Megh: A Private Cloud Provisioning Various IaaS and SaaS

Tushar Bhardwaj, Mohit Kumar and S.C. Sharma

Abstract Cloud computing is a collection of heterogeneous computing resources (both hardware and software) that provide various types of services over the Internet on pay-per-use basis. Therefore, number of users and service requests are increasing day by day in cloud environment. Cloud service provider runs the user application (request) parallel so that user gets the response in minimum time, and resource utilization should be maximum. In this paper, the authors have developed a private cloud computing environment called *Megh*, using OpenNebula, that is capable of hosting various IaaS and SaaS services for the end users. For now, Megh is delivering two types of SaaS (i) *Cloud-WBAN*: A pervasive healthcare system that delivers SaaS in terms of analysis service on the sensory data collected from wireless body area networks (WBAN) and (ii) *High-performance computing* (*HPC*): It is the use of parallel processing for running advanced application programs efficiently, reliably, and quickly. *Virtual Machine Provisioning*: Virtual machine provisioning privileges control activities related to deploying and customizing virtual machines. Authors have tested Megh with a case study of high-performance computing scenario. We run several instances of an application on conventional server as well as cloud environment, and computational results show that cloud environment executes the application with minimum response and makespan time.

Keywords High-performance computing · Virtual machine · Cloud computing Task scheduling

T. Bhardwaj (✉) · M. Kumar · S.C. Sharma
Wireless & Cloud Computing Lab, IIT Roorkee, Roorkee, India
e-mail: tushariitr1@gmail.com

M. Pant et al. (eds.), *Soft Computing: Theories and Applications*,
Advances in Intelligent Systems and Computing 584,
https://doi.org/10.1007/978-981-10-5699-4_45

1 Introduction

1.1 Overview of Cloud Computing

Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources (e.g., Networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1] by NIST, USA.

1.1.1 Deployment Models

The deployment models are briefly described below:

- Public Cloud: Public cloud (off-site and remote) is owned by large organization group and is available for public use. Example: Amazon Web Service, IBM Blue cloud, Sun cloud, Google App Engine, Microsoft Azure Service platform.
- Private Cloud: Private cloud (on-site) is owned and operated for the exclusive use of a particular organization. The cloud may be either on or off premises, but in most cases, it is developed on the premise for security reasons.
- Hybrid Cloud: A hybrid cloud is the combination of public, private, or community cloud. Therefore, it can be also seen as some portion of computing resources on-site (on premise) and off-site (public cloud).

1.1.2 Service Models

The service model offers various kinds of services, which take the form of XaaS, of "<Something> as a Service." Primarily, there are three kinds of service models that have been universally accepted:

- Infrastructure as a Service (IaaS): It provides virtual machines, virtual storage, and virtual hardware infrastructure to the end user for provisioning. Examples: Amazon Web Services (AWS), Microsoft Azure, Google Compute Engine (GCE).
- Platform as a Service (PaaS): It provides the virtual machines, application services, operating system, application development framework, and control structures. Examples: OpenShift, Google App Engine, Cloud Foundry.
- Software as a Service (SaaS): It is a complete set of operating environment and applications that are used by the end users. Examples: Google Apps, Salesforce, Workday, Concur, Citrix's GoToMeeting, Cisco WebEx.

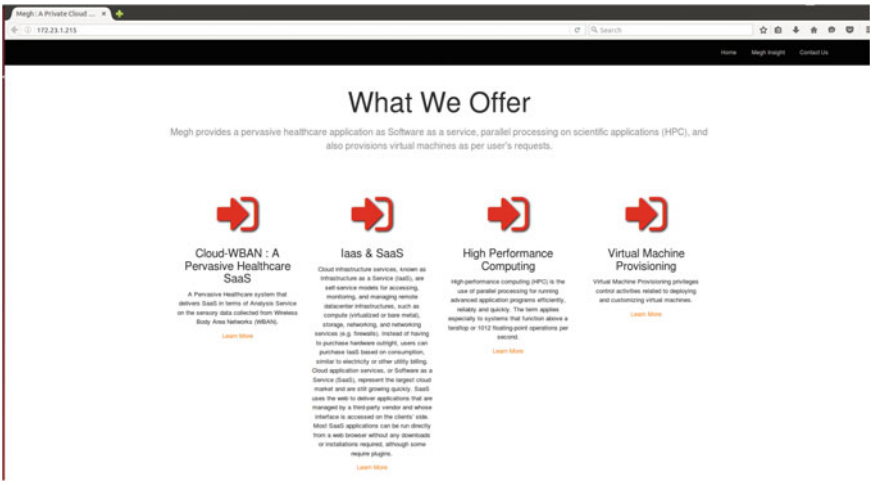

Fig. 1 Services offered by Megh

1.2 Our Contributions

In this paper, the authors have developed a private cloud computing environment called *Megh* using OpenNebula [2] that is capable of hosting various IaaS and SaaS services for the end users, Fig. 1. For now, Megh is delivering two types of SaaS (i) *Cloud-WBAN*: A pervasive healthcare system that delivers SaaS in terms of analysis service on the sensory data collected from wireless body area networks (WBAN) and (ii) *High-performance computing* (*HPC*): It is the use of parallel processing for running advanced application programs efficiently, reliably, and quickly. We run the different application parallel to different virtual machine to test our scheduling algorithm response time and makespan time. *Virtual Machine Provisioning*: Virtual machine provisioning privileges control activities related to deploying and customizing virtual machines.

The rest of the paper is structured as follows. Section 2 shows the related work about private cloud deployment using OpenNebula and task scheduling algorithms. Section 3 describes the system design and deployment of Megh cloud. Section 4 showcases the comparison between high-performance computing using conventional server and HPC. Finally, Sect. 5 includes conclusive remarks and directions of future work.

2 Related Works

There are several task scheduling algorithms that have been proposed in last decade, some of them were static and other were dynamic algorithm. Tripathy and Nayak [3] implement a deadline-based scheduling algorithm considering the parameter

average resource utilization ratio and number of job accepted or rejected using the private cloud OpenNebula, and in some cases, backfilling algorithm does not work well; therefore, authors use the analytic hierarchy process (AHP) as a decision-maker in the backfilling algorithm to choose the possible best lease from the given best effort queue in order to schedule the deadline sensitive lease. Hossain [4] implement a load-balancing algorithm at OpenNebula considering the execution time as a parameter. Author allocates the job to the virtual machine and calculates the load on each virtual machine, if any virtual machine is overloaded then transfer the task to other virtual machine. Parmar and Champaneria [5] compare and analyze the performance of different private cloud like Eucalyptus, OpenStack, OpenNebula and CloudStack for different services such as scalability, hypervisor, storage, networking, authentication. The main function of OpenNebula, Eucalyptus, and Nimbus private cloud is to manage the virtual machine at the time of provisioning IaaS services. Authors compared the performance of each system [6] and discussed common challenges in these systems like fair job scheduling in the absence of preemption, priority, deadline, cost, etc., network issues, user security, internal security, etc. Samal and Mishra proposed round-robin technique considering the parameter response time and resource utilization to solve the problem of load-balancing in cloud environment [7], but algorithm did not minimize the response time and makespan time. Lakra and Yadav [8] proposed a multi-objective algorithm to reduce the turnaround time and cost. Trade-off always occurs between cost and time parameter, but authors proposed an optimal task scheduling that improved the throughput and reduced the cost without violating the service-level agreement for an application in SaaS. Proposed algorithm will not work well after considering some qualities of service parameter such as deadline, priority of task. Thomas et al. [9] proposed an improved task scheduling algorithm after analyzing the traditional algorithm that was based on user task priority and task length. In this paper, authors does not give any special importance to high-priority task when they arrive, and consider all the important factor for task scheduling to reduce makespan time, but unable to minimize makespan time for large number of task. It is easy to implement the static and dynamic algorithm in simulation environments, but difficult to implement in real cloud environment like OpenNebula, Eucalyptus, OpenStack, Nimbus, etc. In this paper, we deployed the private cloud and run the algorithm parallel to all the virtual machine and find out the results.

3 Megh: A Private Cloud

3.1 OpenNebula: IaaS Deployment

OpenNebula is a private cloud that provides the resource with the help of Haizea as a lease manager. It contains many number of characteristics which make it a unique software for cloud resource management. One of the unique functionality of

OpenNebula is to group the host in clusters and allows the segmentation of servers according to their characteristics. One more important point regarding the job scheduling in cloud environment is that it have powerful matchmaking algorithm that is useful to balance the allocation of virtual resources in a private cloud, specially, in conjunction with clusters. Cloud is turning into a suitable computing infrastructure for delivering services-oriented computing. Many open-source cloud solutions are available for the same. Open-source software platforms offer an enormous amount of adaptability without immense licensing charges. Accordingly, open-source programming is generally utilized for deploying and provisioning cloud, and private cloud is being assembled progressively in the open-source way. "*OpenNebula* is a *cloud computing* platform for managing heterogeneous distributed *data center* infrastructures. The OpenNebula platform manages a data center's virtual infrastructure to build private, public, and hybrid implementations of infrastructure as a service. OpenNebula is free and open-source software, subject to the requirements of the Apache License version 2" [3].

3.2 Experimental Setup

We have deployed Megh on 3 HP Z620 Workstation Servers, as shown in Fig. 2. It comprises of 36 cores and 3 TB of virtualized storage and currently has over 10 virtual machines commissioned for high-performance computing delivering IaaS and SaaS services. The data center's characteristics are shown in Table 1, and virtual machine's characteristics are shown in Table 2.

Fig. 2 Megh: data center laboratory

Table 1 Data center characteristics

Item	Properties
Industry standard architecture (ISA)	×86
Operating system	Linux
Virtual machine monitor (VMM)	KVM
Storage capacity	3 Terabyte
Number of CPU	3 with 12 cores per CPU
Memory capacity	48 GB

Table 2 Virtual machine characteristics

Item	Properties
Industry standard architecture (ISA)	×86
Operating system	Linux
Virtual machine monitor (VMM)	KVM
Storage capacity	10 GB
Number of CPU	1 with 1 core per CPU
Memory capacity	1 GB

4 Case Study: High-Performance Computing

User submit the job J1, J2,…J*n* through the graphical user interface or Web interface with service requirement in terms of quality of service parameter such as deadline, cost, priority of task. When user submits the application to cloud, it is accepted by cloud resource controller. Cloud resource controller checks whether that request is coming from legitimate user or not; if request is coming from legitimate user, then it allocates the resource or provides the service otherwise it discards the request. Cloud resource controller contains the information about all the virtual machine and all the application (tasks) as shown in Fig. 3.

After that, cloud resource controller checks the following parameter:

- Upcoming request is priority-based or non-priority-based.
- Calculate the total number of request and requirement of resource (number of processing element, memory, bandwidth, etc.) by request for processing.
- Upcoming request has a deadline or not.
- User-required parameter (response time, makespan time, execution time, etc.)

After checking all the parameter, cloud resource controller stores the request in a queue based upon required parameter. If application is high-priority or deadline-based then cloud resource controller allocates such type of resource to the application that executes in minimum time, i.e., before the deadline. If application is based on lower priority then it allocates a simple virtual machine to execute them because time or deadline is not crucial factor in this case.

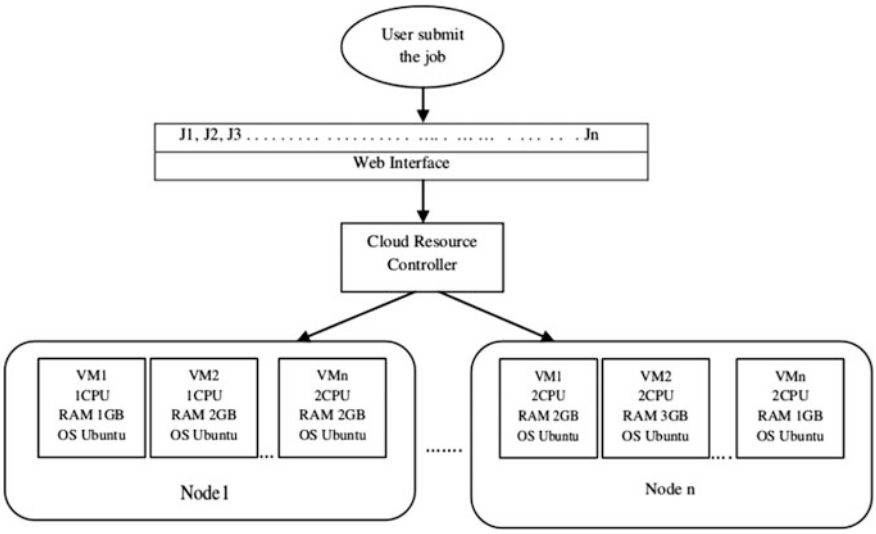

Fig. 3 Architecture of cloud

To test the functionality of Megh, authors have tested it with a case study of high-performance computing scenario. The main aim of this case study is to authenticate the credibility of using parallel processing using our private cloud Megh. The performance metric comprises of response and makespan time. Response time is the total time it takes to respond to a process request. If we ignore the transmission time, then response time is just the sum of service and waiting time. On other hand, the *makespan time* is the total time elapse from start till the end of a process execution. The above two parameters justifies the characteristics of a task to get executed in different computing environments.

For experimental purpose, we have taken a simulated application, which takes *n* seconds (depending upon its complexity and input data set) to get executed on a dedicated server. The application and scheduling algorithms are implemented using bash scripting. Firstly, we have executed several instances of that application on a machine with (4 GB RAM, 4 Cores, Ubuntu 14.04 OS) using the bash script given in Fig. 4 and calculated the response and makespan time. When such an application runs on a single machine, there will be two criteria: context switching and non-context switching. As our application is non-parallel, we opted for non-context switching mode. Moreover, context switching takes more time and leads to performance degradation. In this mode, the second application instance gets the CPU when the first instance gets completely executed.

Secondly, we have utilized HPC SaaS, offered by Megh and executed the same applications on four machines in parallel with (1 GB RAM, 1 Core, Ubuntu 14.04 OS) and calculated the response and makespan time. Here, we have scheduled the application instances to the virtual machines on first-come-first-serve (FCFS) basis —see the bash script given in Fig. 5. Here, the virtual machines can be identified by

```
#!/bin/bash
 num=numberofapplicationInstances
 for((i=1; i<=num; i++))
   do
     ssh root@172.23.1.223 'bash -s' < task$i.bash
   done
```

Fig. 4 Bash script for scheduling on conventional server

```
#!/bin/bash
 num=numberofapplicationInstances
 for((i=1; i<=num; i++))
  do
     ssh root@172.23.1.219 'bash -s' < task$i.bash &
     if [ $i -eq $num ]
     then
     break
     fi
     i=`expr $i + 1`
     ssh root@172.23.1.220 'bash -s' < task$i.bash &
     if [ $i -eq $num ]
     then
     break
     fi
     i=`expr $i + 1`
     ssh root@172.23.1.221 'bash -s' < task$i.bash &
     if [ $i -eq $num ]
     then
     break
     fi
     i=`expr $i + 1`
     ssh root@172.23.1.222 'bash -s' < task$i.bash &
     if [ $i -eq $num ]
     then
     break
     fi
  done
```

Fig. 5 Bash script for FCFS Scheduling on HPC

the IP addresses starting from 172.23.1.219 to 172.23.1.222. This can be scaled up and down with respect to the user demand and SLA requirements. The application instances are given by task$i.bash.

Each virtual machine has its own buffer that stores the tasks for further execution. Figure 6 shows the comparative analysis of response time between the two approaches. It can be clearly seen that using parallel approach the response time is quite improved. On the other hand, Fig. 7 showcases the variation in the makespan time of application instances running on conventional server and HPC. Using Megh's HPC an average speedup factor of 2 has been observed in makespan time.

Fig. 6 Response time comparison

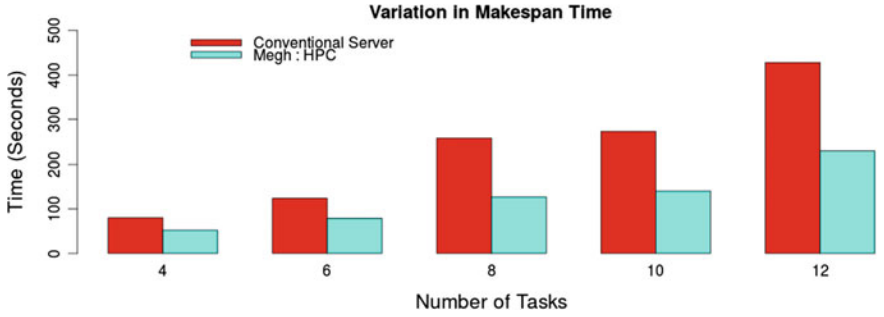

Fig. 7 Makespan time comparison

5 Conclusion

There exist various open-source cloud platforms such as OpenNebula, Eucalyptus, Nimbus, OpenStack. We develop a private cloud environment 'Megh' using the OpenNebula because it is easy to deploy the software in OpenNebula, and it is more flexible compared to all other private cloud. We test the task scheduling algorithm at Megh and compare the computational results (as shown in Figs. 6 and 7) with the algorithm running at conventional server. Our controller node distributes the task in parallel to all running virtual machines, so that all the tasks will get executed in minimum time as compared to conventional server. We will try to implement dynamic task scheduling algorithm and auto-scaling mechanism at Megh in future.

References

1. Mell, P., Grance, T.: The NIST definition of cloud computing. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (2011)
2. OpenNebula. https://en.wikipedia.org/wiki/OpenNebula

3. Tripathy, C., Nayak C.: Deadline sensitive lease scheduling in cloud computing environment using AHP. J. King Saud Univ. Comput. Inf. Sci. (2016)
4. Moniruzzaman, A.B.M., Nafi, K.W., Hossain, S.A.: An experimental study of load balancing of OpenNebula open-source cloud computing platform. In: 2014 International Conference on Information. Electronics & Vision (ICIEV). IEEE (2014)
5. Chavan, S.V., Tilekar, S.K., Ladgaonkar, B.P.: International Journal of Advanced Research in Computer Science and Software Engineering. Int. J. **5**(12) (2015)
6. Sempolinski, P., Thain, D.: A comparison and critique of eucalyptus, openNebula and nimbus. In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science (Cloudcom). IEEE (2010)
7. Samal, P., Mishra, P.: Analysis of variants in round Robin algorithms for load balancing in cloud computing. Int. J. Comput. Sci. Inf. Technol. **4**(3), 416–419 (2013)
8. Lakra, A., Yadav, D.: Multi-objective tasks scheduling algorithm for cloud computing throughput optimization. Procedia Comput. Sci. **48**, 107–113, 2015
9. Thomas, A., et al.: Credit based scheduling algorithm in cloud computing environment. Procedia Comput. Sci. **46**, 913–920 (2015)