Understanding performance interference in multi-tenant cloud databases and web applications

Miguel G. Xavier*, Kassiano J. Matteussi, Fabian Lorenzo and Cesar A. F. De Rose*
*Pontifical Catholic University of Rio Grande do Sul (PUCRS)
Faculty of Informatics, Porto Alegre, Brazil

***Abstract*—The number of e-commerce customers and database services in cloud computing platforms has grown increasingly, leading providers to adopt resource-sharing solutions to meet growing demand for infrastructure resources, such as processing and storage. Consolidating database applications has become arguably a de-facto solution to support a large number of customers/tenants at low infrastructure costs. However, the friction generated in shared hardwares (resource contention) is converted to performance interference, which is felt by tenants' database applications running on upper layers (VMs). Hence, there is a real concern on how to manage and prevent multi-tenant cloud databases from performance interferences sourced by either resource contention or isolation flaws. In this paper, we analyzed the performance interference tolerated by multi-tenant e-commerce cloud databases in resource-sharing infrastructures. We claimed that multiple-different workloads (e.g. memory-/CPU-intensive, and e-commerce applications) might be consolidated with database systems to minimize performance interference and increase resource-efficiency.**

***Keywords*—Cloud Computing; Database Systems; Virtualization; Performance Interference; Disk-Intensive Workloads**

I. Introduction

The world has witnessed a steady increase of data on the Internet. This vast amount of data has generated heterogeneous data processing ecosystems to extract useful information. As a result, immense data centers have been built to accommodate thousands of database systems and provide services in a pay-as-you-go model, as in cloud computing. A greater challenge involves accommodating multiple IT-related infrastructure resources in a flexible and cost-effective way. Virtualization is the key enabling technology, which facilitates the implementation of modern Internet services in many contexts such as formal education, shopping, social networks and so on [1]. It is the technology beneath cloud computing platforms like Google [2], Azure [3], Amazon [4] and other [5].

Web applications, such as e-commerce, are composed of many services distributed in clouds, including web servers, database systems, storages, network substrates, etc. Hence, cloud platforms require scalable infrastructures to provision them with SLA guarantee at a low cost. Most customers pay for a minimum-costly infrastructure necessary to run their services, and they are frequently consolidated and shared among multiple tenants. This might cause uncontrolled performance variations due to the competition for the same physical resource, or also known as resource contention. For instance, when a database system is overloaded, all tenants running on the same shared machine will be likely affected and vice versa. The resource contention generated in the shared hardware is converted to interference, which is felt by the tenants' applications. To illustrate, the dispersion in Figure 1 presents a disturbed scenario in which two disk-intensive applications write to/read from a single disk while the bandwidth is not enough to supply applications' demands, making the performance fluctuate.

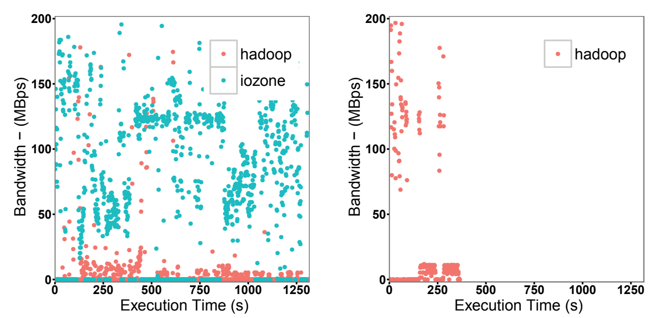

Fig. 1. Performance interference between two co-located applications due to the disk contention

There have been studies evaluating resource contention and the impact on applications' performance [6], [7], [8], [1], [9], but none of them explore cloud database systems under stress conditions sourced from e-commerce applications. Hence, in this paper we present a performance interference analysis of multi-tenant cloud databases using e-commerce applications as workload and show the database sensitivity as that hardware resources are fully consumed by other co-located applications. Our results are useful for virtual machine scheduling decisions in the sense of exhibit a workload combination in which the database performance is minimally affected. And also which resource most affect the performance of multi-tenant e-commerce cloud database when under high load conditions.

The rest of this paper is organized as follows: Section II briefly describes the concepts intrinsics to disk I/O contention; web applications and database systems. Section III outlines performance isolation metrics and how they can be used to evaluate multi-tenant systems like databases. Section IV presents indeed the evaluations. Related work are presented in Section V and conclusion in Section VI.

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

II. Background

Traditional distributed architectures lack the flexibility required to maintain the processing speed and the scalability necessary for dynamic workloads. To improve the management of resources and workloads, data centers have used virtualization to deliver scalability, rapid provisioning, efficiency, flexibility, and the speed necessary to process and store their data and applications [10]. Accommodating multiple IT-related resources with high processing levels requires a virtualization layer with subsystems that emulate and provide VMs as close as possible to the physical substrate. By doing so it is possible to reduce operational costs, idle resources, power consumption, and physical space to be more resource efficient.

Virtualization is the key enabling technology of the cloud computing model [11]. Users have been encouraged to use on-line services in clouds on an unprecedented scale leading data centers to handle heavy loads and manageability issues. This scenario typically jeopardizes applications performance when dozens or even thousands of users concurrently access multi-tenant applications that share the same resource, such as CPU, memory, disk, and network. This disturbing scenario is called resource contention. Due to the significant competition for resources, application performance tends to fluctuate, compromising the systems stability and user satisfaction.

A. Disk I/O Contention and Resource Allocation

Uncontrolled access to shared resources can cause performance variations that lead applications to fail or run unsteadily. The friction generated by the competition to access RAM, disk storage, cache or internal busses is called resource contention. Many efforts have been made to alleviate the contention in the operating system level that range from better scheduling techniques in multi-core architecture [12] to dynamic addressing mapping to minimize memory contention [13]. The steady growth of the Database-as-a-Service (DaaS) [11] cloud computing model brought a concern about I/O contention and the impact it might cause in environments where performance is crucial and SLA cannot be violated. Disk I/O contention occurs when multiple virtual instances compete for a disk bandwidth portion in a scenario where the demand is higher than the available resource.

Operating system level I/O schedulers, such as CFQ, deadline and noop, detect resource utilization bottlenecks and attempt to divide block devices by reordering/prioritizing tasks in a fairly-balanced manner. As a result, the overhead is distributed equally across the multi-tenant instances, but it does not prevent their performance from varying unpredictably since the schedulers are unable to predict and make decisions based on workload characteristics.

On the other hand, performance interference may also be sourced from isolation issues in the virtualization layer, which occurs when an application exceeds the amount of allocated resources. Even though a virtual instance receives a limited portion of resources, it does not prevent the resource utilization from leakage due to isolation flaws [14], [15]. Hence, performance interference may be sourced from either resource contention or performance isolation issues. Datacenter administrators exaggerate the amount of allocated resources to sidestep performance interference, leading the datacenter to underutilized and no longer resource efficient.

B. Web Applications and Database Systems

The Internet industry has continuously grown in recent years due to the popularity of cloud computing, which uses virtualization technology as middleware to provide on-demand resources such as servers, processing units, storage, and networking to supply a set of application requirements. Web and database servers are two strongly connected entities and in most cases are consolidated in the same physical machine. Depending on the characteristic of the application, it may consume more or less resources and the types of resources may vary during execution. For example, image processing applications essentially read data from disk to memory and execute in-memory processing most of the time, which increases the memory access rate. Therefore, these applications should not suffer significant performance interference from a side-by-side, co-located database server since they differ from each other in terms of resource consumption. Disk resources (e.g. DAS, NAS, local disk) for web applications are essential because they store the application data, statistics, registers of sales, purchases, etc. For this reason, understanding the performance interference between web applications and database systems is an important task to maintain a more balanced environment and guarantee the best use of resources.

Preserving database performance in virtualized data centers is essential for customer satisfaction. When database systems started to be migrated from the traditional hosting methodology to clouds, they suffered performance overheads due to the nature of the virtualization layers. Yet, this was offset by better use of resources and lower infrastructure and maintenance costs. However, when the number of per-machine co-located database systems increase, the disk I/O rate also grows. This leads the machine to a disk-intensive usage scenario, in which the rates must be controlled to ensure fair disk resource distribution and prevent applications from interfering with each other. Database systems are disk-intensive workloads and require disk storage and I/O bandwidth to perform their operations gracefully. This is the case of Oracle [16] and MySQL [17], which handle the processing and storage of a vast amount of data.

III. Performance Interference Analysis Technique

Uncontrolled competition for shared computing resources of co-located applications is denoted as resource contention. This situation creates unexpected performance variations that affect multi-tenant workloads. The most common way to quantify performance interference between two virtualized workloads consists of running them concurrently within different consolidated VMs. Figure 2 illustrates this scenario, using a commercial DBMS as an example.

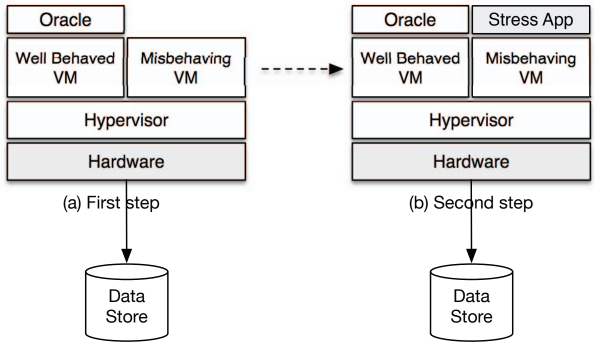

Fig. 2. Workflow for quantifying performance interference.

As observed, the total amount of physical resources is partitioned evenly among the VMs (two VMs per machine). The workflow consists basically of two steps. In the first step (a), the DBMS runs alone in one of the two VMs, while the other remains idle, meaning that no VM has been disturbed. The DBMS's metrics (IOPS, latency) are collected and stored to be used in the future, at the end of the workflow. In the second step (b), DBMS runs into one of the VMs (well behaved), but now it is running side-by-side with a disruptive VM (misbehaving). The metrics are collected again, and the performance degradation is quantified by the difference of the metrics collected from the two steps.

The natural tendency of using virtualization makes the implementation of modern Internet services easier in many contexts, such as formal education (distance learning), shopping (electronic commerce), friendship development (social networking) [1], etc. However, performance overheads are more notable in virtualized environments [6]. This makes the performance evaluation of virtualized workloads a very complex task.

IV. Performance Interference Evaluation of Multi-tenant Cloud Database

Experiments were performed to analyze how performance was affected in a testbed with consolidated database systems and web servers. Firstly, the performance of a database system was analyzed while running it concurrently with very disruptive stress apps within different VMs (well behaved/misbehaved). Each resource (CPU, disk, and memory) was stressed individually to identify per-resource performance degradation. Subsequently, an e-commerce application was tested to produce a scenario closer to a real-world environment.

The testbed consisted of one Dell PowerEdge R610 connected to a distributed storage with 1TB of space. The VMWare's ESXi [18] (version 6.0) was installed on it. A commercial DBMS was installed and configured in a VM with Ubuntu Server 14.04. We chose the ORION tool [19] and the Workload Replay tool [20] to put stress on the database to evaluate its performance under different conditions. ORION runs tests using different I/O loads to measure performance metrics such as Mbps, IOPS, and I/O latency. ORION simulates many sequential tests. Each test (referred to as data point) represents a specific mix of concurrent small and large I/O loads for a fixed duration of time. On the other hand, the Workload Replay tool runs a workload captured from a production environment.

A. CPU Analysis

The Stress-ng tool [21] was used to put stress on the CPU and reach high load levels. It is a typical CPU bomb application based on Fast Fourier Transform (FFT) and Discrete Fourier Transform (DFT). It was configured to consume all of the processing capacity (100%) of the misbehaving VM. The results can be seen in Figure 3.

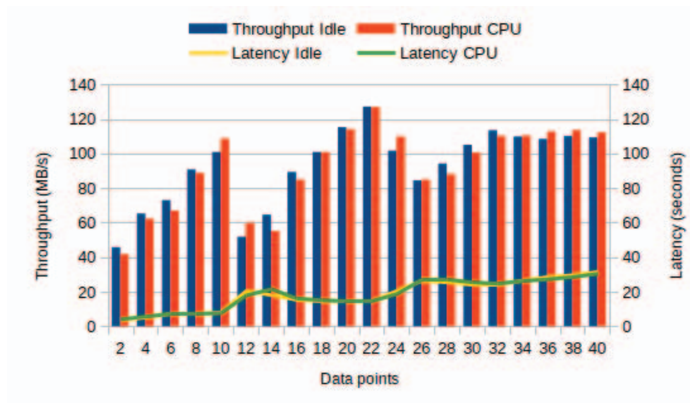

Fig. 3. Performance degradation in MB/s and latency using Stress-ng.

As observed, the performance fluctuated before reaching the maximum throughput induced by the characteristics of the ORION transactions. We believe that latency affected the progress of the test. If the latency increases, the throughput and consequently the number of IOPS decreases. In this case, the performance interference suffered by the DBMS was only 0.2%.

B. Memory Analysis

Stressapptest [22] was used to put stress on the memory subsystem. Stressapptest is an application that simulates random traffic to memory with the intent of creating a realistic memory stress scenario. It was installed and executed into the side-by-side misbehaving VM, while the DBMS was dealing with many simultaneous transactions.

Figure 4 shows that DBMS's throughput under stress remained very close to the idle experiment for small and medium data point sizes. Little disturbance (1.8%) was observed for large data points. In the virtualization layer, the virtual machine monitor (hypervisor) imposes overhead on the memory paging process [23] due to the cost of maintaining tables and synchronizing tasks between VMs and hosts. Thus, this behavior can be explained by the latency imposed by the hypervisor to handle greater loads produced by the Stressapptest.

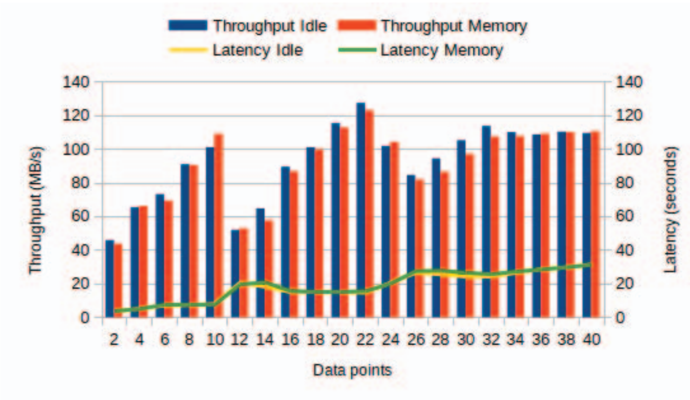

Fig. 4. Performance degradation in MB/s and latency using Stressapptest.

C. Disk I/O Analysis

The Iozone [24] tool was used to put stress on the I/O subsystem. It is a file system benchmark that runs several file operations simultaneously. We ran it using six OLTP-based options, creating synchronous writes and reads to the data store. It produced a lot of I/O intensive tests. As a result, a very disruptive behavior emerged while running multiple and parallel I/O operations. The results are plotted in Figure 5.

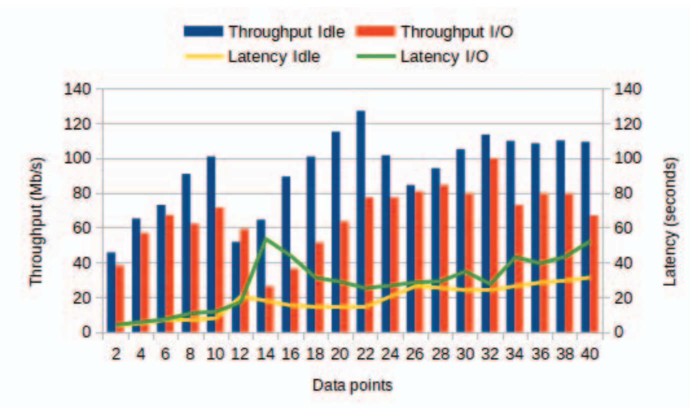

Fig. 5. Performance degradation in MB/s and latency using Iozone.

We believe this highly disruptive behavior is due to the disk contention when more than one VM competes for storage, making performance vary. The testbed did not restrict I/O resources per VM. Rather, the virtual machine monitor handled translating instructions from the upper (VM) to the lower layers (Disk), and the I/O scheduling was done in this process. We plan to do further tests by restricting the I/O bandwidth per VM and analyzing the database behavior in a more controlled scenario.

The experiment presented 28% performance degradation. This interference directly impacted the number of IOPS performed, as we can observe in Figure 6. These results show that an efficient strategy for consolidation is necessary when running several VMs that perform many operations on the same disk.

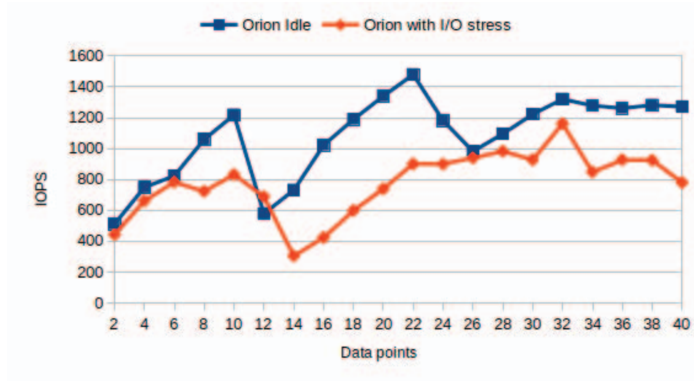

Fig. 6. Performance degradation in IOPS.

D. Consolidating Web Application and Database System

The benchmark RUBiS [25] was configured to simulate an E-commerce web application and produce an experiment closer to a real-world web application. RUBiS performs many transitions between a client and a web server. Yet, RUBiS uses MySQL, which is a database system broadly used in Internet applications. The server was configured into the misbehaving VM in the testbed. Figure 7 depicts the results.

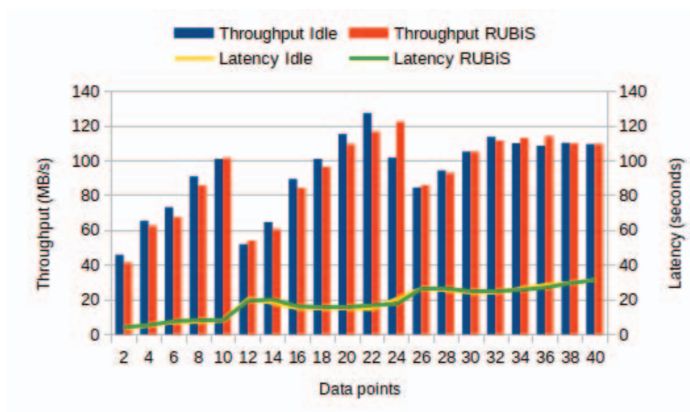

Fig. 7. Sample I/O Load Levels Versus Latency for RUBiS application.

The experiment presented 0.8% performance degradation in the database. This behavior indicates that the consolidation of web applications and database systems may be a good combination to maintain satisfactory performance levels. Hence, we observed that RUBiS did not present the same performance degradation compared to the disk I/O experiment. This behavior occurred due to the configurations imposed by the benchmark, where many mixed transactions were performed.

E. Real World Scenario

We used the Database Replay tool to better understand how real database applications are impacted when running in our proposed setting. The workload was captured from a database cluster environment with two instances running DBMSs with 20 terabyte of data. A full restore was done using the backup provided and the same setting was recreated. About 8.3%

divergence was found during the workload replay. For testing purposes, we replayed only 30% of the captured workload and compared the times the procedure took under on different stress scenarios versus the time it took to complete the same percentage without stress (idle).

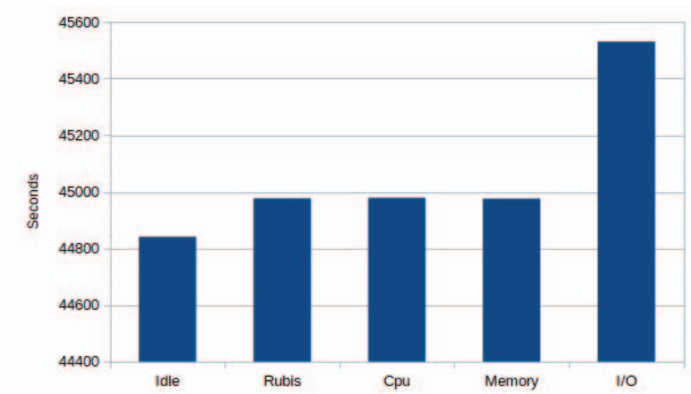

Fig. 8. Workload Replay test set.

We can observe in Figure 8 that the experiment presented roughly 0.3% performance degradation in memory, CPU, and RUBiS. The database suffered about 7.6% performance degradation when stressed by a multi-tenant I/O workload. We argue that behavior is a good indicator that databases can be consolidated with web servers without much impact on performance, as they consume different hardware resources even when under high load conditions.

V. Related Work

Cloud computing enables highly scalable services to be easily consumed by applications within different contexts (e.g., education, shopping, social networks and other [1]. In this context, these applications have dynamic workloads that needs be managed to avoid resource contention and also to minimize interference problems in virtualized environments [6]. In this way, several research are carried out to understand how to measure the impact of interference over performance using virtualized scenarios with mixed workloads [6], [7], [8]. However, only a small number of works have studied the impact of web applications on multi-tenant cloud databases.

Cloud providers promise to deliver on-demand resources and SLA guarantee—resource utilization boundaries hired by the customers. In this way, e-commerce systems must guarantee the response time of all operations performed. It is a complex task because web applications have dynamic workloads that perform an exponential number of disk/network operations/requests, leading to the resource contention and resulting in SLAs violations. The author [9] presents a prototype system to help to guarantee SLAs for cloud providers. The system was proposed to monitor the response time requests made by web applications, gathers resources statistics and then a heuristic is utilized to identify bottlenecks. If identified, the system provides exactly the resource slice needed for the application, minimizing the interference, guaranteed the response time and SLA contracted. The results show that is possible detect bottlenecks and perform dynamic resource allocations to guarantee SLAs of the Web application. However, the work not presented results about how to manage disk-intensive workloads and its impact on web applications.

Recently, many industries have been interested in exchange their hardware infrastructure and software environment for virtualized datacenters on the cloud model. It helps to save costs, allows high availability and scalability, improves fault tolerance, reduces power consumption and so on. Moreover, the cloud allows resource sharing between applications and lets users to pay only for utilized services rather than to make new contracts, on a model called Pay-As-You-Go (PAYG). The main problem involves the mixed workloads processed over virtual environments that commonly not know the processing of each other. It leads to the resource interference, isolation failures, and low application performance.

Workloads with the high demand of resource usage, especially I/O can impair performance in shared environments. For this reason, it is important to predict the performance of applications consolidated in virtualized environments. Thus, the efficient configuration of resources and the placement of workloads can minimize the loss of performance [7]. In this context, the work [6] proposed an analytical model based on Queuing Network (QN) to evaluate the performance of virtualized applications collocated. The results present the effectiveness of the proposed model in capturing the performance metrics of virtualized environment on only 12% of deviation between the model and experimental results, but not present results about the interference involving collocated workloads.

Clouds can support heterogeneous systems such as scientific applications, Database Management Systems (DBMS), data storage, multimedia streaming and so on. Generally, these applications are Disk-Intensive by nature and have different kinds of I/O workload (e.g., random or sequential read/write requests) ranging in terms of response time and throughput [26], [27], [28]. For this reason, the relation between application performance and workload consolidation becomes highly relevant in shared environments that may result in unpredictable performance [29].

VI. Conclusion

In our effort to better understand the performance effects in consolidating workloads in cloud datacenters, we claim that e-commerce cloud databases are more sensitive to disks, than other resources. This behavior is expected given the nature of the database operations. However, we showed up that other resources when highly consumed induce minimally the performance of the database. In summary, the results are presented in Table I.

TABLE I. Performance degradation of the DBMS while consolidated with different workloads

Memory	CPU	RUBiS	I/O
1.8%	0.2%	0.8%	28%

The performance degradation in the worst case (28%) was realized during the disk stress test, which reflected in a great reduction in the number of operations carried out by the DBMS. Neither RUBiS nor the database suffers significant performance variations while they were running side-by-side on the multi-tenant machine. This suggests a good opportunity for consolidating web servers and database systems taking into account an overhead of 0.8%. As presented in Table II, our results using real-world data/traffic showed 7.6% performance degradation during the disk contention analysis.

TABLE II. PERFORMANCE DEGRADATION OF THE REAL WORKLOAD WHILE CONSOLIDATED WITH DIFFERENT WORKLOADS

Memory	CPU	RUBiS	I/O
0.26%	0.28%	0.3%	7.6%

Unlike the analysis using synthetic data, these tests are not much intrusive to the DBMS. It is because the real-world e-commerce application performed less database transactions than the Orion benchmark. It means that the synthetic results represent the performance degradation in the worst case.

Many organizations are looking for opportunities to reduce costs and simplify their infrastructures. With the natural tendency of migrating database systems to cloud platforms, understanding the performance impact that an application tolerate is of paramount importance to ensure SLA and improve resource efficiency.

VII. ACKNOWLEDGMENT

This work was supported by a multinational computer technology company. All experiments presented in this paper were carried out using a infrastructure provided by the company's infrastructure team.

REFERENCES

[1] E. Overby, "Process virtualization theory and the impact of information technology," *Organization science*, vol. 19, no. 2, pp. 277–291, 2008.

[2] Google cloud platform. [Online]. Available: https://cloud.google.com/

[3] Azure, lastchecked = 2016, url = https://azure.microsoft.com.

[4] Amazon. [Online]. Available: https://aws.amazon.com

[5] X. Wen, G. Gu, Q. Li, Y. Gao, and X. Zhang, "Comparison of open-source cloud management platforms: Openstack and opennebula," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*. IEEE, 2012, pp. 2457–2461.

[6] K. RahimiZadeh, M. AnaLoui, P. Kabiri, and B. Javadi, "Performance modeling and analysis of virtualized multi-tier applications under dynamic workloads," *Journal of Network and Computer Applications*, 2015.

[7] X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, C. Pu, and Y. Cao, "Who is your neighbor: Net i/o performance interference in virtualized clouds," *Services Computing, IEEE Transactions on*, vol. 6, no. 3, pp. 314–329, 2013.

[8] Z. Gong and X. Gu, "Pac: Pattern-driven application consolidation for efficient cloud computing," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 24–33.

[9] W. Iqbal, M. N. Dailey, D. Carrera, and P. Janecek, "Adaptive resource provisioning for read intensive multi-tier applications in the cloud," *Future Generation Computer Systems*, vol. 27, no. 6, pp. 871–879, 2011.

[10] F. Ricca and P. Tonella, "Analysis and testing of web applications," *ICSE '01 Proceedings of the 23rd International Conference on Software Engineering*, pp. 25–34, 2001.

[11] P. Mell and T. Grance, "The nist definition of cloud computing," 2011.

[12] S. Zhuravlev, S. Blagodurov, and A. Fedorova, "Addressing shared resource contention in multicore processors via scheduling," in *ACM SIGARCH Computer Architecture News*, vol. 38, no. 1. ACM, 2010, pp. 129–142.

[13] K. H. Potter, "Dynamic addressing mapping to eliminate memory resource contention in a symmetric multiprocessor system," Jan. 7 2003, uS Patent 6,505,269.

[14] M. G. Xavier, M. V. Neves, F. D. Rossi, T. C. Ferreto, T. Lange, and C. A. De Rose, "Performance evaluation of container-based virtualization for high performance computing environments," in *Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on*. IEEE, 2013, pp. 233–240.

[15] M. G. Xavier, M. V. Neves, and C. A. F. D. Rose, "A performance comparison of container-based virtualization systems for mapreduce clusters," in *Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on*. IEEE, 2014, pp. 299–306.

[16] Oracle, *Oracle Database 12c plugs you into the cloud, Consolidation Best Practices: Oracle Database 12c plugs you into the cloud*. Oracle, 2013.

[17] MySQL. [Online]. Available: www.mysql.com

[18] Vmware's vsphere bare-metal virtualization. [Online]. Available: http://www.vmware.com/br/products/esxi-and-esx/overview

[19] Orion calibration tool (oracle i/o numbers). [Online]. Available: http://docs.oracle.com/cd/E11882_01/server.112/e16638/iodesign.htm#PFGRF95222

[20] Oracle workload replay. [Online]. Available: http://www.oracle.com/technetwork/articles/sql/11g-replay-099279.html

[21] Stress-ng. [Online]. Available: http://kernel.ubuntu.com/~cking/stress-ng/

[22] Stressapptest benchmark tool. [Online]. Available: https://code.google.com/p/stressapptest/wiki/Introduction

[23] X. Wang, J. Zang, Z. Wang, Y. Luo, and X. Li, "Selective hardware/software memory virtualization," *ACM SIGPLAN Notices*, vol. 46, no. 7, pp. 217–226, 2011.

[24] Iozone filesystem benchmark. [Online]. Available: http://www.iozone.org/

[25] Rubis. [Online]. Available: rubis.ow2.org

[26] N. Jain and J. Lakshmi, "Pridyn: Enabling differentiated i/o services in cloud using dynamic priorities," *Services Computing, IEEE Transactions on*, vol. 8, no. 2, pp. 212–224, 2015.

[27] S. L. Garfinkel, "Technical report tr-08-07: An evaluation of amazons grid computing services: Ec2, s3 and sqs," *Computer Science Group, Harvard University, Cambridge, Massachusetts, Tech. Rep*, 2007.

[28] M. G. Xavier, I. C. De Oliveira, F. D. Rossi, R. D. Dos Passos, K. J. Matteussi, and C. A. De Rose, "A performance isolation analysis of disk-intensive workloads on container-based clouds," in *2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*. IEEE, 2015, pp. 253–260.

[29] I. A. Ajay Gulati, Chethan Kumar, "Storage workload characterization and consolidation in virtualized environments."