Scheduling approach for HPCaaS (HPC as a Service)

MaitreyPancholi
IT Systems and Network Security
GTU PG School,
Ahmedabad, Gujarat , India

VineetaTiwari
CDAC-ACTS
Pune Maharashtra,India

Manish Kumar Abhishek
IT Manager,
RailTel Corporation of India Ltd.
Delhi, India

Abstract **- Cloud is a leading technology used in current IT environment. Everyone is moving towards the cloud environment. On the other side High Performance Computing (HPC) is a complex requirement for many companies for their data analytics, huge computations tasks, or some artificial intelligence. As HPC requires huge set of resources, every organization cannot invest in hug datacenters or supercomputers. HPC as a service (HPCaaS) can address that issues. Moving HPC systems on the cloud environment can be cost effective for many organizations. On the other hand it can lead to some issues of performance as HPC systems are compute intensive and requires some dedicated environment. This paper addresses those issues and identifies the issues from the scheduling perspective and provides some possible solutions for them. This paper provides one scheduling algorithm that can be used to improve performance of HPC application on cloud environment and provide reconfigurable system. Using those solution inside cloud resource management module and making it HPC aware can gain great performance benefits for HPCaaS and thus enabling the small organization have their own HPC systems at the low costs.**

Index Terms**- HPC, virtualization, HPCaaS, Scheduling**

1. Introduction

Virtualization is enabling technology for the cloud computing. Cloud computing provides certain features:on-demand resource, pay-as-you-go model, scalability, elasticity, cost-effectiveness. HPC is a computing activity involving large number

of computing resources running in parallel co-operatively to complete the task. Companies that rely on HPC systems require large pool of resources to meet their computational requirements. Cost of running and maintain this large pool of resources is high. Moreover, dedicated HPC resources do not provide flexibility as cloud. To address those issues HPCaaS or virtualized HPC can be used.

Cloud provides the benefits of elasticity, flexibility and customization (through virtualization) to the HPC community. It is attracting several users who cannot afford to deploy their own dedicated HPC infrastructure due to large investments or sporadic demands. The ability to rent rather than own a cluster makes cloud cost effective. Cloud can act as a timely solution to HPC users as it does not involve cluster startup and maintenance costs and cluster creation time. It provides elastic resources which results in elimination of risks caused by under-provisioning and avoidance of wastage of resources resulting from underutilization of computing power in case of over-provisioning[2].

However, despite of the benefits of cloud not all HPC applications are suitable for running on it. Only small scale HPC applications are good candidates to run in cloud. The cloud commodity interconnects or absence of lower-latency interconnects, the performance overhead introduced by virtualization stack and the application-agnostic cloud schedulers are main obstacles for efficient execution of HPC applications in cloud. Moreover, resource requirements for traditional cloud web applications are different from the HPC applications. HPC applications are usually composed of tightly coupled processes performing frequent inter-process communication and synchronizations. So it requires low latency and high bandwidth (infiniband) inter-process communication to achieve high performance. Cloud uses a normal Ethernet interconnects which becomes bottleneck for HPC applications. So despite of the benefits of cloud it still remains unclear whether cloud can offer a viable alternative to supercomputer for HPC applications or not.

In addition to above, two less explored challenges are resource heterogeneity and multi-tenancy which are fundamental artifacts of running cloud.[7] Multi-tenancy leads to multiple sources of interference due to sharing of CPUs and memory access. For tightly coupled HPC application resource heterogeneity and multi-tenancy can result in severe performance degradation, since one slower processor can slow down the entire application. Topology awareness is also a concern for HPCaaS. As HPC application communicates frequently, time spent in synchronization can degrade the performance. If the VM placement algorithm (Scheduler or mapper) is topology aware then it can reduce the inter-VM communication delay and increase the performance. HPC application requires co-scheduling of the tasks of the same job to make synchronous progress. In current virtualized system all VMs progress independently of each other so it is required

that all VMs of a same HPC job must be scheduled together using some mechanism like gang-scheduling.

From the above discussion we assume conclude that the placement of VMs to physical machines can have significant impact on performance. With this as motivation, the primary questions we address through this research are: a) Can we improve HPC application performance in cloud through intelligent VM placement strategies or intelligent scheduling. b) How to effectively utilize a common pool of resource for efficient execution of both cloud as well as HPC application.

We identify the opportunities and challenges of VM consolidation for HPC in cloud. In this paper, we mainly focus on the challenges and alternatives for scheduling HPC applications on cloud. Through topology-awareness and heterogeneity awareness we can achieve significant improvement in performance and resource utilization. So we propose scheduling algorithm which optimizes resource allocation while being HPC-aware. We considered HPC-aware VM placement strategies specifically topology awareness and hardware awareness in Open Stack scheduler. This scheduling strategy can result in significant benefits for both HPC users and cloud providers. Using these strategies, cloud providers can utilize infrastructure more and offer improved performance to cloud users. This can allow cloud providers to obtain higher profits for their resources.

The reminder of the paper is organized as follow: Section –II discuss current scheduling scenario for cloud and HPC systems. Section –III explains the proposed scheduling algorithm. Finally conclusion and future work are left for the final section.

2. Existing System

Current cloud scheduler considers three terms: **share, reservation and limit.**[5] Share specifies the relative importance of the VM. Reservation specifies guaranteed minimum allocation for a virtual machine. Limit specifies the upper bound for CPU, memory, or storage I/O resources that can be allocated to a VM.

Current cloud providers use three different approaches for HPCaaS:

a) Consider the single request of *k* VMs as a *k* different scheduling problem.
b) Use dedicated resources for cloud and HPC applications.

According to first approach current cloud scheduler considers request of *k* VMs as a *k* different scheduling problem as shown in figure 2-1. For HPC system requirement of *k* VMs should be considered as a single request so that all VMs can be located on a single capable host or o a single capable rack.[6]

According to second approach current HPCaaS provider manages dedicated resources (clusters) for cloud applications and HPC applications. That means one resource pool for HPC and one for cloud as show in figure 2-2.

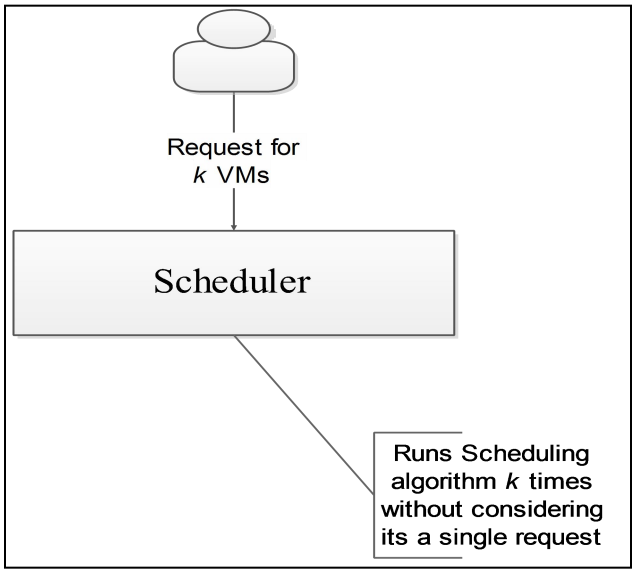

Figure 2- 1: Request for *k* VMs

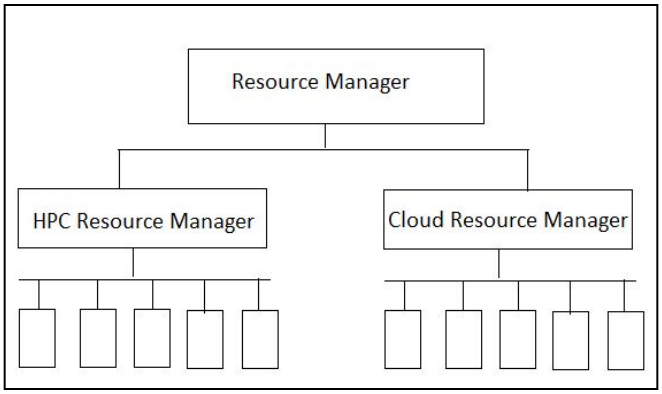

Figure 2- 2 : Dedicated resources for HPC and cloud

Above approach can lead to underutilization of resources. For example if there is no HPC application on HPC resource pool and Cloud resource pool is overloaded than still HPC resources are not used and thus underutilized. To overcome this problem whole resource must be considered as a single resource pool and can be used for both HPC as well as Cloud application and thus providing reconfigurable system.

3. An HPC-aware cloud scheduler

In this paper we address the initial VM placement problem (figure 3-1). The problem can be formulated as – Map *k* VMs each with same, fixed resource requirements to N physical servers which are unoccupied or partially occupied, while satisfying the resource requirements. Moreover, our

focus is on providing the user a VM placement optimized for HPC. For this we consider topology awareness and hardware awareness in our algorithm.

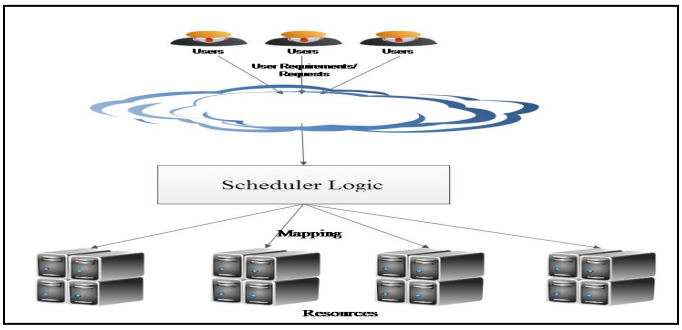

Figure 3- 1: Proposed System architecture

Algorithm-I

Step-1]

Host_capacity_list = list of capacity of unique host

Required_spec = user requirements (No of CPUs, RAM, application type)

Rack_capacity_list = capacity of each rack

Total_host = length (host_capacity_list)

Step-2]

If (app_type == HPC)

Sorted_rack_list ← sorted rack_list in decreasing order of capacity

Sorted_host_list ← sorted host list inside rack (host_capacity_list, rack_id)

Step-3] for HPC applications

Selected_Rack ← choose the best suited rack from sorted list

Selected_host ← choose the best host or hosts that satisfies the requirement from selected rack

Step-4] for cloud applications

Selected_host ← select best suited host from host_capacity_list

Step-5]

Update capacity variables.

As shown in flow chart in figure 3-2 algorithm is divided into basically two parts a) Filtering and b) mapping. In algorithm host_capacity_list contains the capacity of individual host like free CPUs, RAM and other information about host. Other term rack_capacity_list contains the list of hosts and total rack capacity. User need to submit specification of the required VMs like no of CPUs and no of RAM and application type.

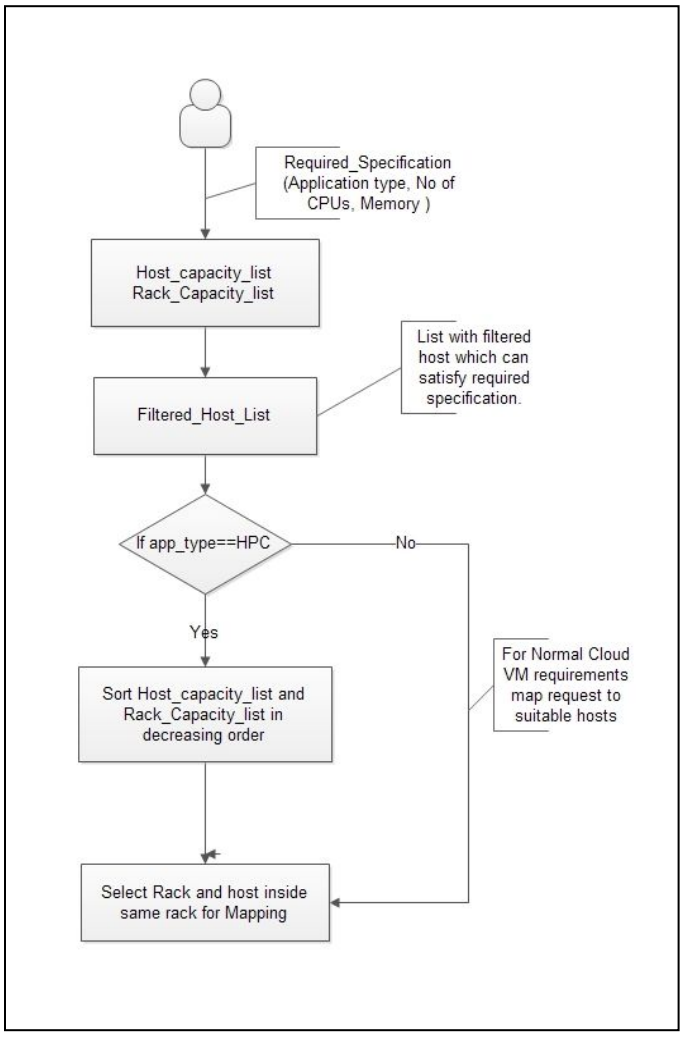

Figure 3- 2 : Scheduling flow chart

Here only two types are considered HPC and Cloud applications. For HPC application scheduler creates a cluster of required CPUs and allocates access of the head node to user. For cloud application scheduler allocates VM of required specifications. For HPC applications scheduler tries to select the single host capable of all the requirements or try to allocate multiple hosts on a single rack for less communication delay. For that it firstly filters out the hosts and racks that are not capable for handling the user requirements. Then it sorts the

list and allocates the appropriate rack and appropriate host/hosts inside that rack. Thus it provides topology aware scheduling algorithm. For normal cloud applications no need to allocate the hosts in the same rack so scheduler algorithm allocates most appropriate hosts either from the same or different racks. Above algorithm can improve the performance of an HPC application on the cloud platform and provide better resource utilization. It can run both cloud as well as HPC application on same hardware resource thus providing reconfigurable cloud (between HPC and Cloud applications).

4. Conclusion and future work

In this paper scheduling algorithm is proposed for the mapping the HPC application onto the cloud infrastructure. This algorithm provides topology awareness and deals with the processor heterogeneity issue for HPCaaS. Through this scheduling algorithm performance of HPC application can be improved on cloud and moreover reconfigurable systems for both cloud and HPC can be achieved thus leading to the proper resource utilization. This approach can be used for small scale business organizations which cannot afford the dedicated resources for the own HPC systems.

In future we plan to simulate the proof of concept for container based virtualized cloud. We plan to perform the benchmarking for the performance of HPC application with customized scheduling and compare the benefits over the current system.

5. References

[1] Joshua Bernstein, Kim McMahon –Computing on Demand—HPC as a Service High Performance Computing for High performance Business

[2] Joshua E. Simons, Jeffrey Buell- "Virtualizing High Performance Computing", ACM SIGOPS Operating Systems Review, December 2010.

[3] Andrew J. Younge, Robert Henschel, James T. Brown, Gregor von Laszewski, Judy Qiu, Geoffrey C. Fox – "Analysis of Virtualization Technologies for High Performance Computing Environments", 2011 IEEE 4thInternational Conference on Cloud Computing, July 2011

[4] Job Scheduling in HPC Clusters, Dell.

[5] vSphere Resource Management, VMware Inc.

[6] Abhishek Gupta, Laxmikant V. Kale, DejanMilojicic, Paolo Faraboschi – "HPC aware VM Placement in Infrastructure Clouds ", IEEE International conference on Cloud Engineering, March 2013.

[7] Abhishek Gupta, Laxmikant V. Kale, - "Towards Efficient Mapping, Scheduling, and Execution of HPC Applications on Platforms in Cloud"

[8] DejanMilojicic, Susanne M. Balle, "Optimizing VM Placement for HPC in the Cloud", Federated Clouds '12 Proceedings of the 2012 workshop on Cloud services, federation, and the 8th open cirrus summit

