A study of an Improvement in Utilization of Clouds Infrastructure

Mr.Dattatray.R.Kale, Mr.Dhanaji.S.Barawade, Mrs.T.H.Patil

[1] *Lecturer, Department of Computer Engineering, AMP Vathar Tarf Vadgaon, Maharashtra, India.*
[2] *Lecturer, Department of Computer Engineering, AMP Vathar Tarf Vadgaon, Maharashtra, India.*
[3] *Lecturer, Department of Computer Engineering, AMP Vathar Tarf Vadgaon, Maharashtra, India.*

ABSTRACT

As with all cloud computing services it provides access to computing resources in virtualized environment. In case of Infrastructure as a Service (IaaS) the computing resources provided is specifically that of virtualized hardware that means computing infrastructure. It offers various virtual server space and network connections so the client can build their own IT platforms. Infrastructure as a Service (IaaS) cloud provides users on-demand access to the various resources. However for on-demand access the cloud providers considerably enlarge their infrastructure with increasing high price for various operating resources with low utilizations. Sometimes cloud providers reject a user's request in which case the access is no longer on demand. We are reviewing a cloud infrastructure that combines on demand allocations of resources with opportunistic provisioning of cycles from idle cloud nodes to other processes by deploying backfill Virtual Machines (VMs).A shared infrastructure between IaaS cloud providers and a High Throughput Computing (HTC) job management systems is giving benefits to both the IaaS cloud providers and HTC users by increasing the proper utilizations of the cloud infrastructures with decreasing the overall cost. This paper provides a short overview of existing standards and technologies which are refers for ongoing projects.

Keyword : - *Cloud Computing, Infrastructure-as-a-Service, Backfill Virtual Machines, High Throughput Computing.*

1. INTRODUCTION

Infrastructure-as-a-Service (IaaS) is one of the basic service models of cloud computing alongside Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). As with all cloud computing services it gives access to computing resources in virtualized environment. The client can access the virtualized components and build their own IT platforms. In current years Infrastructure-as-a-Service (IaaS) cloud computing [2] gives a gorgeous alternative to the purchase and management of physical resources. Effective use of existing network and IT infrastructure can be achieved by providing combined network and IT resources on demand as an infrastructure services that are capable of supporting complex technological processes. The on-demand provisioning gives the permission to the users to dynamically expand their work and allocate the resource base available to them based on the immediate needs. This pattern is very useful to the organizations to configure their private cloud that allow their users a faultless transition community or commercial clouds supporting well-matched virtual machine images and cloud interfaces. This type of private clouds are normally configured by using open sources IaaS performances such as Nimbus[3],Eucalyptus[4].But such a private cloud installations also face utilization problems. So for satisfying the on-demand availability a provider needs to keep a large number of nodes idle so that they can be used to satisfy an on-demand request .If all these nodes keep idle it shows low utilizations. The only way to improve it is to keep fewer nodes idle. But this means a provider no longer provides on-demand computation. This low utilization is creating a problem in the adoption of cloud computing in the scientific world.

There having many systems which work as volunteer computing systems which are capable of taking advantages of resources available opportunistically. The use of High Throughput Computing (HTC) is one of the example of scientific world. For many research and engineering projects, the quality of the research or the product is heavily dependent on the quantity of computing cycles available. It is not uncommon to find problems that require

weeks or months of computation to solve. Scientists and engineers engaged in this sort of work need a computing environment that delivers large amount of computational powers over a long period of time. Such an environment is called High Throughput Computing (HTC) environment. The HTC community is concerned with strength and consistency of jobs over a long time scale. That is being able to create a reliable system from unreliable components. This research is similar to transaction processing, but as much larger and distributed scale. Some HTC systems, such as HTCondor [5] and PBS (Portable Batch Systems) can run tasks on opportunistic resources. These types of applications are designed to scavenge unused resource cycles, here we are giving one example, when a user stops using their desktop, and the screensaver might use the resource to run a volunteer computing program. The job may then be preempted when resource becomes busy, in which case the job is normally re-queued and rescheduled on another accessible resource by the HTC system that manages it [1]. In this paper awe implement a cloud infrastructure that combines opportunistic provisioning of idle cloud nodes with on demand resource allocation.

2. BACKFILL VM'S

We are studying a cloud infrastructure that combines on demand allocation of resources with opportunistic provisioning of cycles from idle cloud nodes to other processes, such as HTC, by deploying backfill VM's. If resources necessitate is satisfied by on demand request then a backfill VM is terminated. A Backfill VM's can be provided to users at a lower cost than a on-demand VM's because of the cloud providers capacity to stop the instances when needed, thus for users that work with HTC resources and perhaps expect such behavior already, backfill VM's would provide a smaller amount expensive option when moving their workloads to the cloud. So by using this we can creates a path to higher utilized clouds. Here we are also use the Nimbus toolkit[4] which is used to install backfill VM's on idle Virtual Machine Monitor nodes[6].Nimbus is an open source toolkit for deploying IaaS clouds. It allows a client to lease remote resources by deploying virtual machines on those resources and configuring them to represent an environment desired by the user. Nimbus is comprised into two products,

- Nimbus Infrastructure is an open source EC2/S3 compatible Infrastructure-as-a Service implementation purposely targeting features of awareness to scientific community such as support for proxy credentials, batch schedulers, best effort allocations and others.
- Nimbus platform is an integrated set of tools, operating in a multi-cloud environment, that deliver the power and versatility of infrastructure clouds to scientific users.

The Nimbus cloud client permits the user to condition customized work out nodes called a workplace, and preserve a full control. The Nimbus cloud-computing infrastructure allows scientists working on data-intensive research to create and use such virtual machines with a cloud provider. Nimbus also allows users to create multiple virtual machines to complete specific computational jobs that can be deployed throughout the cloud and still work in tandem with each other. This suppleness allows a user to configure a virtual machine and then connect it to resources on a cloud, regardless of who is providing the cloud. To explain how the system works we are studying the configuration of backfill VMs as Condor workers that integrate with the condor pool to process HTC jobs. Here we assess the ability of the system to boost utilization of the IaaS cloud infrastructure without sacrificing the ability of the IaaS cloud to provision resources on-demand. We also come across the capability of the system to put in the cycles

3. GENERAL APPROACH

The backfill implementation used in this paper was an initial prototype created to demonstrate of usefulness of combing IaaS cloud infrastructure resources with other purposes, such as HTC, through backfill VM's. The prototype implementation used in this work is publically available on GitHub [7].

An on demand user is a user that requests on demand VM's from an IaaS cloud. An on-demand VM is an IaaS VM that has been provisioned via on on-demand charter for a specific user. A backfill VM is a VM that has been deployed repeatedly by the IaaS cloud manager on an idle IaaS node using a preemptible lease. An IaaS cloud administrator is the person or persons responsible for configuring and managing the IaaS cloud resource.HTC user is a user that submits jobs to an HTC queue and HTC worker is a system that process jobs from the HTC queue.

In the IaaS clouds, backfill VM's are common VM's deployed on IaaS resources using a preemptible lease that may be configured to execute any function. Backfill VM's have two major restrictions; First backfill VM's may be terminated unexpectedly in order to free up space for the IaaS cloud manager to service an on-demand lease. Second, because of the random timing of on-demand leases, unpredictable number of backfill VM's may be accessible at any given time. Thus, we suppose that applications executing within backfill VM's are designed to handle environments that contain a variable number of workers that may join or leave the system at any time.

We are using the open source Nimbus cloud computing toolkit, which provides on-demand access to resources, to support the deployment of preemptible leases on idle cloud nodes. For the proper implementation we create a some assumptions. First the Nimbus administrator must configure backfill. On demand cloud users can not choose to install backfill VM's. Second, the present implementation is able of using only a single backfill VM image per VMM node. Different backfill VM images could be deployed on different VMM nodes, allowing numerous backfill VM images to work within the same IaaS cloud, each performing dissimilar functions. Finally, our backfill implementation cleanly shuts down the backfill VM. Clean shutdown requires additional time over trashing the VM, however performing a clean shutdown notifies services and applications running inside the backfill VM's that the VM will be terminated, allowing them to respond properly.

4. RELATED WORK

Volunteer computing systems, such as BOINC[8], harvest cycles from idle systems distributed across the internet. Major examples of volunteer applications include SETI@Home[9] and Folding@Home[10].These applications are designed to accommodate interruptions in service since widely distributed computers, operated by a seemingly infinite number of disparate users, cannot provide any guarantee of service. Much research on volunteer computing focuses on the usefulness, efficiency, and failure prediction of these volatile environments [11],[12].

As our work utilizes backfill to achieve high utilization of an IaaS infrastructure, it is different from work that uses backfill scheduling to increase the utilization of large supercomputers [13].Scheduling on supercomputers does not typically assume that backfill jobs will be preempted by an on-demand request, seeking to immediately access the resources, while our work assumes this to be the default case. Instead these backfill scheduling algorithms only attempts to backfill unused resources with requests that match the available slots both in their res ource needs as well as their expected runtime.

Spot pricing is another work related to our work. This is exemplified by Amazon[14]. With spot pricing users place bids for examples and the cloud provider occasionally adjusts the price of spot instances, terminating the spot instances with bids that fall below the new spot price and launching instances that meet or exceed the spot price.

5. CONCLUSIONS

In this paper we shows a cloud infrastructure that combines on-demand allocation of resources with opportunistic provisioning of cycles from idle cloud nodes to other processes, such as HTC, by deploying backfill VM's. The open source Nimbus IaaS toolkit is used to deploy backfill VM's on idle cloud nodes.

6. REFERENCES

[1]. Paul Marshall, Kate Keahey and Tim Freeman ,"Improving Utilization of Infrastructure Clouds" IEEE/ACM Cloud Computing May 2011.

[2] Michael A, et al., "Above the Clouds: A Berkeley View of Cloud Computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb 2009.

[3] Nimbus. [Online]. Retreived December 6, 2010, from: http://www.nimbusproject.org/

[4] Nurmi D, Wolski R, Grzegorczyk C, Obertelli G, Soman S, Youseff L, and Zagorodnov D. Eucalyptus opensource cloud-computing system. In CCA08: Cloud Computing and Its Applications, 2008.

[5] Tannenbaum T, Wright D, Miller K, Livny M. Condor - A Distributed Job Scheduler, in Thomas Sterling, editor, Beowulf Cluster Computing with Linux, The MIT Press, 2002.

[6] Snell Q, Clement M, and Jackson D. Preemption based backfill. In Feitelson, Rudolph, and Schwiegelshohn, editors, Job Scheduling Strategies for Parallel Processing, pages 24–37. Springer Verlag, 2002. Lect. Notes Comput. Sci. vol. 2537.

[7] Nimbus 2.6 Backfill (prototype). GitHub. [Online]. Retreived December 6, 2010, from: https://github.com/pdmars/nimbus/tree/backfill-2.6.

[8] FutureGrid. [Online]. Retreived December 6, 2010, from: http://futuregrid.org/

[9] Amazon Web Services. Amazon.com, Inc. [Online]. Retreived December 6, 2010, from: http://www.amazon.com/aws/

[10] Larson SM, Snow CD, Shirts M, Pande VS. Folding@Home and Genome@Home: Using distributed computing to tackle previously intractable problems in computational biology. Computational Genomics, Horizon Press, 2002.
[11] Acharya A, Edjlali G, and Saltz J. "The Utility of Exploiting Idle Workstations for Parallel Computation," SIGMETRICS '97, pp. 225-34
[12] Anderson D and Fedak G. "The Computational and Storage Potential of Volunteer Computing," CCGRID'06, 2006, p. 73-80.
[13] Feitelson DG, Rudolph L. Parallel job scheduling: Issues and approaches. Lecture Notes in Computer Science: Job Scheduling Strategies for Parallel Processing, 949, 1995
[14] Amazon Web Services. Amazon.com, Inc. [Online]. Retreived December 6, 2010, from: http://www.amazon.com/aws/