A Performance Evaluation of Public Cloud Using TPC-C

Jinhui Yao[1,3], Alex Ng[2], Shiping Chen[1,3], Dongxi Liu[3],
Carsten Friedrich[3], and Surya Nepal[3]

[1] School of Electrical and Information Engineering, University of Sydney, Australia
jin.yao@sydney.edu.au
[2] The University of Ballarat, Australia
alexckng@ieee.org
[3] Information Engineering Laboratory, CSIRO ICT Centre, Australia
{firstname.familyname}@csiro.au

Abstract. Cloud is becoming the next-generation computing paradigm for enterprises to deploy services and run business. While most Cloud service providers promise some Quality of Service (QoS) through a Service Level Agreement (SLA), it is very hard for Cloud clients to know what impacts these QoS have on their businesses. In this paper, we study this issue by conducting a simple performance evaluation of two public Clouds. We selected TPC-C to benchmark three types of instances (Small, Medium and Large) provided by the Cloud providers in order to find out how the typical online transaction process systems perform on the cloud nodes. Our testing results show that the different Cloud environments deliver very different performance landscapes with different Cloud instances. Our work demonstrates the importance and opportunity to choose the appropriate Cloud instance in achieving an optimal cost-performance ratio for a class of cloud applications.

Keywords: Cloud Computing, Public Cloud, QoS, Service Level Agreement, Performance, TPC-C Benchmarking.

1 Introduction

Cloud is becoming the next-generation computing paradigm for enterprises to deploy services and run business. According to IDC, Cloud Computing will generate 14 million jobs [11] and $72.9 billion revenues [15] by 2015. Cloud Computing enables IT organisations to become more agile and cost effective with less emphasis on the efforts in maintaining traditional in-house software and hardware. Cloud Computing offers three major delivery models: Infrastructure-as-a-Service (IaaS, for compute, memory, storage, and network resources), Platform-as-a-Service (PaaS, for application development tools and runtime services) and Software-as-a-Service (SaaS, for applications delivered as a service). The primary deployment models include on-premise (private Cloud), off-premise (public Cloud), or a mix of both (hybrid Cloud). Cloud Computing promises to deliver a range of benefits, including avoiding the significant start-up IT investment, reducing operational costs, and enabling greater

A. Ghose et al. (Eds.): ICSOC 2012, LNCS 7759, pp. 3–13, 2013.

agility. By adopting Cloud Computing, business organisations can be more focused on their core business so they can offer better services and innovations to stay competitive in the market.

Service Level Agreement (SLA) is a formal binding contract stating the Quality-of-Service guarantees offered by the Cloud provider (typically including maximum response time, throughput and error rate). Other non-functional QoS such as timeliness, scalability, and availability may also be included. While most Cloud service providers can promise some QoS through the SLA, it is very hard for Cloud clients to know the impacts of these QoS on their businesses. For instance, a Cloud service provider may guarantee that the remote virtual machine the client is hiring has the computing power equivalent to a 1000GHz Intel CPU, however, this statement tells little about how many transactions per minute that virtual machine is able to handle. Furthermore, considering the various virtualization techniques available [2], and the issues with resource sharing among different virtual instances [8] it is quite possible that, while Cloud providers are trying to save money by utilizing resources optimally. Cloud customers (including PaaS and SaaS) are concerned with the runtime performance of their platform and applications. Therefore, it is important and valuable to understand how different Cloud services perform in specific commercial environments for a specific class of applications and how their performance outcomes are related to their claimed SLAs.

In this paper, we conducted a simple performance evaluation of two public Clouds to clarify the above concerns. We selected TPC-C to benchmark three types of instances (Small, Medium and Large) provided by the two public Cloud providers respectively in order to find out the performance of typical online transaction process systems deployed on these different computing instances. We provided our testing results with our observations and analysis in this paper. Since the core of this paper is to study the cloud performance behaviors of different clouds rather than head-to-head comparison, we intend to provide only relative data of cloud instance specifications and prices to preserve the identity of the Cloud vendors.

The rest of this paper is organised as follows: a brief discussion of the Cloud Services versus QoS and SLAs is given in Section 2. Section 3 explains the rationales of our choice of the TPC-C benchmarking technique in performance evaluation of our target Cloud providers. Section 4 provides details of our setup parameters in the Cloud environments and the TPC-C benchmark settings. Section 5 provides the results of our analysis of the data collected in our measurements. Section 6 discusses related work and our conclusions are presented in section 7.

2 Cloud Services vs QoS/SLA

Nowadays, the term ‘Cloud’ has many different definitions. But in its essence, ‘Cloud’ always refers to the computing resources that are to be provided to the clients. Different kinds of resources are provided as different services on a pay-as-you-go basis to extend one’s computing capacity. Amazon S3, for example, is a storage service that allows clients to store and fetch data; another example is Amazon EC2,

which provides computing instances (virtual machines) to let clients deploy and run whatever programs they want.

Coming along with those services, are the QoS guarantees that the service provider offers. QoS guarantees are stated in the form of SLAs which specify the details about the commitment; and (usually) the compensation strategy that will be applied should any violations occur. For example, Amazon offers an SLA about Amazon S3 which states that, if the uptime of the service is less than 99%, the client is entitled to apply for a refund of 25% of his charges incurred in the same billing cycle[1]. It is apparent that the SLAs about storage services are relatively easier to verify and understand than other QoS guarantees about different types the service, which can be unambiguously defined, such as upload speed, download speed, uptime, consistency, etc. However, as we have briefly discussed in the introduction, certain types of the QoS of the computing instances the Cloud is providing is difficult to evaluate. In the practise, what concerns the clients is not how fast the CPU can conduct arithmetic computation; rather, it is the efficiency of whatever program that is running in the instance.

It is the ambiguous link between the SLA, and the true QoS that the client experiences that motivated our research. We are interested to testify the consistency between the claimed and the actual computing capacity of the computing instances offered by Cloud service providers. While it may be infeasible to estimate the true performance of the computing instance based on the performance of a given program; alternatively, we can evaluate the performance of the same program on different computing instances, which are offered by different service providers with the same or similar claimed computing capacity. Significant differences among the results indicate inconsistency in one or more service providers, i.e. the service provider(s) is either over-claiming, or under-claiming.

3 Why TPC-C?

We used Hammerora[2] - an open source load test tool to conduct TPC-C [1] (industry standard benchmark for on-line database processing) to evaluate the performance of two public Clouds. The TPC-C benchmark provides a standard mechanism for performance evaluation of the two different Clouds using the same database architectures and operating systems.

TPC-C is one of the industry-standard benchmark for online transaction processing. TPC-C simulates a complete on-line transaction environment where a population of users executes transactions against a database. The benchmark is built upon a set of principal activities (transactions) of an order-entry environment. These transactions include entering and delivering orders, recording payments, checking the status of orders, and monitoring the level of stock at the warehouses. TPC-C is established since 1992 and has been widely used by the database community for many years and it continues to evolve in order to remain as representative of the current

[1] The complete and formal SLA can be found at `http://aws.amazon.com/ec2-sla/`

[2] Hammerora `http://hammerora.sourceforge.net/`

practice as possible. Both the database and software architecture communities can readily understand the performance figures produced from this exercise.

As shown in Table 1, TPC-C involves a profile of five concurrent transactions of different types and degrees of complexity. The database is comprised of nine types of tables with a wide range of records and population sizes.

Table 1. Summary of TPC-C Transaction Profiles

Transaction	Query Weighing	Database Access	Frequency of Execution	Response Time Requirements
New-Order	Moderate	Read and Write	High	Stringent response time requirement to satisfy on-line users
Payment	Light	Read and Write	High	Stringent response time requirement to satisfy on-line users
Order-Status	Moderate	Read Only	Low	Low response time requirement
Delivery	Light	Read and Write	Low (with deferred execution)	Must complete within a relaxed response time requirement
Stock-Level	Heavy	Read Only	Low	Relaxed response time requirement

Although there are other benchmark standards (e.g. TPC-E & TPC-H) available, we choose TPC-C due to the distributed nature of TPC-C which fits nicely in the distributed requirements of this evaluation exercise. Further, TPC-C sample test codes are readily available from existing software libraries which can shorten the development effort for this exercise and minimize the effect of newly written programs instability issues. Finally, TPC-C is mature and widely recognized as a standard benchmark for database, OS and whole online transaction processing systems.

4 Benchmark Setting

We selected two public Cloud providers (we shall refer to them as Cloud-A & Cloud-B in this paper) to benchmark their Small, Medium and Large Instances respectively. In an effort not to disclose any hint that can be used to identify these two providers, certain ambiguities are introduced in this paper when elaborating the details about the benchmarking processes and results.

All types of instances are running the same operating system. CPU and memory specifications of the instances of the same type provided by the two providers are slightly different. Their differences are listed in the following table.

Table 2. Differences of Cloud-A and Cloud-B Processing Environments

	CPU power	**Memory**
Small instance	25%	3%
Medium instance	25%	7%
Large instance	25%	13%

Please note that the performance results to be shown in the next section does not necessarily correlated to the differences listed (i.e. the computing instance with less claimed CPU power may outperform the one with more).

MySQL 5.1 is installed in each of the subject instances, along with Hammerora 2.10. In every set of load testing, we create different amount of virtual users in Hammerora who will conduct the testing on the local MySQL server by sending requests at a constant rate, then we record the maximum NOPM. The more virtual users created the more concurrent transactions the local MySQL server will undertake. Our presumption for maximum NOPM is that, initially, increasing the number of users will result an increase of NOPM, however, at a certain point, an increase in the number of virtual users will no longer result a higher NOPM, as it has reached the limit of the computing power of the instance, and this is the maximum NOPM the subject instance is able to achieve.

5 Experimental Results and Analysis

We are going to discuss in this section the performance metrics used in our experiments and the analysis of the results observed.

5.1 Performance Metrics Used

TPC-C is measured in transactions per minute (tpm). In our experiments, we measured the Number of TPC-C Opeartion Per Minute (NOPM). We increased the

numbers of concurrent Virtual Users gradually from 8 to 256. We then measured the maximum NOPM as experienced by each Virtual User.

The second metric we use is the Peak-to-Saturation-Bottom (PTSB) ratio to determine how stable the Cloud environments are. We calculate the ratio between the Peak NOPM value and the Lowest/Bottom NOPM value (as the instance went into saturation mode) offered by a particular instance for a particular Cloud environment. The ratio between these two values will give a good idea how well the Cloud environment can offer a stable processing environment to meet the increase in demand. Therefore, the closer this ratio is to 1 the better.

The third metric we use is the Total NOPM (TNOPM) which calculates the total NOPM offered by a particular instance. It is the total sum of all individual NOPMs. This metric will give a good idea how well the instance will cope with all the transactions working within a particular instance.

The fourth metric is the Cents-Per-Million-NOPM (CPMNOPM), it looks at the cost required to deliver One Million NOPM for each instance of Cloud-A and Cloud-B. It is a ratio of the costs of acquiring each instance and the total number of NOPM supported by each instance. This metric will inform us the actual cost of supporting the application.

5.2 Cloud-A Analysis

Figure 1 shows the maximum NOPM experienced by different of number of concurrent virtual users for Cloud-A using the Small, Medium and Large Instances of Cloud-A.

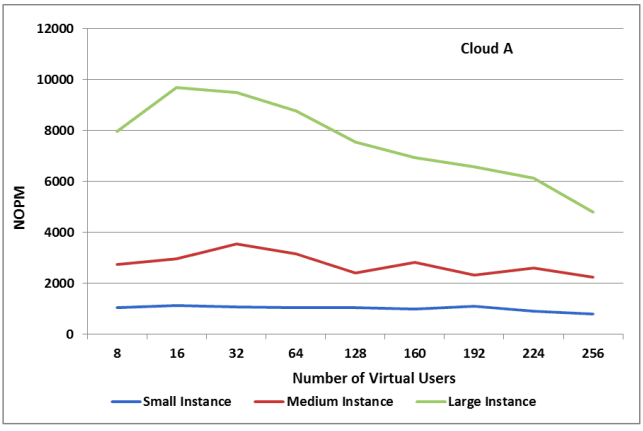

Fig. 1. Cloud-A Number of Operation Per Minute

We observed that Cloud-A with Small Instance is able to deliver a fairly steady NOPM for different number of virtual users with peak 1100 NOPM at 192 virtual users and bottom 795 NOPM at 256 virtual users. Cloud-A with Medium Instance provides peak performance of 3554 NOPM at 32 virtual users and then decreases

gradually to 2246 NOPM when the number of virtual users increases to 256. As for Large Instance in Cloud-A, the peak 9691 NOPM occurs at 16 virtual users and then decreases steeply to 4807 NOPM at 256 virtual users. This analysis of NOPM for Cloud-A shows the following:

1. The NOPM jumps from 1015 NOPM for Small Instance to 2758 NOPM for Medium Instance and 7542 NOPM for Large Instance. We found that Medium Instance offers 170% increase in NOPM to Small Instance and Large Instance also offers 170% increase in NOPM to Medium Instance.
2. We use the Peak-to-Saturation-Bottom (PTSB) ratio to determine how stable (close to 1 is stable) the Cloud environment offers. It seems the Small Instance of Cloud-A provides a more stable operating environment (PTSB ratio of 1.38) than the Medium Instance (PTSB ratio is 1.58) and the Medium Instance is more stable than the Large Instance (PTSB ratio is 2.01) of Cloud-A.

At this point, we would like to assert that the larger the resource instance for Cloud-A, the slower, or more time is required to compensate for the increase in the demand for more resources. Our explanation is that it is a logical behaviour because the larger the instance, the more effort and time are required to acquire the extra resources to meet the increase in demand. Hence, this is an important research area for Cloud providers in identifying better algorithms in dealing with this situation.

The PTSB ratio of NOPM only gives us an idea how well different instances are able to provide a stable and consistent environment when the workload is varied from small to large number of concurrent virtual users. This shows the perspective of NOPM as seen at each Virtual User only and may not reveal the overall elasticity nature of the Cloud. We calculate the Total NOPM (TNOPM) for different number of Virtual Users to find out the total volume handled by each instance. The results are summarised in the following Figure 2.

We can see that the TNOPM for Cloud-A Small, Medium and Large Instances all increase in a linear fashion until the maximum values occur at 224 Virtual Users, with 203616 TNOPM for Small Instance, 583520 TNOPM for Medium Instance, and

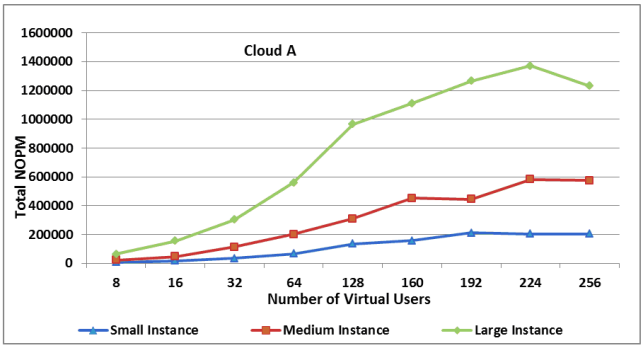

Fig. 2. Cloud-A Total Number of Operation Per Minute

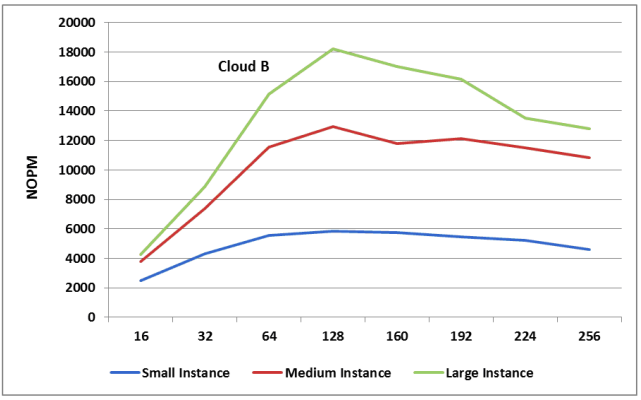

Fig. 3. Cloud-B Number of Operation Per Minute

1371328 TNOPM for Large Instance. At 256 Virtual Users, all three instances either maintained at similar level or start to show decrease in TNOPM. This shows that Cloud-A is able to scale linearly, or elastic, up to a certain threshold point, in this case 224 Virtual Users. Hence, we consider that Cloud-A is able to demonstrate good elasticity nature from 8 to 224 Virtual Users.

5.3 Cloud-B Analysis

We observed that Cloud-B with Small Instance delivers a steady NOPM for different number of virtual users with peak 5831 NOPM at 128 virtual users and decrease gradually to 4619 NOPM at 256 Virtual Users. Cloud-B with Medium Instance provides peak performance of 12942 NOPM at 128 virtual users and then decreases slightly to 10811 NOPM when the number of Virtual Users increases to 256. As for Large Instance in Cloud-B, the peak 19214 NOPM occurs at 128 Virtual Users and then decreases steeply to 12790 NOPM at 256 Virtual Users. This analysis of NOPM for Cloud-B shows the following:

1. The NOPM for Small Instance is 4909, Medium Instance is 10245 and Large Instance is 13248. Medium Instance offers 108% increase in NOPM to Small Instance and Large Instance offers only 29% increase in NOPM to Medium Instance.
2. Cloud-B seems offer more stable operating environment than Cloud-A as we observed the PTSB ratio for Small Instance is 1.26, Medium Instance is 1.19 and Large Instance is 1.42, all offers lower value than the corresponding Cloud-A environment. That means Cloud-B seems able to react quicker than Cloud-A in acquiring extra resources to meet the increase in demand from the increasing number of Virtual Users.

The TNOPM figures for Cloud-B (Figure 4) show that all three instances were able to scale linearly with no sign of saturation even up to 256 Virtual Users. There was a dip in the Large Instance from 192 Virtual Users to 224 Virtual Users but the upward

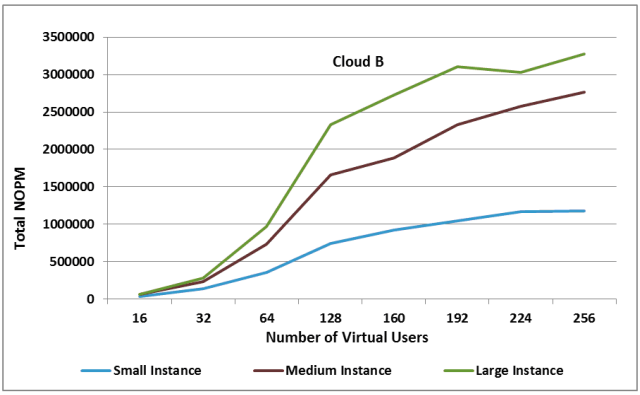

Fig. 4. Cloud-B Total Number of Operation Per Minute

trend continued at 256 Virtual Users. The maximum values are 1182464 TNOPM for Small Instance, 2767616 TNOPM for Medium Instance and 3274240 TNOPM for Large Instance. From this TNOPM analysis, we observed that Cloud-B outperforms Cloud-A in the elasticity aspect of a Cloud environment because Cloud-A reached its saturation point at 224 Virtual Users for all three instances while Cloud-B did not saturate even at 256 Virtual Users.

5.4 Cost Analysis

The next aspect we are interested in the performance of a Cloud environment is the cost effectiveness. Since the costs of acquiring each instance are readily available from the Cloud vendors, we use the metric Cents-Per-Million-NOPM (CPMNOPM) to look at the cost required to deliver One Million of NOPM for each different instances of Cloud-A and Cloud-B. The results are summarised in the following table (only the ranges are provided to preserve the identity of the Cloud vendors):

Table 3. Cents Per Million NOPM for Cloud-A & Cloud-B

	Cloud-A	**Cloud-B**
Small Instance	High – 18.2 Low – 0.72	High – 4.98 Low – 0.17
Medium Instance	High – 13.9 Low – 0.52	High – 6.63 Low – 0.14
Large Instance	High – 9.62 Low – 0.44	High – 11.7 Low – 0.24

We can see that for Small and Medium instances, Cloud-A is more expensive (over 100%) than Cloud-B to support one million NOPM. When it comes to Large Instance, the landscape is a little bit different, because in the case of small number of Virtual Users a low volume level of NOPM is achieved, then Cloud-B appears to be more expensive than Cloud-A to support one million NOPM. This is because the

entry cost to acquire Cloud-B Large Instance is more expensive than Cloud-A. With low transaction volume level, it does not fully utilise the resources available at Cloud-B so that it becomes more expensive than Cloud-A. However, at high level of transaction volume, Cloud-B becomes more cost effective than Cloud-A to run because Could-B is able to deliver much higher number of NOPM than Cloud-A.

This is an interesting but logical finding which shows that when investing into acquiring a Cloud environment to support an application, a proper capacity estimation should be performed to avoid over investment.

6 Related Work

SLA between consumers and Cloud providers is one of the key research topics in dealing with the dynamic natures of the Cloud environment which requires continuous monitoring on various QoS attributes as well as considering many other factors such as delegation and trust. The research in Cloud QoS is progressing [3]. Different mechanisms were proposed to monitor QoS attributes in the Cloud, for example, the Web Service Level Agreement (WSLA) framework [12] for Cloud Web Services; SLA-aware management framework [6, 7]; and using business service management reservoir approach in governing the SLAs for individual application's non-functional characteristics [13].

Bautista et al. [4] integrated software quality concepts from ISO 25010 to measure seven different Cloud quality concepts (Maturity, Fault Tolerance, Availability, Recoverability, Time Behaviour, Resource Utilisation, and Capacity). We found these quality concepts lack clearly measurable metrics. The evaluation methodology used in this work is simpler and can produce more meaningful results.

We found only a few analysis on Cloud performance were available either with focus on the Amazon's EC2 Cloud environment [5] or Cloud for scientific computing [9, 14]. CloudCmp [10] uses a matrix of measures in the elastic computing, persistent storage, and networking services offered by a cloud to act as a comparator of the performance and cost of cloud providers. We consider CloudCmp to be too complicated for the general public.

7 Conclusion

Very little studies are available on the benchmarking of the characteristics of different commercial Cloud platforms. In this paper, we have presented our understanding on the performance of 2 public Cloud environments on supporting typical commercial applications.

We have used the TPC-C benchmarking technique to study the NOPM, PTSB, TNOPM and CPMNOPM characteristics of the 2 Clouds. Our findings are: (1) both Clouds are able to provide stable and elastic processing environments to meet the demand of increase in load; (2) the NOPM analysis, Cloud-B with higher CPU processing power is able to deliver higher NOPM than Cloud-A; (3) the PTSB analysis, Cloud-B is able to react quicker than Cloud-A in acquiring extra resources; and (4) the CPMNOPM analysis reveals that the cheaper cloud environment does not necessary provide the more cost-effective processing environment. It is very

important to understand the cost-effectiveness aspect in acquiring the appropriate Cloud Instances so as to avoid over investment.

Our studies present only the first step in understanding the performance characteristics of the public Clouds. There are more cloud aspects needed to evaluate including the characteristics of Transparency, Trust, Portability, Interoperability, and Continuity in public Clouds.

References

[1] TPC-C Home Page, http://www.tpc.org/tpcc/default.asp (accessed April 26, 2012)

[2] Understanding Full Virtualization, Paravirtualization, and Hardware Assist, VMWare (2007)

[3] Armstrong, D., Djemame, K.: Towards Quality of Service in the Cloud. In: Proceedings of the 25th UK Performance Engineering Workshop, Leeds, UK (2009)

[4] Bautista, L., Abran, A., April, A.: Design of a Performance Measurement Framework for Cloud Computing. Journal of Software Engineering and Applications 5(2), 69–75 (2012)

[5] Dejun, J., Pierre, G., Chi, C.-H.: EC2 Performance Analysis for Resource Provisioning of Service-Oriented Applications. In: Dan, A., Gittler, F., Toumani, F. (eds.) ICSOC/ServiceWave 2009. LNCS, vol. 6275, pp. 197–207. Springer, Heidelberg (2010)

[6] Ferretti, S., Ghini, V., Panzieri, F., et al.: QoS-Aware Clouds. In: Proceedings of the IEEE 3rd International Conference on Cloud Computing (2010)

[7] Fito, J.O., Goiri, I., Guitart, J.: SLA-driven Elastic Cloud Hosting Provider. In: Proceedings of the 18th Euromicro Conference on Parallel, Distributed and Network-based Processing (2010)

[8] Gupta, D., Cherkasova, L., Gardner, R., Vahdat, A.: Enforcing Performance Isolation Across Virtual Machines in Xen. In: van Steen, M., Henning, M. (eds.) Middleware 2006. LNCS, vol. 4290, pp. 342–362. Springer, Heidelberg (2006)

[9] Iosup, A., Ostermann, S., Yigitbasi, M.N., et al.: Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing. IEEE Transactions on Parallel and Distributed Systems 22(6), 931–945 (2011)

[10] Li, A., Yang, X., Kandula, S., et al.: CloudCmp: comparing public cloud providers. In: Proceedings of the 10th Annual Conference on Internet Measurement (IMC 2010), Melbourne, Australia, pp. 1–14 (2010)

[11] McKendrick, J.: Cloud Will Generate 14 Million Jobs By 2015: That's A Good Start (March 5, 2012), http://www.forbes.com/sites/joemckendrick/2012/03/05/cloud-will-generate-14-million-jobs-by-2015-thats-a-good-start/ (accessed April 16, 2012)

[12] Patel, P., Ranabahu, A., Sheth, A.: Service Level Agreement in Cloud Computing. In: Proceedings of the Cloud Workshop at OOPSLA (2009)

[13] Rochwerger, B., Breitgand, D., Levy, E., et al.: The Reservoir model and architecture for open federated cloud computing. IBM Journal of Research and Development 53(4) (2009)

[14] Wang, G., Ng, T.S.E.: The Impact of Virtualization on Network Performance of Amazon EC2 Data Center. In: Proceedings of the INFOCOM 2010 (2010)

[15] http://www.idc.com/prodserv/idc_cloud.jsp