Smart CloudBench - Test Drive the Cloud Before You Buy

Mohan Baruwal Chhetri(✉), Sergei Chichin, Quoc Bao Vo, and Ryszard Kowalczyk

Faculty of Information and Communication Technologies, Swinburne University of Technology, Melbourne, VIC 3122, Australia
{mchhetri,schichin,bvo,rkowalczyk}@swin.edu.au

Abstract. In recent years there has been an exponential growth in the number of vendors offering Infrastructure-as-a-Service (IaaS), with a corresponding increase in the number of enterprises looking to migrate some, or all of their IT systems to the cloud. Prospective cloud consumers need to identify providers that offer resources with the most appropriate pricing and performance levels to match their specific business needs before making any migration decisions. However, no two vendors offer the same resource configurations, pricing models or provisioning models. Moreover, cloud vendors tend to use different virtualization techniques which impact the performance of the software systems running on top of their infrastructure. Since consumers only have a black-box view of the cloud, it makes the task of comparing and selecting appropriate computing resources a very complex exercise. In this paper, we present Smart CloudBench, which is a suite of software tools that allows prospective cloud consumers to *test drive* the cloud and *make purchasing decisions* based on price, specification AND performance. Cloud consumers can use Smart CloudBench for the automated, on-demand, real-time and customized benchmarking of cloud infrastructure and use the benchmarking results along with the pricing and specification information to make more informed purchasing decisions. Tests using Smart CloudBench show that the performance of higher priced servers is not necessarily better than that of lower priced ones, and it has to be tested extensively in order to substitute assumptions with facts.

Keywords: Cloud infrastructure selection · Performance benchmarking · Automated benchmarking

1 Introduction

In recent years there has been an exponential growth in the number of vendors offering Infrastructure-as-a-Service (IaaS), with a corresponding increase in the number of enterprises looking to migrate some, or all of their IT systems to the cloud. Prospective cloud consumers would like to obtain a quick assessment of the price, specification and performance of competing IaaS providers

J.G. Davis et al. (Eds.): ASSRI 2013, LNBIP 177, pp. 59–73, 2014.
DOI: 10.1007/978-3-319-07950-9_5,

before making any migration decisions. While the pricing and specifications is public information, the performance of the computing infrastructure is unknown. Different providers use different virtualization techniques which impacts the performance of software systems running on top of their infrastructure; the only way to compare providers based on performance is benchmarking software systems on top of the cloud infrastructure and not relying on any assumptions based on price and specification. One approach to do this is to benchmark the cloud infrastructure performance by deploying own applications on selected cloud platforms and testing them under variable workloads. However, this approach can be complex, time-consuming and expensive, and very few organizations possess the time, resources and in-house expertise to do a thorough and proactive evaluation in this manner. A more practical alternative is to test representative applications[1] against representative workloads to estimate the performance of cloud providers. The benchmarking results can then be used to quantify application performance on the different IaaS platforms and to obtain valuable insights into the difference in performance across providers. By combining the benchmarking results with pricing information and resource specification, enterprises can better identify the most appropriate cloud providers and offerings based on their specific business needs.

In this paper, we present Smart CloudBench, a suite of software tools that allows prospective cloud consumers to test-drive public cloud infrastructure. It enables the measurement of infrastructure performance in an efficient, quick and cost-effective manner, through the automated execution and analysis of representative benchmarks on multiple IaaS clouds under customized workloads. Prospective cloud consumers can use Smart CloudBench to (i) select the representative application/s to use for evaluating cloud performance, (ii) configure the test harness, (iii) select and acquire instances on the cloud platforms to be tested, (iv) run the benchmark tests, and (v) aggregate the results to build a price/specification/performance matrix that can help with decision-making for provider and resource selection. The key benefits of using Smart CloudBench include:

- Reduction in time and effort involved in benchmarking cloud platforms. If the number of cloud instances to benchmark is high, and the number of representative applications is large, then manually executing the benchmarking process becomes a very cumbersome exercise.
- Reduction in performance testing costs. Since the cloud resources to be tested can be commissioned just in time and decommissioned immediately after completion of the tests, there are significant cost savings.
- Simplification of repetition of the benchmark process with reduction in human error.
- Automated and customized generation of reports and analytics for consumption by technical and non-technical audiences.

[1] Some example representative applications include TPC-W for a transactional e-commerce web application [5] and Media Streaming benchmark application for media streaming applications such as Netflix or Yuku [6].

– Centralised storage of performance data, which over time enables analysis of performance evolution.
– Performance benchmarking of cloud infrastructure can be offered as a service.

The rest of the paper is organized as follows: In Sect. 2, we briefly discuss performance benchmarking and how it relates to cloud infrastructure. In Sect. 3, we give an overview of Smart CloudBench and present its key components. In Sect. 4 we explain how Smart CloudBench works. We present some benchmarking results and discuss their significance in Sect. 5. We discuss related work in Sect. 6 and conclude the paper by discussing future work in Sect. 7.

2 Performance Benchmarking of Cloud Infrastructure

In the IaaS service model, the service provider gives consumers the capability to provision processing, storage, network and basic computing resources on demand. While the consumer has control over the operating system, assigned storage and the deployed applications, it has no control over the underlying cloud infrastructure. When a client requests and receives virtual machines from a cloud provider, it perceives the provisioned resource as a black-box whose runtime behaviour is unknown. The use of different virtualization techniques by different providers affects the performance of software systems running on top of the cloud infrastructure. Therefore, there is a need for tools and techniques to measure and compare the performance of computing resources offered by different cloud providers. Benchmarking is a traditional approach for verifying that the performance of a system meets the expected levels and to facilitate the informed procurement of computer systems. In the context of cloud infrastructure, performance benchmarking can serve a number of different purposes including (a) determining whether a particular server configuration meets the performance criteria, (b) comparing two configurations to find out which one performs better, and (c) determining the level of QoS that can be guaranteed to end-users of software systems deployed on the cloud infrastructure.

2.1 Elements of Benchmarking

The key elements of any benchmarking process are (a) *System Under Test (SUT)*, which refers to the system whose performance is being evaluated, (b) the *workload*, which refers to the operational load that is used to test the SUT, and (c) the *test agent (TA)* which is the test infrastructure that is used to carry out the benchmark tests. In our work, the SUT is the virtual cloud server whose performance we are interested in. It is viewed as a *black box*, whose operational details are not exposed and evaluation is based only on its output. The test agents are also deployed on cloud infrastructure as the cloud is perfectly suited to deliver scalable test tool environments which are necessary for the different types of performance testing. Thus, with Smart CloudBench, the cloud infrastructure forms the test environment and can also be used as the test harness.

There are two ways to benchmark the cloud infrastructure: micro benchmarking and application stack benchmarking. While a set of micro benchmarks can offer a good starting point in evaluating the performance of the basic components of the cloud infrastructure, application stack benchmarking offers a better understanding of how a real-world application will perform when run on top of the cloud infrastructure. Hence, we focus more on benchmarking the performance of the entire application stack. If prospective consumers can find representative benchmarks for their in-house applications, they can design experiments to match the internal load levels and load variations, and then test the representative application to determine how the different clouds compare performance wise and cost wise. By using representative performance benchmarking, consumers can quickly assess multiple cloud providers and their offerings in an objective, consistent and fully automated manner without having to deploy their own applications on the various cloud platforms.

2.2 Performance Characteristics

Performance is a key quality of service attribute that is important to both cloud consumers and cloud service providers. It should not only be specified and captured in Service Level Agreements (SLA) but should also be tested in order to substitute assumptions with hard facts. For example, intuitively, *a 16 GB server with 8 vCPUs is expected to perform better than a 8 GB server with only 4 vCPUs*. However, the actual performance benchmarking might reveal different results as shown in Sect. 5. In the context of cloud-based IT solutions and applications systems, the following performance characteristics can be of particular interest to prospective cloud consumers.

- *Time behaviour:* This performance characteristic captures the response time, the processing time and the throughput rate of the software system running on the cloud infrastructure, which subjected to a given workload.
- *Capacity:* This performance characteristic describes the maximum limits of the software system parameters i.e. the number of concurrent users of the system, the communication bandwidth, the throughput of the transactions etc.
- *Resource utilisation:* This performance characteristic describes the degree to which the amounts and types of resources are utilised by the system under a given workload. This characteristic can help identify over-provisioned and/or under-performing resources.

2.3 Types of Performance Tests

Depending upon the objectives of performance testing, there are different types of performance tests that can be carried out:

- *Response Performance Testing:* This form of performance testing is used to measure the responsiveness and duration of an IT system. This is conducted to understand the behaviour of the system under a specific expected load.

The load can be the expected concurrent number of users of the system performing a specific number of transactions within the set duration.
– *Stress Testing:* This form of testing is used to determine the boundaries of the SUT. A heavy load is generated to simulate unusual user behaviour and is used to determine if the system will perform sufficiently under extreme load conditions.
– *Soak Testing:* This form of testing is used to determine if the system can sustain continuous expected load without any major deterioration in performance. It involves testing the system with a significant load continuously over a significant period of time and observe the system behaviour under sustained use.
– *Scalability Load Test:* This test is used to determine how the SUT will scale for increasing load.
– *Spike Testing:* This form of testing is used to determine how the system behaves when subjected to sudden spikes in workload - will the system performance suffer, will it fail or will it successfully handle the dramatic changes to load.

3 Overview of Smart CloudBench

In this section we present a detailed description and reference architecture for Smart CloudBench. Smart CloudBench is a configurable, extensible and portable system for the automated performance benchmarking of cloud infrastructure using representative applications from a suite of benchmark applications. It also enables the comparison and ranking of different cloud service offerings based on user requirements in terms of infrastructure specifications, application performance, costs, security, geographic location, compliance, regulatory requirements and other requisite criteria [1,2]. It forms a key component of the larger Smart Cloud Broker suite[2] which comprises of the following additional components:

– *Smart CloudMonitor* - is a solution that enables the monitoring of cloud resource consumption patterns. It can be used in conjunction with Smart CloudBench to monitor cloud resource utilization during benchmarking in order to identify over-provisioned and under-performing configurations.
– *Smart CloudPurchaser* -enables the automated procurement and consumption of computing resources based on business rules specified by the cloud consumer [3,4].
– *Smart CloudMarketplace* - offers an open electronic market where multiple cloud consumers and providers can efficiently trade IaaS based on the supply and demand mechanisms.

The main components of Smart CloudBench include:

– *Benchmark Orchestrator (BO)* - This is the main module of Smart CloudBench. It orchestrates the automated performance benchmarking of IaaS

[2] www.smartcloudbroker.com

clouds. It controls the entire process including benchmark and provider selection, workload description, resource management, workload generation, workload execution and result collection. It automates all the tasks that would be manually carried out in a normal benchmarking exercise.

- *Cloud Comparator (CC)* - This module allows users to automatically compare the different cloud providers based on the cost and configuration of the offered servers (which is stored in the provider catalog database), and the performance benchmarking results stored in the benchmark results database. *Report Generator* generates test reports in different formats including graphical, tabular and textual formats for consumption by both technical and non-technical users. *Visualizer* component allows users to visualize the test results and use different ranking and evaluation criteria to rank them.
- *Cloud Manager (CM)* - This module performs fundamental cloud resource management. *Instance Manager (IM)* procures appropriate instances on the different providers - both for the System Under Test (SUT) and the Test Agents (TA) based on the resource provisioning instructions from the BO. It is also responsible for the decommissioning of the instances at the end of each test. *Virtual Machine Image (VMI) Manager* is responsible for creating and maintaining virtual machine images on the different cloud providers. *Common Cloud Interface (CCI)* provides a common interface to different public cloud providers and enables the automated management of cloud instances including instantiation and termination.
- *Cloud Provider and Benchmark Catalogs* - Smart CloudBench maintains a catalog of supported IaaS providers and their offerings. It also maintains a catalog of supported benchmarks for the different types of representative applications.
- *Benchmark Results Database* - The results of the performance benchmarking are stored in the benchmark results database and can be used for analysing the evolution of cloud performance over time.

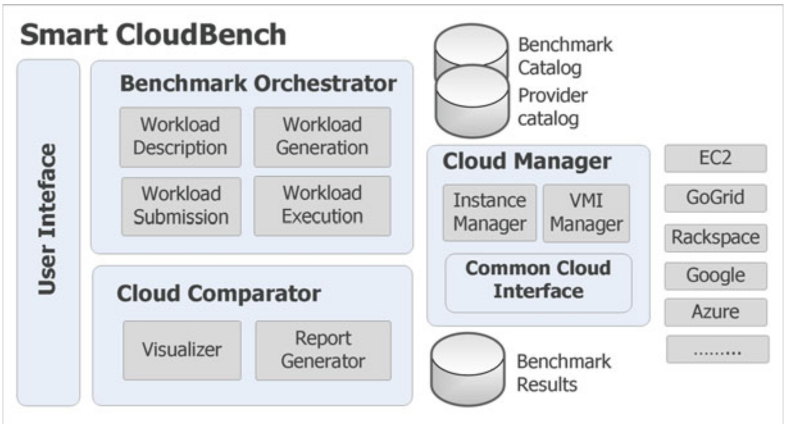


Fig. 1. Smart CloudBench architecture

– *User Interface (UI)* - The user interacts with Smart CloudBench through a browser-based UI (Fig. 1).

4 Using Smart CloudBench

In this section, we explain the steps involved in executing a typical benchmark using Smart CloudBench (see Fig. 2). We also include relevant screenshots to illustrate the usage scenarios (see Fig. 3).

– *Provider Selection* - In Step 1, the user selects the specific cloud providers and resource configurations to test. This selection is done based on user requirements, which could include resource configuration, cost, geographic location, supported operating systems etc.
– *Benchmark Selection* - In Step 2, the user selects the representative benchmark application/s from the list of available benchmarks that is to be used to evaluate the performance of the selected cloud server configurations.
– *Workload Specification* In Step 3, the user defines different scenarios to be tested against the selected benchmark. The request (comprising of the selected benchmark, test scenarios, and cloud servers to be tested) is submitted to the BO. The first and second steps can be used interchangeably.
– *Instance Procurement* - In Step 4, the BO receives the benchmarking request and directs the CM to procure the required cloud server instances from the

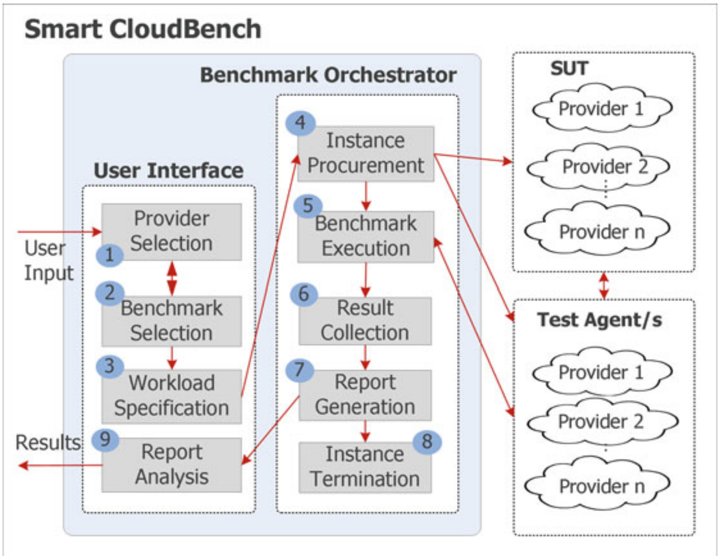


Fig. 2. Smart CloudBench workflow

	Server ID	Logo	Provider	Location	CPU	CPU Model	Memory	Storage	OS	Price
	Small		Amazon	US-West(California)	1	EC2	1.7	160	Windows	0.096
	Small		Amazon	Asia-Pacific(Tokyo)	1	EC2	1.7	160	Windows	0.115
	Small		Amazon	Asia-Pacific(Singapore)	1	EC2	1.7	160	Windows	0.115
	Small		Amazon	Asia-Pacific(Sydney)	1	EC2	1.7	160	Windows	0.115
	Medium		GoGrid	US-West	2	GG Core	2	100	Windows	0.16
	2GB		Rackspace	Chicago	2	RS Core	2	80	Windows	0.16
	Medium		Amazon	US-West(California)	2	EC2	3.75	410	Windows	0.192
	Medium		Amazon	Asia-Pacific(Singapore)	2	EC2	3.75	410	Windows	0.23
	Medium		Amazon	Asia-Pacific(Sydney)	2	EC2	3.75	410	Windows	0.23
	Medium		Amazon	Asia-Pacific(Tokyo)	2	EC2	3.75	410	Windows	0.23
	High-CPU.medium		Amazon	US-West(California)	5	EC2	1.7	320	Windows	0.245
	High-CPU.medium		Amazon	Asia-Pacific(Sydney)	5	EC2	1.7	320	Windows	0.285
	High-CPU.medium		Amazon	Asia-Pacific(Singapore)	5	EC2	1.7	320	Windows	0.285
	High-CPU.medium		Amazon	Asia-Pacific(Tokyo)	5	EC2	1.7	320	Windows	0.285

(a) Provider Catalog

WorkLoad : Servers : Client Location : Asia-Pacific(Sydney) Scenario : Browsing Retrieve Images? :

	Date	Server ID	Number of Servers	Load	Scenario	Get Images	Provider	Server Location
	06/06/2013	xLarge	1	100	Browsing	No	Amazon	JS-West(Califor
	06/06/2013	X-Large	1	100	Browsing	No	GoGrid	JS-West
	06/06/2013	8GB	1	100	Browsing	No	Rackspace	hicago
	06/06/2013	xLarge	1	200	Browsing	No	Amazon	JS-West(Califor
	06/06/2013	X-Large	1	200	Browsing	No	GoGrid	JS-West
	06/06/2013	8GB	1	200	Browsing	No	Rackspace	hicago

(b) Workload Specification

(c) Result Summary

Fig. 3. Smart CloudBench UI

selected providers. Technically, the CM generates requests to the required cloud provider's APIs in order to launch VMs with specific server configurations as specified in Step 1. Pre-built images containing the packaged applications are used to start up the SUT and the TA.

– *Benchmark Execution* - In Step 5, the BO executes the benchmark by issuing remote calls to the test agents running on the newly started cloud machines and waits for the benchmark results to be returned to it.
– *Result Collection* - On completion of the tests, the TAs return the benchmarking results to the BO in Step 6.
– *Report Generation and Visualisation* - In Step 7, the BO updates the Benchmark Results Database. The user can visualize the results either in tabular format or in graphical format. The reports combine the pricing and configuration information with the performance results. Users can use the test results for further analysis and decision-making.
– *Instance Decommissioning* - Once the tests have finished, the BO requests the CM to decommission the instances that were initally started up for the tests in Step 8.
– *Report Analysis* - In Step 9, the user can analyze the test results returned by Smart CloudBench.

5 Benchmarking Results

In this section, we describe the experimental environment we have used to demonstrate and validate the usefulness of Smart CloudBench. The representative benchmark application that we have used in our experiments is TPC-W [5], which simulates an on-line retail store. We have selected this particular application because it represents the most popular type of application running on the cloud and its behaviour is relatively simple and well understood. We first describe the experimental setup and the measured metrics followed by the results of the benchmarking tests performed on 3 different servers offered by a large IaaS provider in Australia.

5.1 TPC-W Benchmark

The TPC-W application models an online bookstore which is representative of a typical enterprise web application. It includes a web server to render the web pages, an application server to execute business logic, and a database to store application data. It is designed to test the complete application stack and does not make any assumptions about the technologies and software systems used in each layer. The benchmark consists of two parts. The first part is the TPC-W application which supports a mix of 14 different types of web interactions and three workload mixes, including searching for products, shopping for products and ordering products. The second part is the remote browser emulation (RBE) system which generates the workload to test the application. The RBE simulates the same HTTP network traffic as would be seen by a real customer using the browser. An open source implementation of TPC-W is available online.[3]

[3] Source code is available for both the TPC-W benchmark server implementation as well as the client implementation (TA) is available online at http://www.cs.virginia.edu/th8k/downloads/.

During each benchmarking cycle, the TPC-W client generates a random number of simultaneous requests to the server, depending on the specified number of emulated browsers. A single emulated browser can request only one web-page at a time. The client also simulates the waiting time between the browsing sessions of each emulated user. The server responds to the requests of the client by generating the corresponding web-pages. In case the request time exceeds 25 s, the request is dropped by timeout. The total number of requests that are made in a single benchmarking cycle varies depending on the response time. If the server cannot cope with the workload, the average response time and the number of timeouts will be high. In such case the number of generated requests will be lower, than when the server is capable of handling the generated workload and responds faster to the incoming requests.

5.2 Experimental Setup

We selected three large servers - 8GB, 16GB and 24GB servers (as shown in Table 1) to run the benchmark application on. The workload for each server was generated from separate test agents which operated on 24GB servers. Both the SUT and the TAs were located in Sydney. Workloads of 500 and 1000 concurrent clients were used to test the server performance over time. The benchmarking tests were run in parallel for 5 full days starting on Friday, 23/08/2013 at 6 pm and finishing on Wednesday, 28/08/2013 at 6 pm.

Table 1. Configurations of the instances used in the benchmarking experiments (prices correct on 23/08/2013)

Server	RAM (GB)	vCPU	Price (AUD/h)
S1	8	4	0.629
S2	16	8	1.246
S3	24	8	1.8

The benchmarking exercise was configured to run as follows. The duration of each benchmarking cycle was set at 5 min.; 2–2.5 min. for the benchmarking exercise and 2.5–3 min. pause before resuming the next round of testing. We paused at the end of each benchmarking cycle in order to minimise the impact of the congested server requests on the server performance in the next benchmarking cycle. As part of the benchmarking exercise, we collected the following metrics:

- Average Response Time (ART)
- Maximum Response Time (MRT)
- Total Number of Successful Interactions (SI)
- Total Number of Timeouts (T)

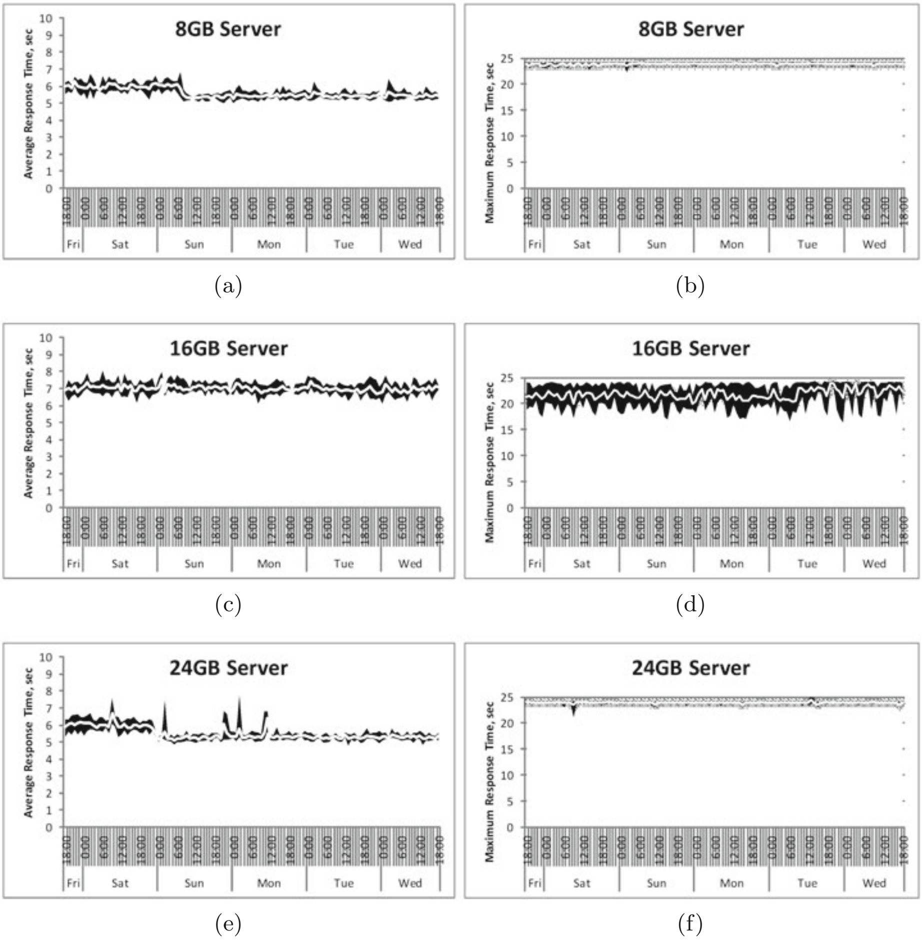
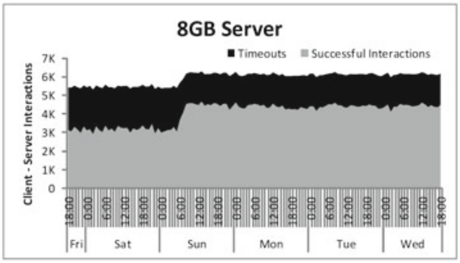


Fig. 4. Average and maximum response time (over time)

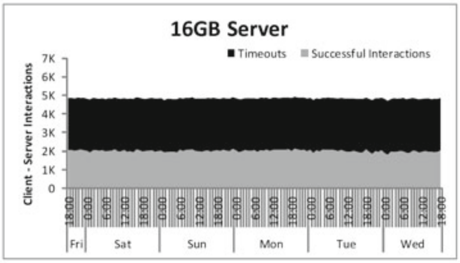
5.3 Discussion of Results

The results of the benchmarking exercise are presented in Figs. 4 and 5. The figures display the average response time, the maximum response time, the total number of successful interactions and the total timeouts, all measured over time. The black zones around the average figure (white line) represent the variance of performance in a particular hour. On analysing the benchmark results, we made the following observations.

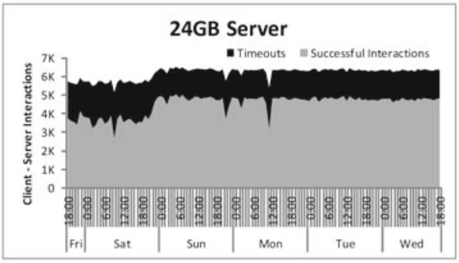
– **Performance of 16GB server is significantly lower compared to that of 8GB and 24GB servers.** In Fig. 4 we can see that the ART of the 16GB server fluctuates consistently between 6 and 8 s. In contrast, 8GB and 24GB servers have a much better ART, which is on-average around 4 s until



(a)



(b)



(c)

Fig. 5. Timeouts and successful interactions

Saturday midnight and then drops to around 2 s afterwards, showing improved performance. However, the 8 and 24GB servers do have several spikes, where the performance drops significantly, whereas the 16GB server shows more consistent performance. If we compare the price of the three servers, we can see that the 16GB server is nearly twice as costly as the 8GB server, and the 24GB server is nearly three times more expensive than the 8GB server. However, the performance of the 8GB server is better than that of the 16GB server and comparable with that of the 24GB server (for the workload of 500 and 1000 concurrent users). These results give a clear indication that making assumptions about the performance of cloud infrastructure based on the price and the specification is not a good decicion-making approach.

- **The server performance varies quite significantly over time.** We can observe that the performance of the 8 and 24GB servers improved significantly on Sunday; the ART dropped from 4 to 2 s and the total number of client-server interactions increased up to 1000 requests, while the number of requests timeouts dropped to insignificant value. A potential reason for such behaviour could be CPU bursting which is essentially the availability of additional CPU cycles due to less CPU contention.
- **When the workload increases the server performance becomes more predictive.** In the case of all three servers we can observe that when the workload increases, the deviation in server performance becomes smaller. Such

behaviour is most likely linked with the way the TPC-W client generates the requests to the server. As TPC-W simulates the real user behaviour, each emulated browser requests a web-page and waits for the server response before issuing another request to the server. Obviously, when the server is congested, it takes longer time to respond and fewer requests are generated in a single benchmarking cycle (about 150 s). As a consequence, it is possible that 500 and 1000 EBs can generate the same number of requests. We can see in the Fig. 5 that the 16GB server is overloaded and receives 2500 requests in total; however the 24GB server receives 5000 requests, which is twice more. Moreover, when the server is not capable to cope with the generated workload more requests are dropped by the timeout.

6 Related Work

There are a number of commercial and academic tools that provide support for cloud performance benchmarking. CloudHarmony[4] provides an extensive database of benchmark results for a fee across a number of public cloud providers using a wide range of benchmark applications. Cloud Spectator[5] is another provider which carries out periodic benchmarking and publishes the results in reports which can be purchased. ServerBear[6] measures CPU, IO, IOPS and network performance and provides customised reports against selected providers for a fee. Cedexis[7] offers tools for the real time monitoring of response times to over 100 cloud providers and Global Delivery Networks.

There are also several academic research projects in this area. CloudCmp [8] is a framework to compare cloud providers based on the performance of the various infrastructure components including computation, scaling, storage and network connectivity. CloudProphet [9] is a tool to predict the end-to-end response time of an on-premise web application when migrated to the cloud. CloudSuite [6] is a benchmark suite for emerging scale-out workloads. CloudRank-D [11] is a benchmark suite for benchmarking and ranking the performance of cloud computing systems hosting big data applications. SkyMark [7] is a tool that provides support for micro performance benchmarking in the context of multi-job workloads based on the MapReduce model. The Cloud Architecture Runtime Evaluation (CARE) framework [10] evaluates cloud platforms by using a number of pre-built, pre-configured and reconfigurable components for conducting performance evaluations across different target platforms.

There are three key features that differentiate Smart CloudBench from the other cloud performance benchmarking tools. The first feature is *real-time benchmarking* - users can conduct live, real-time benchmarking of selected cloud providers and servers (they can also make use of historical benchmark results). The second feature is the ability to *customize workloads*. Users are not restricted

4 http://cloudharmony.com/benchmarks
5 http://www.cloudspectator.com/
6 http://serverbear.com/
7 http://www.cedexis.com/products/radar.html

to pre-defined workloads but can instead specify workloads that are representative of their own in-house workloads making the benchmark results more meaningful and relevant. The third feature is the ability to do *performance baselining*. Users can baseline the performance of cloud servers against a wide range of workloads. This helps them select the cloud configuration and provider with the most appropriate specifications that best meet the user's requirements.

7 Conclusion

Prospective cloud consumers would like to obtain a quick assessment of the price, specification and performance of different IaaS providers before making any migration decisions. While the pricing and specification is public information, the performance of computing infrastructure is unknown. The use of different virtualization technologies by cloud providers impacts the performance of software systems running on top of the their infrastructure. The only way to get a measure of cloud infrastructure performance is by benchmarking software systems on it rather than relying on assumptions based on price and specification. In this paper, we have presented Smart CloudBench, which allows the automated execution of representative benchmarks on different IaaS clouds under representative load conditions to quickly estimate their performance levels. It helps decision-makers make informed decisions about migrating their in-house systems to the cloud by evaluating available options based on their price, specification and performance. Users of Smart CloudBench can design different types of experiments to test the performance of representative applications using load conditions that match the load levels of their own in-house applications. Smart CloudBench is particularly useful for organizations that do not possess the time, resources and in-house expertise to do a thorough evaluation of multiple cloud platforms. Tests conducted using it show that higher price does not necessarily translate to better or more consistent performance and highlight the need for tools such as Smart CloudBench to provide greater visibility into cloud infrastructure performance and to aid in the cloud migration decision-making process.

Acknowledgment. This work was partially funded by the Service Delivery and Aggregation Project within the Smart Services CRC.

References

1. Baruwal Chhetri, M., Vo, Q.B., Kowalczyk, R., Lan Do, C.: Cloud broker: helping you buy better. In: Bouguettaya, A., Hauswirth, M., Liu, L. (eds.) WISE 2011. LNCS, vol. 6997, pp. 341–342. Springer, Heidelberg (2011)
2. Baruwal Chhetri, M., Chichin, S., Vo, Q.B., Kowalczyk, R.: Smart CloudBench: automated performance benchmarking of the cloud. In: Proceedings of CLOUD 2013, 27 June–2 July 2013, California (2013)

3. Baruwal Chhetri, M., Vo, Q.B., Kowalczyk, R.: Policy-based automation of SLA establishment for cloud computing services. In: Proceedings of the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID-12), Ottawa (Canada), 13–16 May 2012
4. Baruwal Chhetri, M., Vo, Q.B., Kowalczyk, R.: AutoSLAM - a policy-driven middleware for automated SLA establishment in SOA environments. In: Proceedings of the 9th International Conference on Service Computing (SCC 2012), Honolulu, Hawaii, USA, pp. 9–16. IEEE (2012)
5. Transaction Processing Performance Council. TPC Benchmark W (Web Commerce) Specification, version 2.0r. Technical Specification (2003). http://www.tpc.org/tpcw/spec/TPCWV2.pdf
6. Ferdman, M., et al.: Clearing the clouds: a study of emerging scale-out workloads on modern hardware. In: The 17th International Conference on Architectural Support for Programming Languages and Operating Systems (2012)
7. Iosup, A., Prodan, R., Epema, D.: IaaS cloud benchmarking: approaches, challenges, and experience. In: Proceedings of 5th Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS) (2012)
8. Li, A., Yang, X., Kandula, S., Zhang, M.: CloudCmp: comparing public cloud providers. In: Proceedings of the 10th Annual Conference on Internet Measurement, November 2010
9. Li, A., Yang, X., Kandula, S., Yang, X., Zhang, M.: CloudProphet: towards application performance prediction in cloud. In: Proceedings of ACM SIGCOMM 2011, Toronto, pp 426–427 (2011)
10. Zhao, L., Liu, A., Keung, J.: Evaluating cloud platform architecture with the care framework. In: Proceedings of 17th APSEC, pp. 60–69 (2010)
11. Luo, C., et al.: CloudRank-D: benchmarking and ranking cloud computing systems for data processing applications. Front. Comput. Sci. **6**(4), 347–362 (2012)