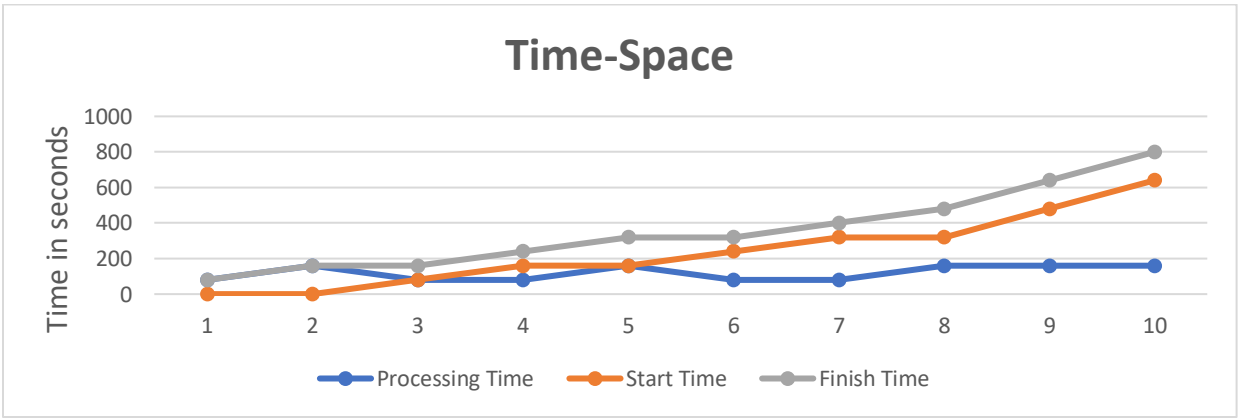
c) 2c Simulation

Scheduling policies	Cloudsim simulation of **Time-Time shared policy**	Cloudsim simulation of **Space-Time shared policy**	Cloudsim simulation of **Space-Space shared policy**	Cloudsim simulation of **Time-Space shared policy**
Total Processing Time	800 s	160 s	160 s	160 s
Total number of cloudlets	10	10	10	10
Time taken per cloudlet	400	80	80	80s
Processing Elements	1	1	1	1
Virtual Machines	1	1	1	1
Data Centers	1	1	1	1
Host	2	2	2	2
	It simulates 10 cloudlets at a time where it executes first 5 cloudlets the second set of cloudlets need to wait before they could be executed. There is a delay time for the second set of cloudlets due to the single processing element.	It simulates 2 cloudlets at a time where it executes only 1 cloudlet the second cloudlet needs to wait before it could be executed. There is a delay time for the second cloudlet due to the single processing element.	It is the same as the Space-Time shared policy. But it simulates the third cloudlet as soon as the second cloudlet is being executed because it executes the cloudlet allocated to the first virtual machine then the other.	It is same as the Space-Space shared policy




Space-Time
Time in seconds
1000
800
600
400
200
0
1
2
3
4
5
6
7
8
9
10
Processing Time
Start Time
Finish Time


Space-Space
Time in seconds
1000
800
600
400
200
0
1
2
3
4
5
6
7
8
9
10
Processing Time
Start Time
Finish Time


Time-Space
Time in seconds
1000
800
600
400
200
0
1
2
3
4
5
6
7
8
9
10
Processing Time
Start Time
Finish Time

9. Task 2: Business Organization

Fashion

E-commerce Management(Palar, Manongga, and Utomo, 2012)
The revolution of fashion has given a new transformation in the advancement of e-commerce online retail that is prone for specifically only for clothing and type of fabrics this model is best fit for the upcoming business transformation idea makes shopping go easy.

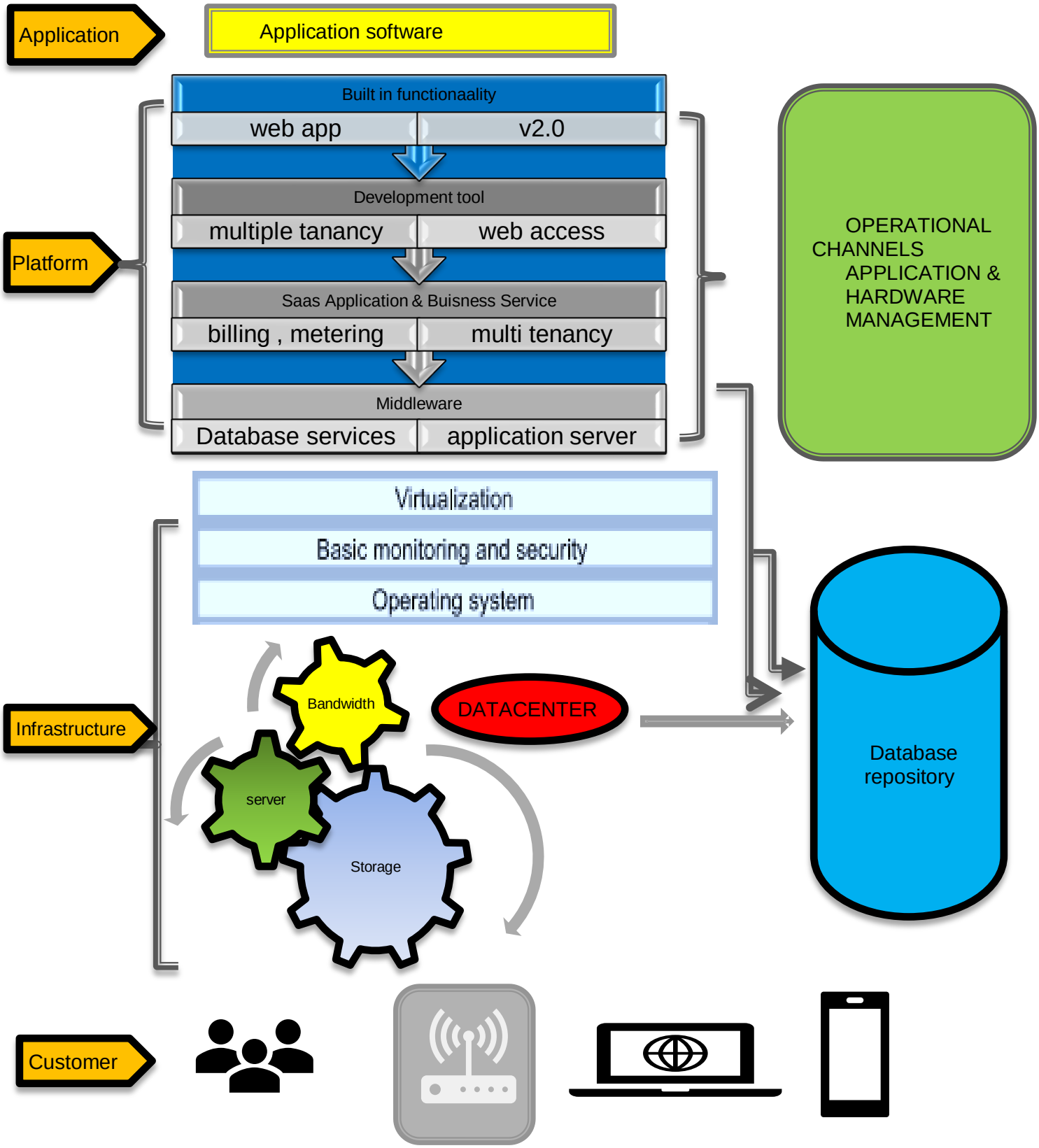
This business model is specific for the type of fabric clothes such as cotton, nylon, lyre and many admins has access to all a minimum of
100terrabyte storage cloud is allocated due to the globally distributed sellers and must compensate with all the system specification

The features of it are as follows

(a**) Centralized infrastructure**(Humane and Varshapriya, 2015) with lower costs in a certain location.

(b) Improves **efficiency** of the unutilized system for maximum utilization(Humane and Varshapriya, 2015)

The resources are available to end user uses a **hybrid cloud** they make unique storage that benefits from resources storage of confidential and non-confidential information between the private and public cloud it eases more secure control over the cloud and allows various parties to access information over the Internet. The public cloud contains the type of the fabric, clothes of various type size and stocks available and its prices the private cloud contains information about the users logged the fabric that needs to be restocked and seller information about which seller is selling the particular fabric or cloth type. The payment connection with the seller's payment account information with a commission taken by the cloud. The platform SaaS is used as a third-party application available for end-user access available in laptop and mobile where the user can get info of fabrics and cloth from world wide suppliers that sell under one platform. The Cloud storage repository contains all the information about users, supplier, list of clothes, and payment information's checks for demand about which cloth type has higher demand and sales then it request seller for restocking the service provide is an end to end user friendly platform where user can provide his info and address, choose the type of clothes he likes and selects the size there is further a tool that could help the buyer to be more confident on the cloth by making the user answer few questions and suggesting the style based on their answers. The advantage of this model is the use of **Saas** doesn't require much of cost and user can use the web application very easily the service provider doesn't require much of high-end development tools that to complex the centralized infrastructure


Application
Application software
Built in functionaality
web app
v2.0
Development tool
multiple tanancy
web access
Platform
Saas Application & Buisness Service
billing , metering
multi tenancy
Middleware
Database services
application server
OPERATIONAL
CHANNELS
APPLICATION &
HARDWARE
MANAGEMENT
Virtualization
Basic monitoring and security
Operating system
Bandwidth
DATACENTER
Infrastructure
server
Storage
Database
repository
Customer

11. Scalability in cloud environments area unit as follows:

Many third-party web applications are hard to scale in large size. Particularly communication infrastructure technologies(Goyal, Singh and Agrawa, 2012). They work stable until a certain scale reached and then fails. So they are forced to build their own.
Proactive scaling is to predict the scalability level, where the traffic requests are expected or this we should first understand the expected traffic flow in order to perform proactive scaling(Falatah and Batarfi, 2014). but scheduling things that creates a way to problems when expectations turn wrong.
There might be an occurrence of a red zone that is the time when there is an increase in web traffic and high bandwidth where the cloud must be able to justify the encrypted at a high and non-encrypted data speed. There is a limitation in bandwidth for cloud access.
Data Storage: Business-based User cloud service are concerned with in capabilities of controlling the place where the data is to be stored, replaced, portioned and distributed. This is due to the inability of control that makes client wary of procedure policies in use.
Data redundancy and replication is the major concern where many sellers info is stored in multiples in a single data center though it prevents from loss of data it increases the difficulty of identification of the uniqueness of particulars(Alshammari, Singer and Storer, 2017).
Software if there is a concern about manipulating the information in the cloud then all the services must be downloaded on all the devices.
Design issues
Isolating a large portion of the cloud while designing affects restrictions in the handling of application states is automatically dealt with the scaled applications.
Loosely coupled since the cloud environment is globally distributed there are a large number of IT resources on which the application relay on that frequently changes that increases the application dependency components which impact on the failure of a component in the application.
Elasticity, resources are continuously added and deleted during execution time. The increasing number of workload leads to scale out of cloud application where the individual resources are incapable of handling a large number of data.

12. Business continuity:

The main aim of the organization is to extend the business growth
Downtime Elimination, SaaS make sure emails are not lost Generally concerning invisible to complete users in spite of anything that happens to employees or the centralized infrastructure. Better Network and Information Security Management. The company supplies non-critical applications and information to the cloud with which it can run with higher performance. Which permits the corporate IT department for specializing in crucial applications. This also improves in the network security and user access management.
The successful backup from a disaster relays on the frequency and quality of backups. Cloud provides much higher stratified recovery strategy(Palar, Manongga, and Utomo, 2012). This feature provides a stronger backup Point Objective. Disaster Recovery – Geographic Redundancy. Cloud offers an inbuilt geographically redundancy is in the style of regions and accessibility zones. This characteristic offers to decrease the Recovery Time Objective . We remember that many of the events in Table one are geographically connected. Avoid or eliminate disruption of operations. Some clouds expose a hash once store object, then eliminating the need for Rhetorical image verification time. Increased accessibility. The scalability characteristic of cloud computing facilities permits for larger accessibility.
Redundancy is present all over the cloud environments and as per demand resource capacity

can increase service accessibility. DoS Attack Depreciation. Redundancy as per requirement scalability of resources is done collectively offer higher resistance if distributed denies service attacks, and for faster backup from serious traumas.
Availability of cloud no cloud is free of failure. This is a major challenge for services and applications where components compete for resources and rely on other internal or external components/services that fail or rely on faulty software. Planning for the detection, logging, fixing and recovery of these failures involves not only developers but all teams.
each provides customers with its own interface to interact with underlying infrastructures. The information security requirements of each Business Cloud model with ISO standards are also investigated. This gives confidence in the selection of an appropriate and highly protected cloud framework
Automated management is because the elasticity in cloud applications .wherever the data center hosts many VM's in a very single physical machine that has excess resources to fulfill the demand.
Distribution: Cloud environments are globally distributed environments made up of several IT resources. Due to which , Cloud applications need to be split into separate application components that will be distributed with all the environmental resources.

13. References

Alshammari, D., Singer, J. and Storer, T. (2017) 'Does CloudSim Accurately Model Micro Datacenters?', in *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*. IEEE, pp. 705–709. doi: 10.1109/CLOUD.2017.97.

Calheiros, R. N. *et al.* (2010) 'CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms'. doi: 10.1002/spe.995.

Falatah, M. M. and Batarfi, O. A. (2014) 'CLOUD SCALABILITY CONSIDERATIONS', *International Journal of Computer Science & Engineering Survey (IJCSES)*, 5(4). doi: 10.5121/ijcses.2014.5403.

Goyal, T., Singh, A. and Agrawa, A. (2012) 'Cloudsim: Simulator for cloud computing infrastructure and modeling', *Procedia Engineering*, 38(December 2012), pp. 3566–3572. doi: 10.1016/j.proeng.2012.06.412.

Haider, W. and Wahab, A. (2011) 'A review on Cloud Computing Architectures and Applications', *Computer Engineering and Intelligent Systems*, 2(4), pp. 206–210. Available at: http://iiste.org/Journals/index.php/CEIS/article/view/383.

Humane, P. and Varshapriya, J. N. (2015) 'Simulation of cloud infrastructure using CloudSim simulator: A practical approach for researchers', in *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*. IEEE, pp. 207–211. doi: 10.1109/ICSTM.2015.7225415.

Kumar, S. and Goudar, R. H. (2012) 'Cloud Computing – Research Issues, Challenges, Architecture, Platforms and Applications: A Survey', *International Journal of Future Computer and Communication*. doi: 10.7763/IJFCC.2012.V1.95.

Palar, R. T., Manongga, D. and Utomo, W. H. (2012) *An Appropriate Cloud Computing Business Model and Its Services for Developing Countries: A Comparison of Cloud Computing Business Model in Indonesia*, *International Journal of Computer Applications*. Available at: https://pdfs.semanticscholar.org/04ca/f9573be76706e06835994e29f9ee1ac47c50.pdf (Accessed: 8 December 2018).