An Improved Approach for Cloud Computing By Optimal Utilization of Resourcewith High Performance Computation

1Shachi P Shah ,2 Prof. Ishan Rajani
[1]Student, [2]Assistant Professor
[1]Department of Computer Engineering, [2]Information Technology
[1,2]Silver Oak College Of Engineering and Technology , Ahmedabad,Gujarat, India

Abstract:High performance computing improves usability and scalability of data processing over cloud. When used with different type of data having large size values with higher processing requirements, its compulsory for us to provide better range machines. Conventionally, HPC outperforms much low capacity system when used with cloud processing over data centers. But it requires a physical setup that involves purchasing, transport, wiring and configuration that could take weeks to implement. Our proposed algorithm setup can be done in matter of minutes and is better in processing for energy efficiency. My algorithm uses VM migration to achieve load balance for tightly coupled parallel applications executing in virtualized environments that suffer from interfering jobs. While restoring load balance, it not only reduces the timing penalty caused by interfering jobs, but also reduces energy consumption significantly. My result will show significant improvement in energy consumption level and thus comparison will be based on number of migrations and energy consumption in watt.

***Keywords*:High performance computing, cloud computing, migration .**

I .INTRODUCTION

High-performance computing (HPC) is the use of parallel processing for running advanced application programs efficiently, reliably and quickly.HPC functions above 10^{12}floating point operation per second. The most common users of HPC systems are scientific researchers, engineers and academic institutions. Some government agencies, particularly the military, also rely on HPC for complex applications. High-performance computing (HPC) evolved due to meet increasing demands for processing speed. HPC brings together several technologies such as computer architecture, algorithms, programs and electronics, and system software under a single canopy to solve advanced problems effectively and quickly. A highly efficient HPC system requires a high-bandwidth, low-latency network to connect multiple nodes and clusters.

II. BACKGROUND THEROIES

2.1High Performance computing

High-performance computing (HPC) is the use of parallel processing for running advanced application programs efficiently, reliably and quickly.HPC functions above 10^{12}floating point operation per second. The most common users of HPC systems are scientific researchers, engineers and academic institutions. Some government agencies, particularly the military, also rely on HPC for complex applications. High-performance computing (HPC) evolved due to meet increasing demands for processing speed. HPC brings together several technologies such as computer architecture, algorithms, programs and electronics, and system software under a single canopy to solve advanced problems effectively and quickly. A highly efficient HPC system requires a high-bandwidth, low-latency network to connect multiple nodes and clusters.

2.2 HPC Architecture

A high performance computer appropriate for most small and medium-sized businesses today is built from what are basically many ordinary computers connected together with a network and centrally coordinated by some special software. Because the computers are usually physically very close together, the common term for a high performance computer that you'd used in your business today is cluster. Because clusters are just collections of computers that are connected together with a computer network and special software to help them all work together, you're probably already familiar with the components used to build a cluster. All the basics are still there: processors, disk, memory, and so on. The primary difference is one of scale: there's more of everything in a cluster.

2.3 Cloud Computing

cloud computing is the delivery of computing services—servers, storage, databases, networking, software, analytics, intelligence and more—over the internetThe cloud offer faster innovation, flexible resources You typically pay only for cloud services you use, helping lower your operating costs, run your infrastructure more efficiently and scale as your business needs change..
Types of Cloud: IaaS, PaaS, SaaS
Ex. of cloud ->Google , Microsoft Azure , Amazon AWS, IBM Cloud

2.4 Benefits of HPC and Cloud Computing

HPC has a unique set of requirements that might not fit into standard clouds. [3] [1] However, plenty of commercial options, including cloud-like services, provide the advantages of real HPC without the capital expense of buying hardware. [2]The advantage of pay-as-you-go computing has been an industry goal for many years. [1] [7]In HPC users can construct their own operating system instance and run it in a cloud whenever they need computational resources. [5]In addition to cloud computing, cloud storage also can be used independently or combined with OS instances. [6]Cloud computing would seem to be an HPC user's dream offering almost unlimited storage and instantly available and scalable computing resources, all at a reasonable metered cost.

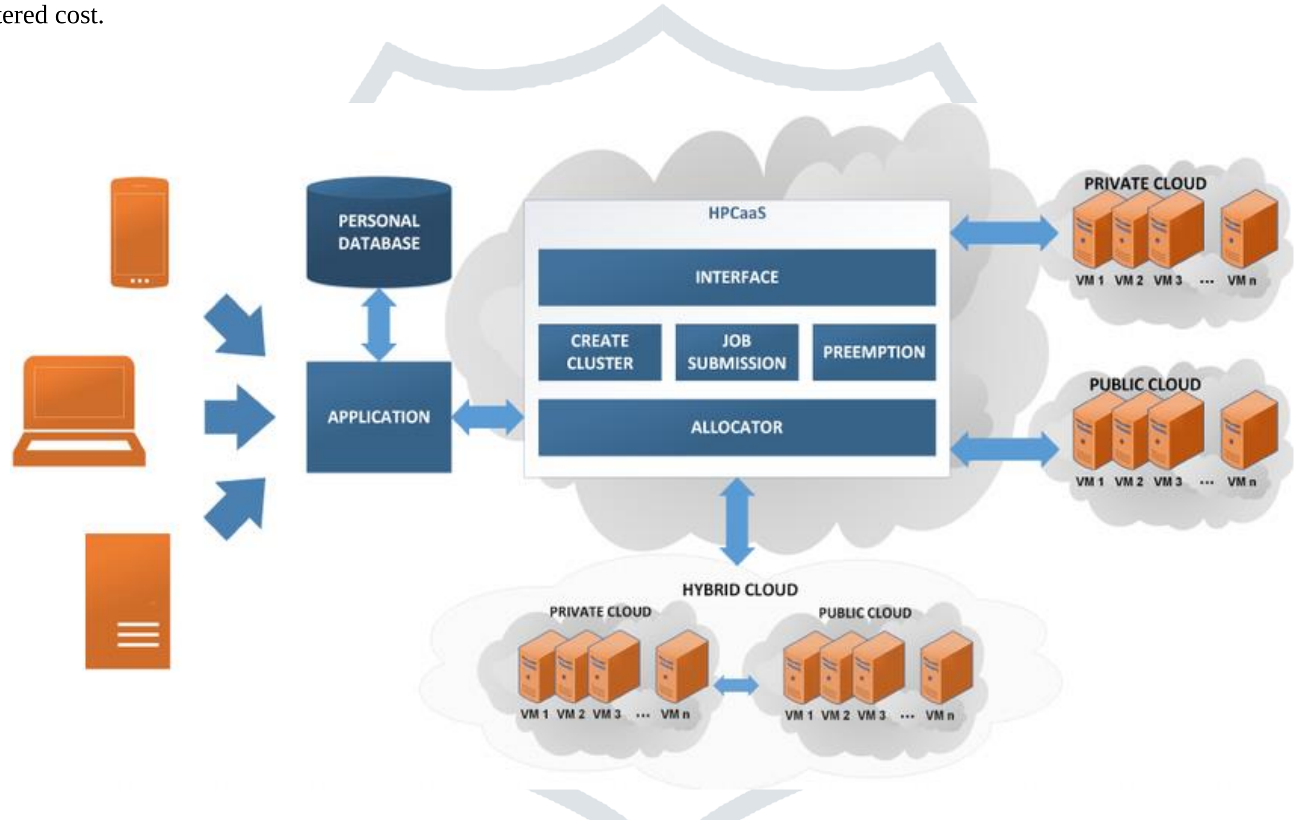


Fig (1).General Architecture

2.5 Scheduling

Round Robin is the pre-emptive process scheduling algorithm. Each process is provided a fix time to execute, it is called a quantum. Once a process is executed for a given time period, it is pre-empted and other process executes for a given time period. Context switching is used to save states of pre-empted processes.Consider the following processes with arrival time and burst time.

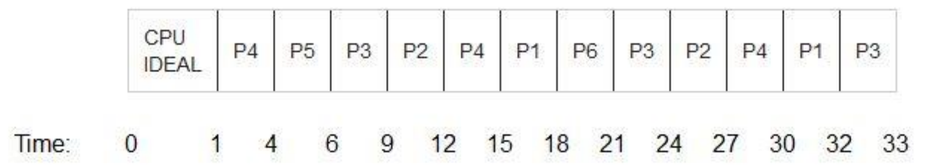
Process id	Arrival time	Burst time
P1	5	5
P2	4	6
P3	3	7

Time

P4	1	9
P5	2	2
P6	6	3

quantum 3

Queue:



Process id	Arrival time	Burst time	Completion time	Turnaround time	Waiting time	Response time
P1	5	5	32	27	22	10
P2	4	6	27	23	17	5
P3	3	7	33	30	23	3
P4	1	9	30	29	20	0
P5	2	2	6	4	2	2
P6	6	3	21	15	12	12

Average turnaround time=(27+23+30+29+4+15)6=21.33

Average waiting time=(22+17+23+20+2+12)6=16

Average response time=(10+5+3+0+2+12)6=5.33

III. LITRATURE SURVEY

In this paper [1]Traditional High Performance Computing (HPC) clusters built to handle big data processing have inherent weaknesses that can be overcome by migrating to a more flexible cloud computing environment In this paper propose solution called HPC+Cloud that enables enterprises to migrate to, and subsequently manage, high performance computing on the cloud High Performance Computing as a service is the pay –per-use –model compared to initial capital investment ,the operational cost ,and proper underutilization of traditional HPC systems, having to pay for only what is used and only when it is needed equates to large financial savings for users. HPC+Cloud architecture is the ability to scale the HPC cluster by provisioning new virtual processing nodes from the cloud on an on-demand basis. All the advantages of the hybrid cloud are inherited by the proposed HPC+Cloud paradigm but with lesser administrative overhead compared to a traditional hybrid cloud since a user does not need to create and manage a private cloud on premise.

In this paper [2]Cloud computing is emerging as a promising alternative to supercomputers for some high performance computing applications.In this paper, a novel heuristics for online application –aware job scheduling in multi-platform environments is presented.Clouds can act as a cost-effective and timely solution to the needs of some academic and commercial HPC users.CLOUDQAL is aquality model is to propose for cloud servicesLightweight virtualization reduces the overhead of network virtualization by grating VMs native accesses to physical network interfaces.The advantages of proposed System are:

- Better turnaround time for job
- Assign job to least loaded platform
- Good performance in cloud

In this paper [3]They propose a pre-copy live VM migration using Distributed Shared Memory(DSM) computing model The setup is built using two identical computation node to construct the environment services architecture namely the virtualization infrastructure ,the shared storage server, and the DSM and High performance Computing Cluster. In this model downtime is reduced by 50% in the idle workload of Windows VM and 66% in case of Ubuntu Linux idle workload In general this model not only reduces the downtime and the total amount of data sent, but also does not degrade other metrics like the total migration time and application performance In this System architecture runs four services which work in a cooperative way to handle the live VM migration of XenServer hypervisor in an optimized way In this paper logical modules as layer architecture to build the VM migration The first part is the shared storage NFS Protocol, which is a transparent protocol that allows the shared storage server update to be synchronized with all virtual members. Second part is Virtualization Infrastructure using Citrix XenServer version 6.2 hypervisor which is used to create the virtual machines, and managed by Citrix Xen-Center management console Third part is the HPC Cluster Distributed Memory with message Passing Fourth part is the Distributed Shared Memory framework

In this paper [4]Quanta Cloud Technology **(QCT**)customized HPC cluster software stack including system provisioning ,core HPC services ,development tools ,and optimized applications and libraries which are distributed as pre-built and validated binaries and are meant to seamlessly layer on the top popular Linux distributions with the integration conventions define by OpenHPC project The architecture of QCT HPC Cluster Software Stack is Intentionally modular to allow end users to pick and choose from provided components, as well as to foster a community of open contribution. The system architecture used for any HPC Cluster has several network:

1. In-band internal management Ethernet network
2. Out-to-Bans power control and console Ethernet network
3. High-performance low –latency network or fabric
4. A Shared or optional dedicated network for parallel file system service

In this paper presented an overview of QCT HPC Cluster tool kits with QCT HPC Cluster Software stack ,a customized and layered based collections of software components used for rapidly build a HPC Cluster System and run tests on QCT Developer CloudIn the future ,we will actively participate in OpenHPC community and work with professionals to focused on optimizing software components and user applications for OpenHPC Future efforts focus on providing automation for more advanced configuration and tuning to address scalability , power management, and high availability concerns

In this paper [5]Adaptive CPU Sharing approach that reduces co-tenants interference and provides predictable application performanceOur approach is to monitor the progress of submitted application at runtime, tracks the slowdown of individual application at runtime, track the slowdown of individual application and applies adjustment until convergence we predicted application performance degradation by creating a mathematical relationship between high-level application performance and low –level machine eventsOur approach helps mitigate co-tenant interference and reduces unfairness by minimizing the overall application slowdownsIn this paper Adaptive CPU Sharing approach that reduces co-tenants interference and provides predicatable application performanceOur approach is to monitor the progress of submitted application at runtime, tracks the slowdown of individual application at runtime, track the slowdown of individual application and applies adjustment until convergence we predicted application performance degradation by creating a mathematical relationship between high-level application performance and low –level machine eventsOur approach helps mitigate co-tenant interference and reduces unfairness by minimizing the overall application slowdowns

In this paper [6]they suggest two approaches to make HPC resources available in a dynamically reconfigurable hybrid HPC/Cloud architectureThe first approach, from the University of Melbourne, generates a consistent compute node operating system image with variation in the virtual hardware specification. The second approach, from the University of Freiburg, deploys a cloud-client on the HPC compute nodes, so the HPC hardware can run Cloud-Workloads for backfilling free compute slots On top of the HPC configuration it enables users to run virtual machines as standard compute jobs (VM jobs). To run virtual machines on every compute node on the cluster the KVM hypervisor is on each of these nodes. This architecture enables users to run compute jobs on bare metal through the resource manager (bare metal jobs) or inside a virtual machine (VM job) without partitioning the cluster into two parts.In this paper they used University of Freiburg three workloads:

1 job Submission via Moab scheduler without running a resource manager client in the VM.,

2 job Submission via Moab scheduler with a resource manager client (Torque) running in the VM

3 job Submission via OpenStack Dashboard/API.

IV. PROPOSED FLOW

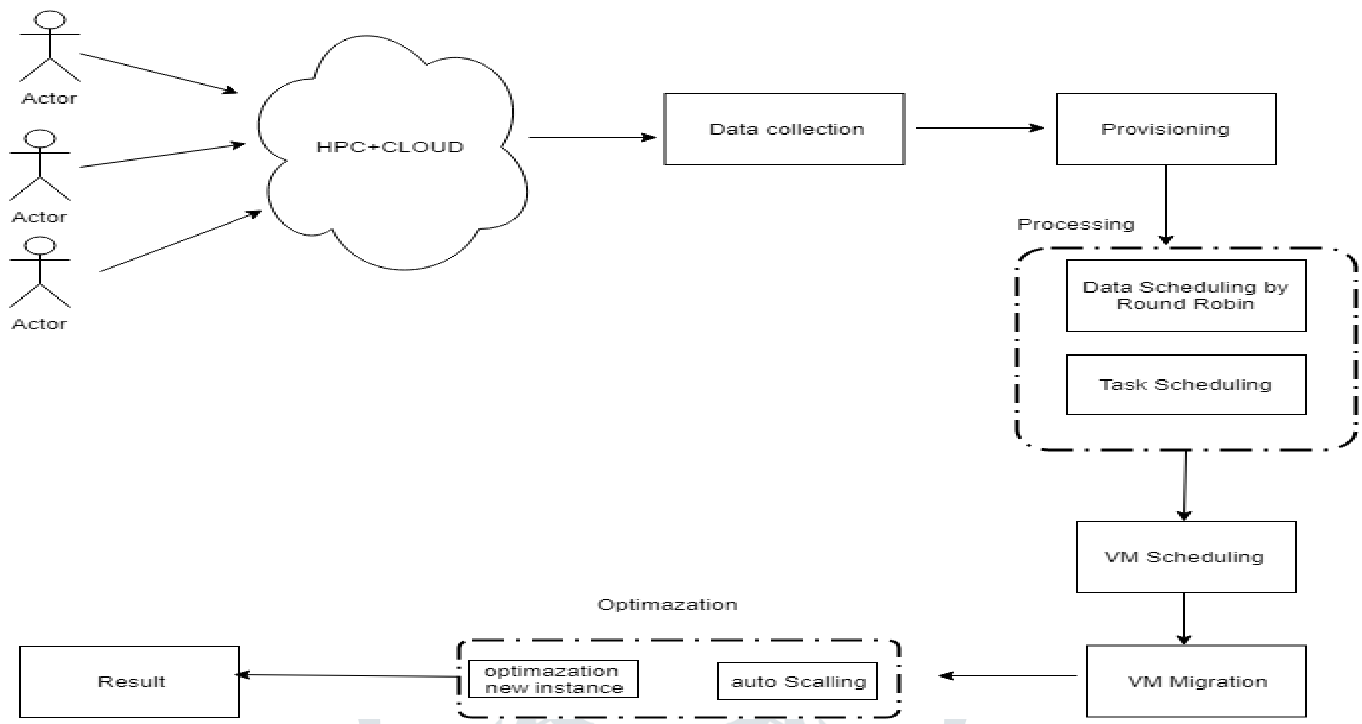


Fig (2).Proposed Flow

In this research mainly focus reduce overhead.

1. Client request comes to HPC+CLOUD architecture
2. In data collection , data is collected into single source
3. Data collection process that data get provisioning as scheduling process.
4. After scheduling it will select the task.
5. That task will be perform with required migration process which is energy aware VM scheduling
6. After the scheduling move to optimization process
7. In optimization it will perform auto scaling for changed VM position and host configuration
8. In auto scaling add and remove compute resource depending upon actual usage
9. It will set new instance as changed by the VM scheduling algorithm
10. After that the result will be displayed in the parameters energy consumption and number of VM migration

V. Simulation

5.1 Cloudsim:

Cloudsim is a framework for modeling and simulation of cloud computing infrastructures and services. Originally built primarily at the Cloud Computing and Distributed Systems (CLOUDS) Laboratory.

The main components of the CloudSim framework:

1. **Regions :** It models geographical regions in which cloud service providers allocate resources to their customers
2. **Data centres:** It models the infrastructure services provided by various cloud service providers.
3. **Hosts:** It models physical resources.
4. **The user base:** It models a group of users considered as a single unit in the simulation, and its main responsibility is to generate traffic for the simulation.
5. **Cloudlet:** It specifies the set of user requests. It contains the application ID, name of the user base that is the originator to which the responses have to be routed back, as well as the size of the request execution commands, and input and output files.
6. **Service broker:** The service broker decides which data centre should be selected to provide the services to the requests from the user base.
7. **VMM allocation policy:** It models provisioning policies on how to allocate VMs to hosts.
8. **VM scheduler:** It models the time or space shared, scheduling a policy to allocate processor cores to VMs.

5.2 Cloudsim Plus:

CloudSim Plus is a full-featured, highly extensible simulation framework enabling modeling, simulation, and experimentation of Cloud computing infrastructures and application services. It allows users to focus on specific system design issues to be investigated, without concerning the low-level details related to Cloud-based infrastructures and services.

CloudSim Plus is a fork of CloudSim 3, re-engineered primarily to avoid code duplication. It provides code reusability and ensures compliance with software engineering principles and recommendations for extensibility improvements. It focuses on usage of software engineering standards and recommendations such as Design Patterns, SOLID principles and other ones such as KISS and DRY.

5.3 Output:

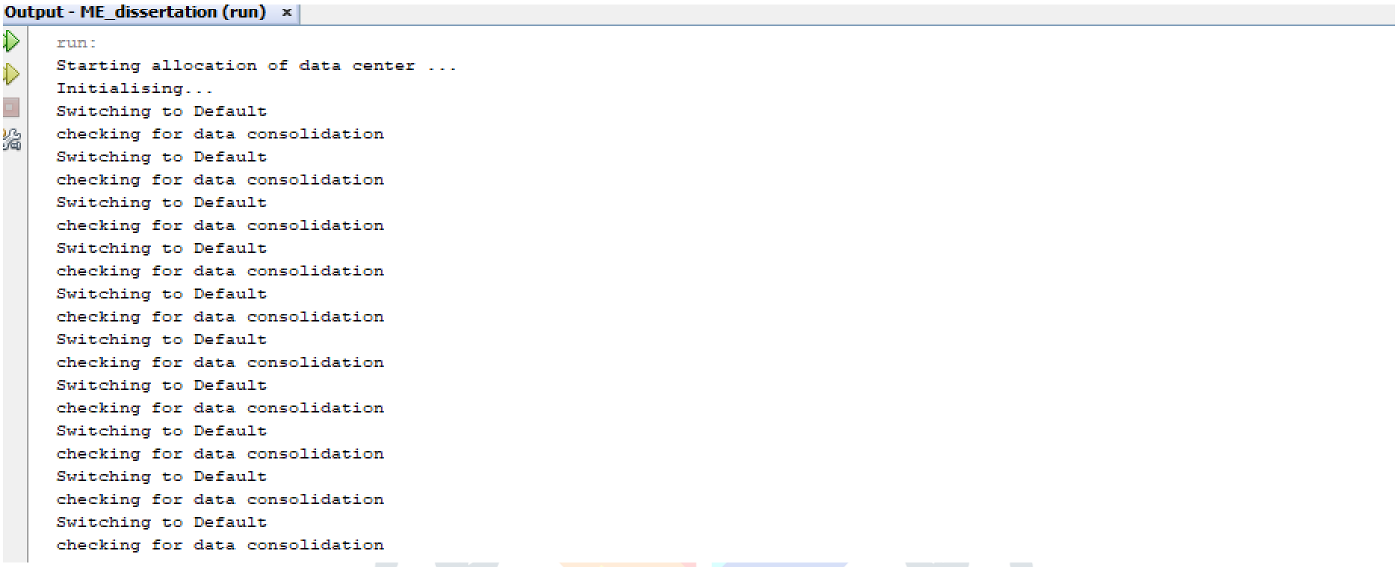


Fig (3).Data collection

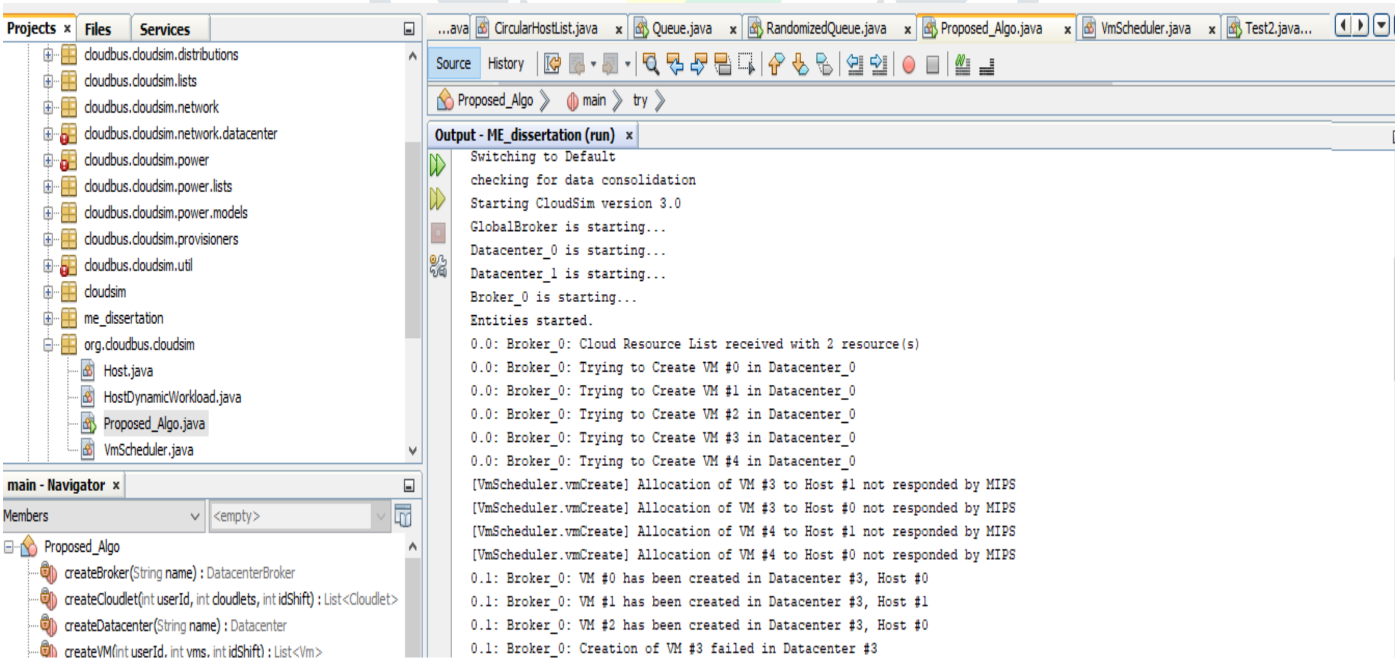
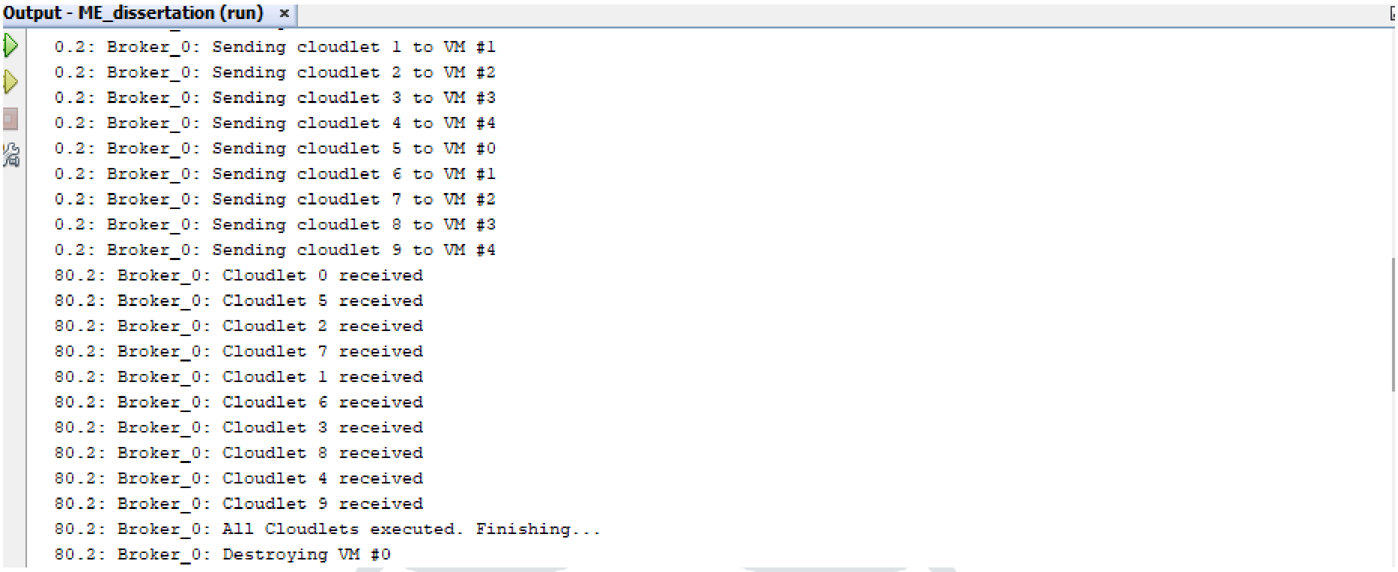
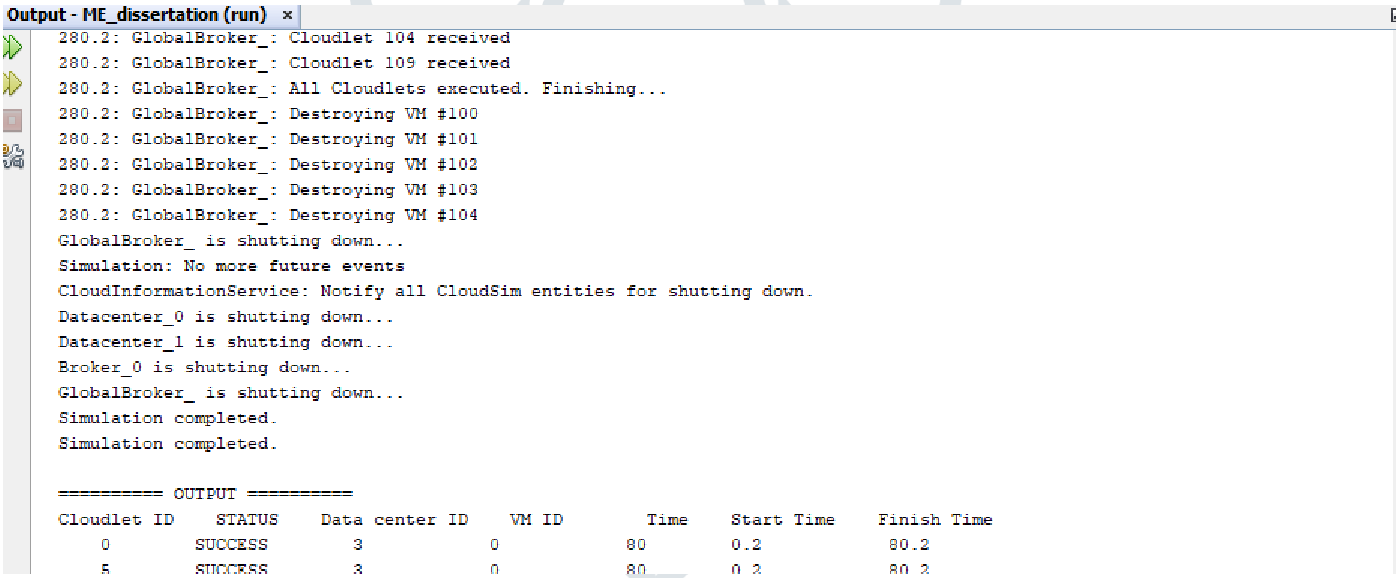


Fig (4).Create VM



```
Output - ME_dissertation (run)
0.2: Broker_0: Sending cloudlet 1 to VM #1
0.2: Broker_0: Sending cloudlet 2 to VM #2
0.2: Broker_0: Sending cloudlet 3 to VM #3
0.2: Broker_0: Sending cloudlet 4 to VM #4
0.2: Broker_0: Sending cloudlet 5 to VM #0
0.2: Broker_0: Sending cloudlet 6 to VM #1
0.2: Broker_0: Sending cloudlet 7 to VM #2
0.2: Broker_0: Sending cloudlet 8 to VM #3
0.2: Broker_0: Sending cloudlet 9 to VM #4
80.2: Broker_0: Cloudlet 0 received
80.2: Broker_0: Cloudlet 5 received
80.2: Broker_0: Cloudlet 2 received
80.2: Broker_0: Cloudlet 7 received
80.2: Broker_0: Cloudlet 1 received
80.2: Broker_0: Cloudlet 6 received
80.2: Broker_0: Cloudlet 3 received
80.2: Broker_0: Cloudlet 8 received
80.2: Broker_0: Cloudlet 4 received
80.2: Broker_0: Cloudlet 9 received
80.2: Broker_0: All Cloudlets executed. Finishing...
80.2: Broker_0: Destroying VM #0
```

Fig (5).Sending VM to Cloudlet



```
Output - ME_dissertation (run)
280.2: GlobalBroker_: Cloudlet 104 received
280.2: GlobalBroker_: Cloudlet 109 received
280.2: GlobalBroker_: All Cloudlets executed. Finishing...
280.2: GlobalBroker_: Destroying VM #100
280.2: GlobalBroker_: Destroying VM #101
280.2: GlobalBroker_: Destroying VM #102
280.2: GlobalBroker_: Destroying VM #103
280.2: GlobalBroker_: Destroying VM #104
GlobalBroker_ is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker_0 is shutting down...
GlobalBroker_ is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    0          SUCCESS        3              0       80        0.2          80.2
    5          SUCCESS        3              0       80        0.2          80.2
```

Fig (6).Destroying VM

Output - ME_dissertation (run)

========== OUTPUT ==========

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	3	0	80	0.2	80.2
5	SUCCESS	3	0	80	0.2	80.2
2	SUCCESS	3	2	80	0.2	80.2
7	SUCCESS	3	2	80	0.2	80.2
1	SUCCESS	3	1	80	0.2	80.2
6	SUCCESS	3	1	80	0.2	80.2
3	SUCCESS	4	3	80	0.2	80.2
8	SUCCESS	4	3	80	0.2	80.2
4	SUCCESS	4	4	80	0.2	80.2
9	SUCCESS	4	4	80	0.2	80.2
100	SUCCESS	3	100	80	200.2	280.2
105	SUCCESS	3	100	80	200.2	280.2
102	SUCCESS	3	102	80	200.2	280.2
107	SUCCESS	3	102	80	200.2	280.2
101	SUCCESS	3	101	80	200.2	280.2
106	SUCCESS	3	101	80	200.2	280.2
103	SUCCESS	4	103	80	200.2	280.2
108	SUCCESS	4	103	80	200.2	280.2
104	SUCCESS	4	104	80	200.2	280.2

Fig (7).Simulation start and finish time for VM

VI. CONCLUSION

HPC for energy efficient VM scheduling is a challenging task. The two models - HPC with Cloud VMs on Compute Nodes, and HPC with Compute Nodes as Cloud VMs - represent different hybrid systems to solve different problems. In effect, here the HPC compute nodes are replaced with cloud virtual machines. Thus to make existing compute nodes available to perform the next task for their particular configurations. In this case, virtual machines have made accessible to an HPC system to provide cost-efficiencies and improved throughput.

VII. REFRENCES

[1] High Performance Computing on the Cloud via HPC+Cloud software framework Suresh Reuben Balakrishnan\ ShanmugamVeeramanii, John Alan Leongi,2, lain Murra/, Amandeep S. Sidhu2,4 i Faculty of Engineering and Science, Curtin University, Malaysia 2 Curtin Sarawak Research Institute, Curtin University, Malaysia 3 Faculty of Science and Engineering, Curtin University, Australia 4 Faculty of Health Sciences, Curtin University, Australia

[2] OPTIMIZATION OF PERFORMANCE AND SCHEDULING OF HPC CLOUD USING CLOUDSIM AND SCHDULING APPROACHD. $BoobalaMuralitharan_1$,$S.ArockiaBabiReebha_2$,D. $Saravanan_3$ Assistant Professor, Department of MCA, Saranathan College of Engineering,Trichy.Head Of The Department Of Computer Science And Engineering,PavendarBharathidas College

[3] Distributed Shared Memory based Live VM MigrationTariq Daradkeh*Department of Electrical and Computer Engineering Concordia University Montreal, Canada t* _darad@encs.concorda.caAnjalAgarwal*Department of Electrical and Computer Engineering Concordia University Montreal, Canada aagarwal@encs.concorda.ca*

[4] Customized HPC Cluster Software Stack on QCT Developer Cloud Stephen Chang Research and Development Department Quanta Cloud Technology Taoyuan City, Taiwan Stephen.Chang@QCT.io Andy Pan Business Development Department Quanta Cloud Technology Taoyuan City, Taiwan Andy.Pan@QCT.io.

[5] Spartan and NEMO: Two HPC-Cloud Hybrid Implementations Lev Lafayette Department of Infrastructure University of Melbourne Melbourne, Australia lev.lafayette@unimelb.edu.auBernd Wiebelt High Performance Computing Albert-Ludwigs-Universität Freiburg, Germany bernd.wiebelt@rz.uni-freiburg.de

[6] A Simple Energy-Aware Virtual Machine Migration Algorithm in a Server ClusterRyo Watanabe1(B), Dilawaer Duolikun1, Qin Cuiqin1, Tomoya Enokido2, and Makoto Takizawa 1 Hosei University, Tokyo, Japan ryo.watanabe.4h@stu.hosei.ac.jp, dilewerdolkun@gmail.com,qcuiqin@gmail.com,makoto.takizawa@computer.orRissho University, Tokyo, Japan eno@ris.ac.jp

[7] Kruener, and G. Tay, "Predicts 2014: Content Gets Bigger, Richer and More Personal," Gartner2013.

[8] N. Heudecker, M. A. Beyer, D. Laney, M. Cantara, A. White, and R. Edjlali,"Predicts 2014: Big Data," Gartner2013. [

[9] G. Bell, J. Gray, and A. Szalay, "Petascale computational systems," Computer,vol. 39, pp. 110-112,2006 2013

[10] J. Monroe, A. Chandrasekaran, A. Dayley, V. Filks, S. Zaffos, J. Unsworth, and D. Russell, "Predicts 2014: More Storage Capacity and Efficiency,Less Cost - Managing Infinite Data From Every Direction," ed: Gartner, 2013