Quality High Performance Over Infratructure Cloud Using Snapshots of Resources

[1]B. Kalyan Chandra, [2]D. Srinivas
[1,2]Dept. of CSE, Kakinada Institute of Engineering & Technology, Korangi, AP, India

Abstract
A key advantage of Infrastructure-as-a-Service (IaaS) clouds is providing users on-demand access to resources. However, to provide on-demand access, cloud providers must either significantly overprovision their infrastructure (and pay a high price for operating resources with low utilization) or reject a large proportion of user requests (in which case the access is no longer on-demand). At the same time, not all users require truly on-demand access to resources. Many applications and workflows are designed for recoverable systems where interruptions in service are expected. For instance, many scientists utilize High Throughput Computing (HTC)-enabled resources, such as Condor, where jobs are dispatched to available resources and terminated when the resource is no longer available.
We propose a cloud infrastructure that combines on-demand allocation of resources introducing the snapshot of clouds using virtual machines. this mechanisms increases the throughput of on demand accesses of entire network. The performances of the sytem automatically increase by utilizing the snapshots of clouds. We demonstrate that a shared infrastructure between IaaS cloud providers and an HTC job management system can be highly beneficial to both the IaaS cloud provider and HTC users by increasing the utilization of the cloud infrastructure and contributing cycles that would otherwise be idle to processing HTC jobs.

Keywords
Cloud Computing, Infrastructure-as-a-Service, High Throughput Computing, VMware

I. Introduction

In the recent years, Infrastructure-as-a-Service (IaaS) cloud computing [14] has emerged as an attractive alternative to the acquisition and management of physical resources. The on demand provisioning it supports allows users to elastically expand and contract the resource base available to them based on an immediate need – a pattern that enables a quick turnaround time when dealing with emergencies, working towards deadlines, or growing an institutional resource base. This pattern makes it convenient for institutions to configure private clouds that allow their users a seamless or near seamless transition to community or commercial clouds supporting compatible VM images and cloud interfaces. Such private clouds are typically configured using open source IaaS implementations such as Nimbus [16] or Eucalyptus [17]. However, such private cloud installations also face a utilization problem. In order to ensure on-demand availability a provider needs to increase the efficiency of IaaS. We are using the snapshot based virtual cloud resources..The clouds are updated periodically. typically ensures high utilization [25] and thus much better resource amortization. Thus, potential low utilization constitutes a significant potential obstacle to the adoption of cloud computing in the scientific world. At the same time, while the on-demand capabilities of IaaS clouds are ideal for many scientific use cases, there are others that do not necessarily require on-demand access to resources. Many systems, specifically volunteer computing systems such as SETI@Home [4] and Folding@Home [11], are capable of taking advantage of resources available opportunistically and are also preemptible, i.e., designed as failure resilient systems where interruptions in service can be handled without compromising the integrity of computation. One example in the scientific community is the use of high throughput computing (HTC), as implemented by e.g., the Condor [24] system where users employ HTC-enabled resources to process their workloads.
In our work we extend the Nimbus toolkit [16] to deploy backfill VMs on idle Virtual Machine Monitor (VMM) nodes[21]. Nimbus is an open source toolkit for deploying IaaS clouds, designed with extensibility in mind, which makes it particularly suitable for projects such as the one described here. The proposed system overcomes the difficulties the previous system .

II. Related Work

A compute infrastructure cloud operates by allowing a user to make leases against its pool of resources; an infrastructure lease makes a resource available to the user based on set of lease terms defining the availability, capacity and general conditions of a lease. In our system we focus on investigating two types of leases: On-demand, non-preemptible and flexible leases give a access to a resource within interactive time of making the request and make the resource available for an agreed-upon period of time. The user can deploy any VM compatible with the system. Opportunistic, preemptible and pre-set leases give a user access to a resource at an indeterminate time and make the resource available to the user for an indeterminate amount of time. Further, this resource is pre-defined for the user by the cloud administrator, i.e. the user cannot provide his or her own VM.
In the rest of the paper we will be referring to them as ondemand leases and preemptible leases respectively. We define the following roles in our system. An on-demand user is a user that requests on-demand VMs/leases from an IaaS cloud. An on-demand VM is an IaaS VM that has been provisioned via on on-demand lease for a specific user. A backfill VM is a VM that has been deployed automatically 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. An HTC user is a user that submits jobs to an HTC queue and an HTC worker is a system that process jobs from the HTC queue. In the context of IaaS clouds, backfill VMs are generic VMs deployed on IaaS resources using a preemptible lease that may be configured to perform any function. Backfill VMs have two major constraints. First, backfill VMs may be terminated suddenly in order to free up space for the IaaS cloud manager to service an on-demand lease. Second, because of the unpredictable timing of on-demand leases (from any number of unique users), a variable number of backfill VMs may be available at any given time. Thus, we assume that applications executing inside backfill VMs are designed to handle environments that contain a variable number of workers that may join or leave the system at any time. Certain applications are not well suited for these volatile environments, for example, parallel applications

that require all processes to be present for the duration of the application's execution and lack checkpoint/restart capabilities. An example of a system that is well-suited for backfill VMs are High Throughput Computing (HTC) workloads [24]. HTC workloads typically consist of a large number of jobs that eventually need to be processed. An assumption of HTC workloads is that they do not have an immediate deadline and therefore do not need resources to be available at a particular time. In addition, terminating individual HTC jobs in the middle of execution and requeuing them for later execution is an acceptable action as long as the system is eventually able to process the workload.

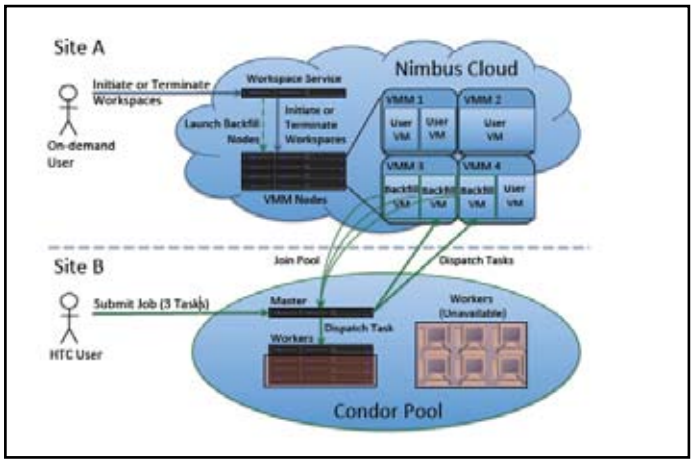


Fig. 1: Examples Backfill Deployment

A. Architecture

Fig. 1 is a simple example deployment consisting of an IaaS Nimbus cloud using backfill VMs to run Condor HTC jobs in order to increase cloud infrastructure utilization. In this example the HTC user submits 3 individual tasks to the Condor master, which is able to schedule 1 task immediately on a local worker. However, because the remaining resources in site B's Condor pool are busy, Condor leverages the cycles provided by site A's backfill VMs to launch the other 2 tasks immediately.

Condor is an ideal candidate for such a deployment because of its original design as a cycle-scavenger where idle desktop machines execute jobs until the system's user returns, after which the job is either migrated to another system or prematurely terminated and re-launched on another system. A backfill deployment could be much more complex then the example depicted in Figure 1. For instance, backfill VMs could host a number of different backfill processes in addition to Condor workers. The Condor workers could also be configured to execute jobs from multiple Condor pools or sites instead of a single pool, as shown in the figure. IaaS cloud administrators must consider a variety of different backfill configurations when deploying a backfill solution on an IaaS cloud. The configuration is influenced by the characteristics of the underlying physical IaaS resources, the users of on-demand leases, and the users of preemptible leases.

Volunteer computing systems, such as BOINC [2], harvest cycles from idle systems distributed across the Internet. Major examples of volunteer applications include SETI@Home [2] and Folding@Home [11]. 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. In the case of volunteer computing systems interruptions in service are usually the result of users returning to their systems to do work, systems crashing, or systems becoming disconnected from the Internet. Much research on volunteer computing focuses on the usefulness, efficiency, and failure prediction of these volatile environments [1,3,8,19]. Our work focuses on providing cycles within an IaaS infrastructure that would have otherwise been idle to other processes, such as HTC or volunteer computing, where the services may be interrupted by the arrival of requests for on-demand VMs. Applications that leverage volunteer computing systems would be ideal candidates for backfill VMs because of their ability to handle unexpected failures in service. In [23] we also leverage recovery techniques, specifically suspending and resuming VMs, to achieve high utilization of IaaS cloud infrastructures. While the goal of maintaining high utilization via introducing different types of leases is the same as the work described here, the leases themselves as well as the recovery technique used, specifically that of suspending and resuming VMs, is different from the focus in our work. Instead of using suspend/resume to support advanced reservations we leverage a recovery system that uses resubmission (Condor) to ensure that high utilization is achieved and no work is lost. Another area that shares related themes to our work is spot pricing, as exemplified by Amazon [2]. With spot pricing users place bids for instances and the cloud provider periodically 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. Our work uses the current demand for on-demand user VMs to determine the availability for backfill VMs while Amazon bases availability of spot instances on a spot price.

III. Proposed Model

We extend the open source Nimbus cloud computing toolkit, which provides on-demand access to resources (in the form of VMs), to support the deployment of preemptible leases on idle cloud nodes, also referred to as backfill. We make a number of simplifying assumptions in our current implementation. First, the Nimbus administrator must configure backfill. On-demand cloud users cannot elect to deploy backfill VMs. Second, the current implementation is capable of using only a single backfill VM image per VMM node. Different backfill VM images could be deployed on different VMM nodes, allowing multiple backfill VM images to operate within the same IaaS cloud, each performing different functions.

We considering the above situation and we investigated a new method for proper utilization of on-demand resources. The system creates a models for the resource sharing on cloud environment. Based on the models the proposed system creates the snapshots of the resources to provide accurate resource allocation.

The snapshots of the resources are automatically updated using different scheduling updating techniques. the updates of the resources are based on peak time and non-peak times of the application resources sharing.

We are using metrics for calculating peak time and non-peak times according the resources accessing for updating the snapshots of resources automatically. By implementing the this method the throughput of entire cloud system increases with compare to the other work implementation. The system follows the parallel processing.

The following fig. 2. illustrates the work nature of the extend previous work with the new proposed system. The proposed system incorporates the previous system work nature.

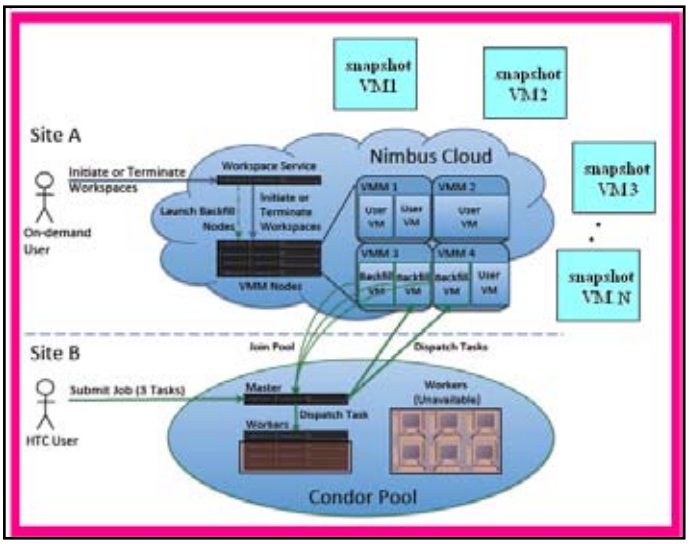


Fig. 2: The proposed Snapshot VMware in Cloud Environment

The proposed system maintains the snapshots created over the cloud resources by using Virtual machines .The snapshots of the resources provides the data access to the different user on-demand access. This method increases the performances of the cloud environment. The following figure 3. Shows the performance evolution over the different methods. We are comparing the system with the previous system.

IV. Evaluation

For the evaluation we deploy a backfill-enabled version of Nimbus 2.6 on FutureGrid [8]. Nimbus runs on a cluster of 16 VMM nodes with 2.4 GHz 8-core Intel Xeon processors and 24 GB of RAM with 20 GB allocated for user VMs, allowing for a total of 128 single-core VMs. The Nimbus workspace service node runs on an additional node. The Nimbus workspace service listens for incoming on-demand user requests for VMs and launches or terminates the VMs on the VMM nodes. This node also hosts the user VM image repository.

In our experiments, we assume that a single backfill VM utilizes the entire VMM node (all 8 cores). We choose this level of granularity in order to reduce virtualization overhead for backfill VMs and avoid additional network contention caused by transferring a backfill VM image over the network.

For the evaluation we define the following metrics:

Utilization is the percentage of user cycles consumed by CPU cores on the VMM nodes in the IaaS cloud that are either running an HTC job or running an on-demand user VM. Because backfill launches VMs on any idle VMM node, regardless of the presence of HTC jobs, it is possible for the entire IaaS infrastructure to be running backfill VMs on all VMM nodes but still have 0% utilization. For our evaluation backfill VMs must be running Condor jobs for them to contribute to the overall utilization of the infrastructure.

• First queued time is the amount of time that elapses between the time when a Condor job is submitted and when it first begins executing.

• Last queued time is the amount of time that elapses between the time the Condor job is first submitted and the time the Condor job finally begins executing for the last time before completing successfully. We note that it is possible for backfill VMs to be terminated by the deployment of on-demand user VMs, preempting Condor jobs executing in backfill VMs, and thus requiring their resubmission. While this may happen to a Condor job any number of times, it is presumed that the job will eventually be able to execute successfully to completion.

• User VM service response time is the amount of time it takes the Nimbus service to respond to an on-demand user request, i.e., the time between when the service first receives the request and the time it determines whether a VM will be launched or that the request will be rejected. This time does not include the amount of time that it takes to actually boot the on-demand user VM or propagate the VM image, only the amount of time it takes the service to determine whether or not the request will be handled. If backfill is enabled and backfill VMs need to be terminated to deploy an on-demand user VM, the user VM service response time will include the necessary time to terminate backfill VMs.

The following figures show the environmental differences between the previous system and the proposed Snapshot based Virtual Machines over the entire cloud environment.

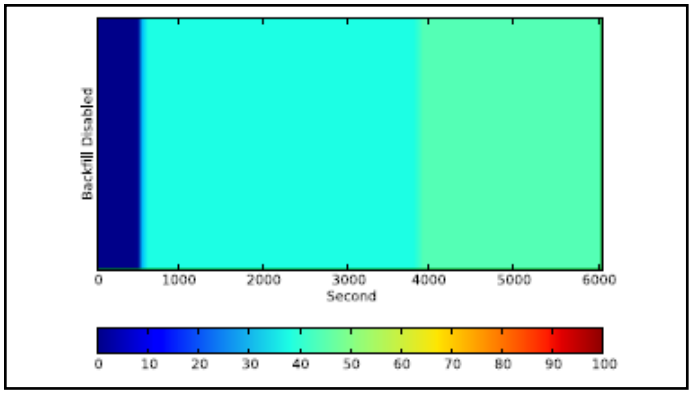


Fig. 3: IaaS Cloud Infrastructure Utilization without Backfill VMs, Running On-Demand user VMs

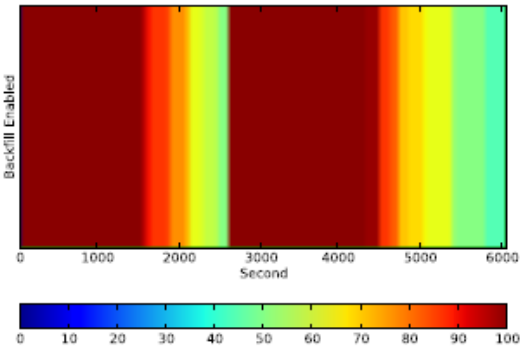


Fig. 4: Iaas Cloud Infrastructure Utilization with On- Demand user VMs and Back VMs Processing Conder Jobs

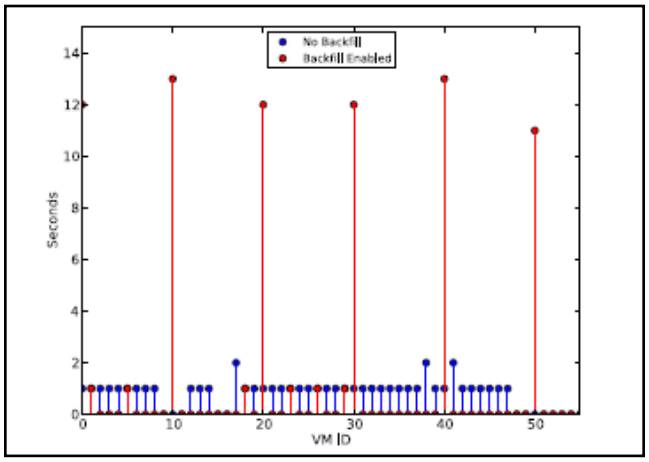


Fig. 5:

Performances differences between the previous system and the proposed system.

V. Conclusions

In this paper we propose a cloud infrastructure that combines on-demand allocation of resources using snapshot of cloud infrastructure for user on-demand accesses. We evaluate the backfill solution using an on-demand user workload and an HTC workload. We find backfill VMs contribute to an increase of the utilization of the IaaS cloud infrastructure Additionally, backfill VMs make available cycles that would have otherwise been idle to assist in processing HTC jobs. we are extending this approach towards the implementation to increase the high availability of cloud environment.

References

[a] Paul Marshall Kate, Tim Freeman,"Improving Utilization of Infrastructure Clouds".

[1] Acharya A, Edjlali G, Saltz J.,"The Utility of Exploiting Idle Workstations for Parallel Computation", SIGMETRICS '97, pp. 225-34.

[2] "Amazon Web Services (2010)", Amazon.com, Inc. [Online]. Retreived December 6, 2010, [Online] Available: http://www.amazon.com/aws/

[3] Anderson D, Fedak G.,"The Computational and Storage Potential of Volunteer Computing", CCGRID'06, 2006, pp. 73-80.

[4] Anderson DP, Cobb J, Korpela E, Lebofsky M, Werthimer D. SETI@home,"An Experiment in Public-Resource Computing". Communications of the ACM, 45(11), November 2002, pp. 56-61.

[5] Anderson, D.,"BOINC: A System for Public-Resource Computing and Storage", 5th IEEE/ACM Workshop on Grid Computing, Nov. 2004.

[6] Douglas Thain, David Cieslak, Nitesh Chawla,"Condor Log Analyzer", [Online] Available: http://condorlog.cse.nd.edu, 2009.

[7] Feitelson DG, Rudolph L.,"Parallel job scheduling: Issues and approaches", Lecture Notes in Computer Science: Job Scheduling Strategies for Parallel Processing, 949, 1995.

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

[9] Internet Retailer Magazine. [Online]. Retreived December 6, 2010, from: http://www.internetretailer.com/top500/list/

[10] Keahey, K., I. Foster, T. Freeman, X. Zhang,"Virtual Workspaces: Achieving Quality of Service and Quality of Life in the Grid", Scientific Programming Journal, Vol. 13, No. 4, 2005, Special Issue: Dynamic Grids and Worldwide Computing, pp. 265-276.

[11] 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.



Mr. B. Kalyan Chandra is a student of Kakinada Institute of Engineering & Technology (KIET), Korangi. Currently he is Pursuing his M.Tech (SE) (10B21D2508) From this college. He received his graduation From Aditya engineering College surampalem In The year 2009. His Areas of interest are cloud computing.



Mr. D.Srinivas is working as Assistant Professor in KIET. He has five years of Teaching experience. He completed his B.tech ffrom KIET in 2007.He completed his M.Tech from GIET Rajahmundry in 2010. His Areas of interests are DBMS & Networks He had Published his paper in International Journal of computer science & Technology.