2nd IEEE International Conference on Cloud Computing Technology and Science

Initial Findings for Provisioning Variation in Cloud Computing

M. Suhail Rehman   Majd F. Sakr

School of Computer Science
Carnegie Mellon University in Qatar
Education City, Doha, State of Qatar
{suhailr,msakr}@qatar.cmu.edu

***Abstract*— Cloud computing offers a paradigm shift in management of computing resources for large-scale applications. Using the Infrastructure-as-a-service (IaaS) cloud computing model, users today can request dynamically provisioned, virtualized resources such as CPU, memory, disk, and network access in the form of virtualized resources. The client typically requests resources based on computational needs and pays for resource instances based on their capacity and time utilized. Mapping these virtual resource requests to physical hardware could vary for identical requests. This can potentially cause variations in the performance of applications deployed on such resources. The performance of the application can vary according to the physical layout of the provisioned hardware (the number of virtual machines (VMs), the size/configuration of the VMs and the inter-VM locality). In this paper, we study the effects of this "provisioning variation" and its impact on application performance using suitable benchmarks as well as demonstrate their effect on a few MapReduce workloads. Our initial findings indicate that provisioning variation can impact performance by a factor of 5 primarily due to I/O contention.**

Keywords- provisioning variation, performance evaluation, provisioning, IaaS

I. INTRODUCTION

The interest in cloud computing has grown significantly as users want to take advantage of the benefits of accessing computing as a service. Many different aspects of computing can be offered as a service, three popular ones are software, platform and infrastructure (Figure 1). The benefits from cloud computing include significant reduction of upfront cost, increased utilization through sharing of physical resources, ease of resource management, flexibility and elasticity of provisioning resources, ease of programmability and management of distributed applications and pay-as-you-go economic model.

All service layers typically include the *infrastructure as a service* (IaaS) model. The service is offered through provisioning virtualized physical resources in order to satisfy different computing requests from potentially different users. Through virtualization and provisioning, the physical resources are shared across multiple compute requests.

A request for virtualized resources is described through a set of parameters detailing the processing, memory and disk needs. Provisioning satisfies the request by mapping virtualized resources to physical ones. The mapping of physical resources is not uniform for requests with identical parameters. This variation in resource mapping is what we refer to as *provisioning variation*. In this paper, we will highlight the potential impact of provisioning variation on the performance of applications running on the cloud.

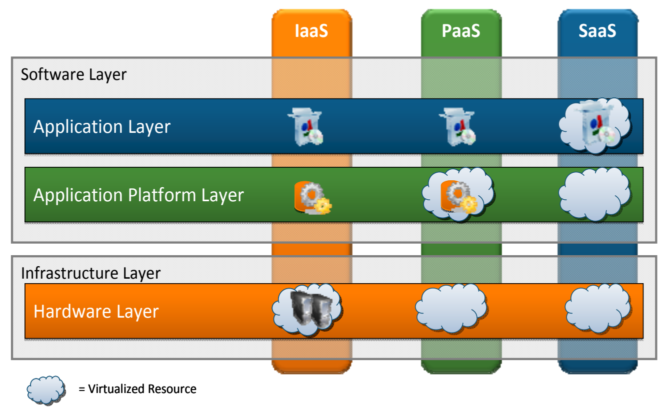

Figure 1. Three Cloud Service Models [22]

II. INFRASTRUCTURE AS A SERVICE

A widely used classification of cloud computing services is based on the type of services they provide, as indicated in Figure 1. Services that provide computational resources (such as processor, memory and disk) packaged in the form of Virtual Machines (VMs) instances are popularly referred to as *infrastructure as a service* or IaaS. Among public IaaS service providers, Amazon Web Service's Elastic Compute Cloud [1, 2] is a pioneer, and similar offerings are available from RackSpace, GoGrid, Microsoft and others.

In addition to public cloud infrastructures, many organizations are seeing benefits in migrating their traditional data centers and server clusters into virtualized private clouds. There are multiple solution providers in this space, typically deployed with hypervisors such as Xen[3] or vSphere[14]. They are able to reduce resource deployment times from days to minutes.

The Infrastructure as a service model is gaining acceptance for deploying enterprise applications and efficiently managing corporate datacenters (Figure 2). IaaS is also being explored for other usage scenarios such as HPC and scientific computing. Since typical HPC hardware is expensive and underutilized, IaaS technologies can potentially bring the benefits of the cloud to this market. However, lack of reliable performance data of such applications in cloud environments as well as uncertainty in performance across vendors, environments and cloud deployments is an impediment to increased adoption.

978-0-7695-4302-4/10 $26.00 © 2010 IEEE
DOI 10.1109/CloudCom.2010.47

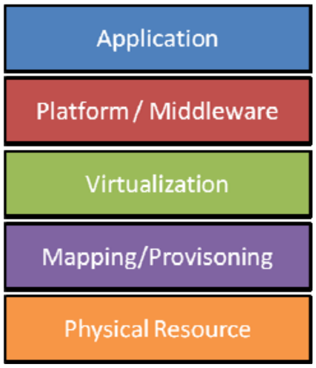

Figure 2. Application Deployment on the Cloud

III. PROVISIONING VARIATION

In cloud infrastructure environments, a cloud provider services requests from a client on shared physical resources (Figure 3). The client is typically offered the resources that can be configured according to their needs. The parameters that can be configured are typically a number of virtual machines, each of which can be individually configured for the number of virtual CPUs (vCPUs), RAM and disk space.

Since many configuration parameters have been simplified from the client's view, virtualized resources that have been provisioned to a client can have ambiguous physical characteristics. For example, when a client requests virtual machines from a cloud provider, these virtual machines can be provisioned in various physical layouts. The client receives the virtualized resources but has no knowledge of the physical ones. Identical client requests could be satisfied using different physical resource layouts. The resource contention in each physical layout could differ substantially. Hence, this mapping can potentially impact runtime performance of the client's application. We term the variation due to ambiguity in the mapping of virtual resources to physical resources in a cloud computing environment as *provisioning variation.*

Different application domains put different loads on available physical resources. Examples vary between compute, memory or I/O bound applications. The performance of an application is limited by the resources available and the contention for these resources. When applications compete for the same physical resource, serious performance degradation may occur.

We conjecture that the mapping of identical virtual resources on different physical layouts could have significant variations in physical resource contention. This could lead to performance variation of the same application on identical virtual resources due to the variation in resource contention.

A. Hardware Virtualization vs. Provisioning

Cloud provisioning systems usually make the decisions of compute resource, network and storage allocation, when it receives a new provisioning request. This allocation was shown to be NP-Hard [5], and current provisioning systems

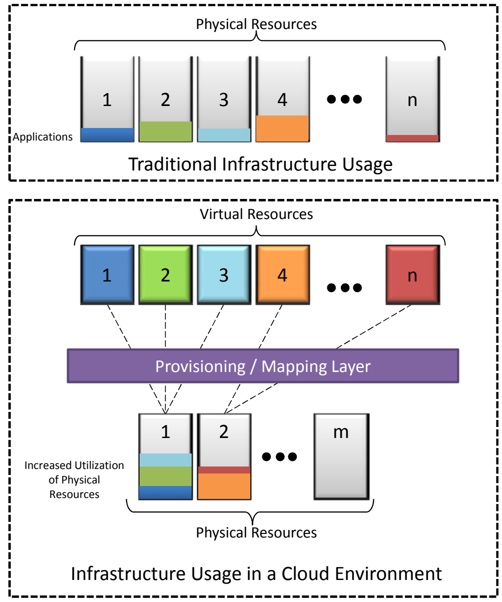

Figure 3. Provisioning on the Cloud

use heuristics to make virtual machine placement decisions. Current cloud systems factor system/CPU load, energy, as well as disk and memory capacity while making these decisions. We believe that provisioning variation stems from the variation in satisfying identical provisioning requests due to load, resources constraints and proximity of VM placement. Hardware virtualization is a factor, but the provisioning system plays a bigger role in provisioning variation.

IV. RELATED WORK

Ever since the advent of cloud-enabling technologies such as virtualization, a lot of studies [4] have been devoted to their applicability and gauging performance for various application domains as well as scientific and HPC applications. Mergen et. al. [19] discussed the trends and issues with hardware virtualization and their value in HPC environments. Some studies have reported successful porting of traditional HPC workloads onto public cloud environments [9, 12, 16]. Jaliya et. al. has also explored the applicability of cloud technologies for HPC applications in a series of studies [10, 11]. The effect of virtualization on new-generation programming models and environments like Hadoop has been explored in [13]. A comprehensive benchmark suite for Hadoop was recently developed by Huang et.al. [21]. These works have studied porting HPC applications to the cloud and evaluated the impact of virtualization on HPC apps.

Recent studies have investigated runtime performance of Service Oriented Applications and MapReduce on the Amazon EC2 public cloud. Dan et. al. [7] show that for service oriented applications, sustained performance can be achieved for a single VM instance monitored over 24 hours. However, they fail to see homogeneous performance of an application even across identical VM instances.

Schad et. al. [20] have done an extensive study on long-term performance variance of MapReduce workloads on Amazon EC2. These studies have highlighted the effect of performance variation in public clouds, but they are limited in their analysis and understanding on some of the underlying factors that are behind these performance variations. As they had worked with public clouds, they had no control over the provisioning and physical resource mapping to experiment with different VM provisioning layouts. In our work we experiment with a private cloud, which provides us full control over the provisioning system and the underlying physical hardware. This allows us to design and test various VM layouts in a controlled environment and should help in determining some of the causes of provisioning variation.

V. Experimental Setup

We will evaluate the impact of provisioning variation on performance through experimentation, using simple but different physical layouts. Figure 4 illustrates the following example; Four VMs requested by a client could be provisioned in a single blade, or could be divided among 2 or 4 blades. While a seemingly infinite number of physical resource mappings are possible with sufficiently large physical infrastructure, for the remainder of this discussion, we are concerned with precisely these three cases:

- **4Vx1B –** 4 VMs are provisioned on a single blade.
- **4Vx2B** – 4 VMs are divided evenly among 2 physical blades.
- **4Vx4B** – 4 VMs are independently assigned individual physical blades.

We have selected these cases for investigation since they potentially reflect the best and worst case scenarios for small cluster instances in a typical IaaS provisioning scenario. We do not intend to provide an exhaustive evaluation of all possible provisioning scenarios for a cloud provider; such work is beyond the scope of this paper. We mainly intend to define provisioning variation and demonstrate its existence in a private cloud environment. We have deployed and tested the scenarios discussed above on the cloud infrastructure (QLOUD) at Carnegie Mellon Qatar, which is described in the following section.

A. *The Qloud Infrastructure*

Qloud is the cloud infrastructure at Carnegie Mellon University in Qatar. It consists of an IBM Bladecenter H chassis populated with 14 blades. The hardware configuration of each of these blades is described in Table 1.

Qloud has been configured with the IBM BlueCloud software stack, which runs on a dedicated control blade. VMs are provisioned on the rest of the 13 VM host blades, all of which run the Xen 3.1.2 Hypervisor [3]. The IBM Bluecloud stack enables cloud users to request for customized VM clusters using a web portal. Tivoli Provisioning Manager 5.1.1 manages provisioning of the VMs on the VM host blades. We use vmstat [25] to monitor resource usage.

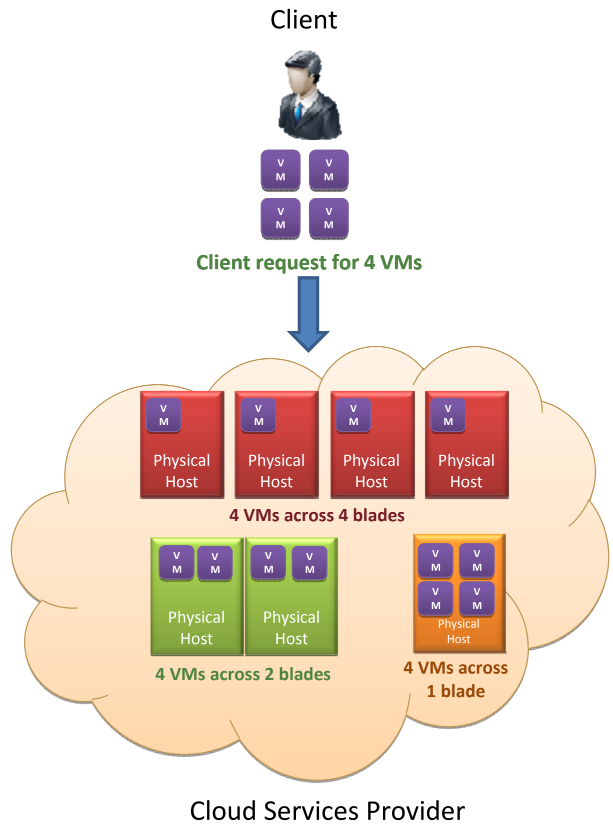

Figure 4. Ambiguity in mapping a virtual resource request to physical resources for cloud providers

TABLE I. Qloud Testbed Hardware Configuration and Virtual Machine Parameters.

Category	Configuration
Hardware	
Chassis	IBM Bladecenter H
Number of Blades	14
Processors / Blade	2x2.5 GHz Intel Xeon
	Quad Core (E5420)
RAM / Blade	8 GB RAM
Storage/Blade	2x300 GB SAS
	defined as 600 GB RAID 0
Virtualization Platform	Xen 3.1.2
Virtual Machine Configuration	
VM Parameters	1 vCPU, 1 GB RAM
	30 GB Disk
OS	Red Hat Enterprise Linux 5.2
JVM	Java SE 6, Update 18
Hadoop	Hadoop 0.20.1

B. *Applications and Benchmarks*

In order to evaluate the effect of provisioning variation on applications deployed on virtual machines, we have chosen a variety of applications and benchmarks. We broadly classify the applications into two types:

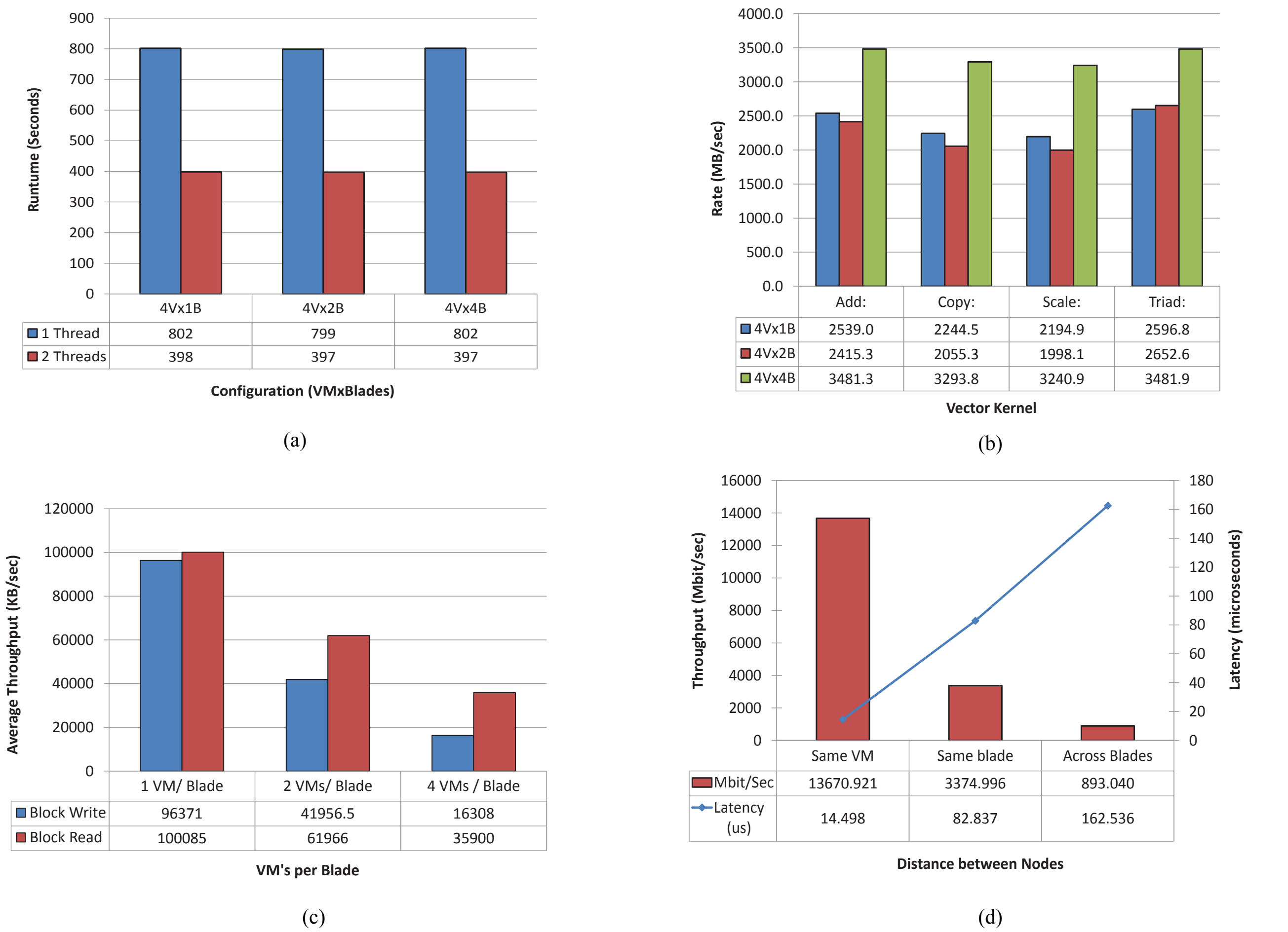

Figure 5. System benchmarks to study the effect of Provisioning Variation on the various components of VM clusters in different physical layouts: (a) Average program runtime to calculate Pi accurate to 4 million digits from the VMs using the SysTester benchmark, (b) Average memory bandwidth obtained from VMs using the STREAM benchmark, (c) Average Disk throughput for a block write/read operation for each VM using the Bonnie++ Benchmark, (d) Average network bandwidth and latency between two processes within and between VMs on a blade and across blades using the NetPerf Benchmark

- Systems Benchmarks – Stand-alone applications that are run on VMs to measure and analyze various system components such as CPU, RAM, Disk and Network.

- Hadoop applications – distributed applications developed in the MapReduce[8] paradigm and implemented in Hadoop[26].

The systems benchmarks that we use are SysTester [23], STREAM[17], Bonnie++[6] and NetPerf[15]. SysTester is a CPU stress-test benchmark that estimates Pi to varying precision. STREAM is used to measure CPU to memory bandwidth by computing simple vector kernels. Bonnie++ measures disk read/write disk throughput and Netperf is used to measure bandwidth and latency of a network using TCP and UDP protocols.

For Hadoop applications, we have chosen to evaluate runtime and resource usage of three different work-loads: Hadoop Wordcount, Sort and the DFSIO benchmark[26].

The applications are described below:

- **Hadoop Sort** - A simple MapReduce application that uses the MapReduce framework to sort data. We use the RandomWriter application included in the Hadoop 0.20 release to generate random binary input data of 64 MB to 1024 MB per host with individual record sizes varying from 10 bytes to 1000 bytes.

- **Hadoop DFSIO** - A Benchmark test included in Hadoop to calculate the total read/write parallel disk I/O bandwidth of an HDFS cluster

- **Hadoop Wordcount** - Counts the number of occurrences of each word in a set of input text files. Text data ranging from 1 GB to 8 GB of random English sentences is used as input.

VI. EXPERIMENTAL RESULTS

A. System Benchmarks

The system benchmarks are designed to measure performance of various system elements in the specific provision variation scenarios as mentioned in Section 5. To achieve this, we execute the benchmarks simultaneously on all the VMs of the cluster, unless explicitly mentioned otherwise (Figure 5).

Figure 5(a) shows the average runtime performance of SysTester in estimating Pi to 4 million digits using 1 or 2 CPU threads. It can be seen that having 2 or 4 VMs on the same blade (4Vx2B or 4Vx4B) does not affect the performance of this CPU-bound benchmark.

Practical CPU to memory bandwidth is measured using STREAM as illustrated in Figure 5(b). The figure illustrates average bandwidth in MB/sec for various vector kernels. It can seen than there is roughly a 25 % drop in memory bandwidth when 2 or 4 VMs are running the benchmark simultaneously on one blade.

Figure 5(c) illustrates the average disk throughput in KB/sec as measured by the Bonnie++ benchmark. With 2 or 4 VMs accessing the local disk simultaneously, we see a 40 to 60% drop in average throughput for read and about 60 to 80% for write operations respectively. Such a significant drop in I/O performance has the potential to impact the performance of I/O-bound applications.

Finally, Figure 5(d) shows the relative differences in Bandwidth and Latency of TCP transmission using the NetPerf benchmark. Two processes running on the same VM is the fastest being about half of the total memory bandwidth as measured by STREAM (Figure 5(b)). However, between VMs running on the same blade, there is an 88% drop in throughput. Finally, the actual bandwidth between two VMs running on different blades is roughly 89% of the peak rated bandwidth of the gigabit link between them. Clearly, network-bound applications can potentially benefit from I/O and networking enhancements within the hypervisor and OS stack when VMs are placed on the same blade. This has been the subject of a lot of research recently [18, 24].

B. Hadoop Sort

We run the standard Hadoop Sort benchmark using the input configurations listed in Table 2. We begin with runtime comparison of Hadoop Sort, as shown in Figure 6. It can be clearly seen that sorting is much faster when 4 VMs are placed on individual physical nodes (4Vx4B) - by up to a factor of 5. We have also run this workload on VM clusters with 1.5 and 2 GB of memory each and the relative speedup of the VMs layouts are presented in Table 3.

For larger input sizes (starting from 2GB input), the VM starts to swap large amounts of memory. The 4Vx1B configuration suffers from decreased I/O performance (as discussed in Section 6A), as well as diminished HDFS read/write performance as indicated by the DFSIO benchmark, which will be discussed in Section 6C.

The aggregate I/O performance of this layout has a cascading effect on the ability of the application to read and write large amounts of data during the various phases of the MapReduce computation and adversely impacts the speed at which data can be swapped to/from disk (as shown in Table 4). Hadoop Sort is indicative of a large-scale I/O intensive application, and the performance of such applications can potentially be adversely impacted due to provisioning variation.

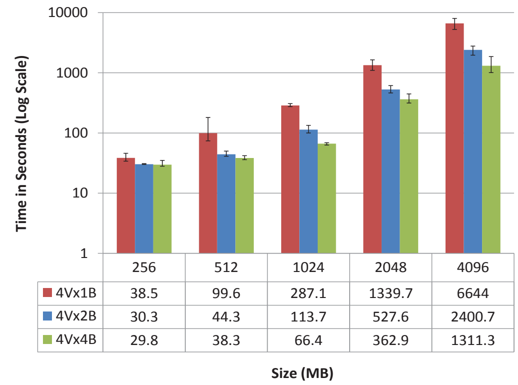

Figure 6. Runtime performance of Hadoop Sort on input sizes ranging from 256 MB to 4 GB for various VM configurations with 1 GB of RAM per VM. The error bars indicate the standard deviation over all iterations.

TABLE II. RUNTIME CONFIGURATION WITH MAPREDUCE STATISTICS FOR VARIOUS INPUT SIZES FOR HADOOP SORT

Input Size (MB)	No. of Maps	No. of Reduces	No. of Input Records
256	4	7	25,558
512	8	7	51,226
1024	16	7	1,02,221
2048	32	7	2,04,565
4096	64	7	4,08,712

TABLE III. RELATIVE SPEEDUP OF THE HADOOP SORT BENCHMARK FOR VARIOUS VM CONFIGURATIONS WITH VARYING AMOUNTS OF MEMORY PER VM.

Physical Configuration	Relative Speedup		
	1 GB	*1.5 GB*	*2 GB*
4Vx1B	1	1	–
4Vx2B	2.7	2.16	1
4Vx4B	5.06	5.75	1.77

TABLE IV. VIRTUAL MEMORY USAGE STATISTICS FROM THE MASTERNODE OF EACH VM CLUSTER WHILE RUNNING HADOOP SORT FOR AN INPUT SIZE OF 8 GB.

Physical Configuration	Growth in Swap Size(GB)	Peak Swapping Rate (MB/sec)
4Vx1B	0.82	3.6
4Vx2B	1.17	9.8
4Vx4B	1.21	14.5

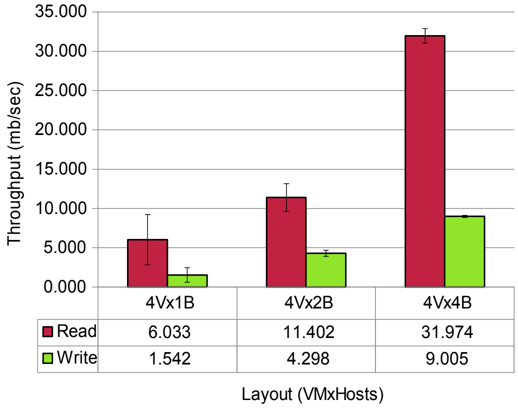

Figure 7. DFSIO Benchmark Read and Write throughput for each of the VM configurations averaged over 10 iterations. Error bars indicate standard deviation of the measurements

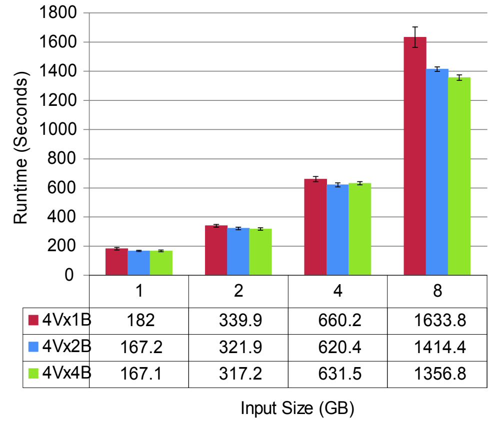

Figure 8. Average runtime of Hadoop Wordcount on input sizes ranging from 1 GB to 8 GB various VM configurations over 10 iterations. Error bars indicate standard deviation over all iterations.

C. *DFSIO Benchmark*

The DFSIO benchmark is a synthetic I/O performance benchmark for the HDFS filesystem packaged along with the Hadoop distribution. It employs a MapReduce application to measure the combined I/O throughput of all the nodes in an HDFS cluster. Figure 7 shows the combined I/O throughput of the HDFS nodes on the clusters, the slowdown from 4Vx4B to 4Vx1B being roughly 5x, which correlates with the performance seen in Hadoop sort (Section 6B).

D. *Hadoop Wordcount*

The effect of provisioning variation is more subtle for the Wordcount benchmark. Figure 8 shows the effect of provisioning variation for input text files ranging from 1 GB to 8GB on the VM configurations considered. For the largest input size of 8GB, the 4Vx4B layout is on average around 20% faster than the 4Vx1B layout. Wordcount is also a CPU-intensive application [21], and the computation to communication ratio is much more than Sort, a purely I/O bound application.

VII. Analysis

A. *Systems Benchmarks*

The systems benchmarks show that the effect of provisioning variation is most pronounced for disk and network-bound applications. With a slowdown varying from 40% to 80% of disk throughput, it can affect disk-bound applications significantly. Furthermore, since the VMs run full-fledged operating systems, the reduction in disk bandwidth can affect virtual memory performance, which was reflected with Hadoop sort in Table 4.

B. *Hadoop Benchmarks*

Hadoop's computational architecture is inherently focused on large-scale data processing, especially for "big data" that spans multiple physical hard disks. It makes terabyte and petabyte scale computation possible by leveraging the power of parallel file systems such as HDFS. However, when two or more VMs are competing for and not complementing average disk throughput (by virtue of two or more VMs sharing a physical disk), the performance degradation is substantial, as revealed in I/O sensitive Hadoop applications such as Hadoop Sort and DFSIO. When sorting larger instances of input data (such as 2 GB or 4 GB of input data), the additional burden of virtual memory swapping comes into play, thus a cascading effect of thrashing combined with decreased I/O throughput from disk even leads to stalling of Hadoop Tasks on the 4Vx1B configuration.

Runtime data from Sort and DFSIO show a high degree of correlation and show that the decreased I/O performance of co-located VMs on a single blade adversely affects the runtime performance of an I/O bound Hadoop application. The effect of provisioning variation on the Hadoop Wordcount appears to be less drastic, but is significant nonetheless.

VIII. Summary and Conclusions

We have studied the potential effect of provisioning variation on application performance in cloud services. We have demonstrated the existence of provisioning variation for VM clusters provisioned using the IaaS cloud computing service model. Using a set of sequential benchmarks, we have shown that, among others, disk I/O throughput is particularly susceptible to degradation owing to multiple VMs on a physical host competing to share a physical disk.

We have also demonstrated the effect of provisioning variation on MapReduce applications such as Sort and Wordcount and proved that there is significant degradation in performance for the Sort application, up to 5x, which is corroborated by the DFSIO benchmark.

We conclude that provisioning variation is real and its impact could significantly vary the performance of the same application instance on identically provisioned virtualized resources. This could have serious implications when thinking about developing provisioning and scheduling techniques that offer better QoS for the cloud user.

IX. FUTURE WORK

This work presents preliminary results on the true impact of provisioning variation, focusing mainly on MapReduce applications. We hope to continue our investigation using more applications, particularly HPC and scientific kernels in the future and evaluate the effects of provisioning variation in much larger VM cluster deployments and possibly public cloud services. We also hope to incorporate profiling techniques in order to understand the behavior of a distributed application. This can help in influencing provisioning decisions and improve performance of applications on the cloud.

X. ACKNOWLEDGEMENTS

We would like to thank Jim Gargani for helping with the system setup of our cloud experiments. We would also like to acknowledge IBM for infrastructure and system support as well as Carnegie Mellon Qatar and the Qatar Foundation for supporting our research efforts.

REFERENCES

[1] Amazon Web Services, Elastic Compute Cloud, http://aws.amazon.com/ec2/

[2] Amazon Web Services, Elastic Mapreduce, http://aws.amazon.com/elasticmapreduce/.

[3] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A.: Xen and the Art of Virtualization: 'Book Xen and the Art of Virtualization' (ACM, 2003, edn.), pp. 164-177

[4] Cherkasova, L., Gupta, D., and Vahdat, A.: Comparison of the Three Cpu Schedulers in Xen, Performance Evaluation Review, 2007, 35, (2), pp. 42

[5] , Resource Management in Multi-* Clusters : Cloud Provisioning, http://www.mosharaf.com/wp-content/uploads/mosharaf-cpp-report-spring10.pdf

[6] Coker, R, Bonnie++, http://www.coker.com.au/bonnie+

[7] Dan, A., Gittler, F., Toumani, F., Dejun, J., Pierre, G., and Chi, C.-H.: Ec2 Performance Analysis for Resource Provisioning of Service-Oriented Applications: 'Service-Oriented Computing. Icsoc/Servicewave 2009 Workshops' (Springer Berlin / Heidelberg, 2010), pp. 197-207

[8] Dean, J., and Ghemawat, S.: Mapreduce: Simplified Data Processing on Large Clusters, Communications of the ACM, 2008, 51, (1), pp. 107-113

[9] Deelman, E., Singh, G., Livny, M., Berriman, B., and Good, J.: The Cost of Doing Science on the Cloud: The Montage Example: 'Book The Cost of Doing Science on the Cloud: The Montage Example' (IEEE, 2008, edn.), pp. 1-12

[10] Ekanayake, J.: Architecture and Performance of Runtime Environments for Data Intensive Scalable Computing: 'Book Architecture and Performance of Runtime Environments for Data Intensive Scalable Computing' (2009, edn.), pp.

[11] Ekanayake, J., Qiu, X., Gunarathne, T., Beason, S., and Fox, G.: High-Performance Parallel Computing with Cloud and Cloud Technologies, Cloud Computing and Software Services: Theory and Techniques, 2009, pp. 275

[12] Evangelinos, C., and Hill, C.: Cloud Computing for Parallel Scientific Hpc Applications: Feasibility of Running Coupled Atmosphere-Ocean Climate Models on Amazon's Ec2, Cloud Computing and Applications, 2008, 2, (2.40), pp. 2.34

[13] Ibrahim, S., Jin, H., Lu, L., Qi, L., Wu, S., and Shi, X.: Evaluating Mapreduce on Virtual Machines: The Hadoop Case, Proceedings of the 1st International Conference on Cloud Computing, Beijing, China2009

[14] Vsphere Virtualization Platform, http://www.vmware.com/products/vsphere/

[15] Jones, R.: Netperf Benchmark, http://www.netperf.org/netperf/NetperfPage.html

[16] Keahey, K., Figueiredo, R., Fortes, J., Freeman, T., and Tsugawa, M.: Science Clouds: Early Experiences in Cloud Computing for Scientific Applications: 'Book Science Clouds: Early Experiences in Cloud Computing for Scientific Applications' (2008, edn.), pp.

[17] McCalpin, J.: A Survey of Memory Bandwidth and Machine Balance in Current High Performance Computers, IEEE TCCA Newsletter, 1995, pp. 19–25

[18] Menon, A., Cox, A.L., and Zwaenepoel, W.: Optimizing Network Virtualization in Xen, Proceedings of the annual conference on USENIX '06 Annual Technical Conference, Boston, MA2006

[19] Mergen, M.F., Uhlig, V., Krieger, O., and Xenidis, J.: Virtualization for High-Performance Computing, SIGOPS Oper. Syst. Rev., 2006, 40, (2), pp. 8-11

[20] Schad, J., Dittrich, J., and Quiané-Ruiz, J.: Runtime Measurements in the Cloud: Observing, Analyzing, and Reducing Variance, Proceedings of the VLDB Endowment, 2010, 3, (1)

[21] Shengsheng, H., Jie, H., Jinquan, D., Tao, X., and Bo, H.: The Hibench Benchmark Suite: Characterization of the Mapreduce-Based Data Analysis: 'Book The Hibench Benchmark Suite: Characterization of the Mapreduce-Based Data Analysis' (2010, edn.), pp. 41-51

[22] Strategic Consulting Incorporated, Abstraction: The Key to Understanding Cloud Computing, http://www.strategi-consulting.com/OurIdeas/Blog/tabid/108/EntryId/3/Abstraction-The-key-understanding-cloud-computing.aspx

[23] Tsatiris, L., Anastasiadis, P. and Kaniadakis, M. , System Stability Tester, http://systester.sourceforge.net/

[24] Wang, J., Wright, K.-L., and Gopalan, K.: Xenloop: A Transparent High Performance Inter-Vm Network Loopback, Proceedings of the 17th international symposium on High performance distributed computing, Boston, MA, USA2008

[25] Ware, H., and Frdrick, F.: Linux Man Page: Vmstat (8): 'Book Linux Man Page: Vmstat (8)' (1994, edn.), pp.

[26] White, T.: Hadoop: The Definitive Guide (O'Reilly Media, 2009. 2009)