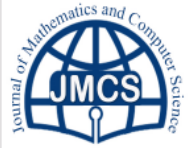




Online: ISSN 2008-949x

Journal of Mathematics and Computer Science



Journal Homepage: www.tjmcs.com - www.isr-publications.com/jmcs

A price-performance analysis of EC2, Google Compute and Rackspace cloud providers for scientific computing

Saeed Ullah*, M. Daud Awan, M. Sikandar Hayat Khiyal

Faculty of Computer Science, Preston University, Islamabad, Pakistan.

Abstract

One of the recently emerging areas in cloud computing is deployment of virtual machines across multiple clouds based on providers' ranking. This involves benchmarking of different cloud providers, development of different techniques for selection of candidate providers and frameworks for ranking cloud providers. Existing benchmarking studies are mostly focused on selection of best-fit cloud provider among a set of cloud providers for a particular set of quality attributes based on industry best standard tools and techniques. However, most of the researches are focused on performance of IaaS cloud providers and price-performance analysis is normally ignored while benchmarking IaaS metrics. In this work, we propose a novel QoS based ranking methodology along with price-performance analysis that can be used as an input for selecting candidate cloud providers. Our proposed mechanism allows cloud consumers to find the most cost effective virtual machines for a given set of user preferences. As a case study, we present performance evaluation and benchmarking results of three major cloud providers: Google, Amazon and Rackspace. ©2016 All rights reserved.

Keywords: Cloud Computing, infrastructure-as-a-service, benchmarking, price-performance analysis, ranking

2010 MSC: 68M14, 68M20.

1. Introduction

Cloud computing is an evolving platform that has grasped the attention of scientific community and business industry towards the provisioning of computing resources as a utility and software as

*Corresponding author

Email addresses: saeedullah@gmail.com (Saeed Ullah), drdaudawan@hotmail.com (M. Daud Awan), m.sikandarhayat@yahoo.com (M. Sikandar Hayat Khiyal)

Received 2016-01-15

a service over a network [4]. The platform of cloud computing shares many similar characteristics of Grids and clusters but it has its own special attributes and capabilities such as strong support for virtualization, dynamically composable services with Web Service interfaces, and strong support for creating 3rd party value added services by building on Cloud compute, storage, and application services. Thus, Clouds are promising to provide services based on pay as you go model to users without reference to the infrastructure on which these are hosted. The success behind the cloud lies in the concept of virtualization where physical resources are further divided into hundreds and even thousands of virtualized devices. Thus, serving more and more consumers the capacity, computation and storage without any conflict unlike the grid.

In cloud, provisioning of computing resources is offered in the form of virtual machines (VMs), being deployed on physical computing nodes on pay-per use pricing policy. Cloud providers specify their offerings to the clients on hourly, monthly, semiannual and annual basis with different performance indicators measured by themselves. Mostly these indicators don't provide comprehensive information about overall performance of virtual machines. For instance, a 4 GB standard instance from Rackspace is offered with 2 vCPU (weighted based on the size of the server), 4 GB RAM, 400 Mb/s network, 160 GB system disk with good disk I/O. If we are interested in quantitative analysis of cloud performance and associated monetary cost, we might also look for alternative cloud providers with similar VM specification. A much similar instance of Amazon t2.medium is offered with 2 vCPU, 4 GB memory with low to moderate network performance. For choosing a best-fit cloud provider, the user needs a good understanding of mapping from low, moderate and good in terms of disk I/O and network performance. However, because of the over-provisioning of cloud resources and underlying hardware, these are not necessarily the decisive indicators [10, 19].

In cloud computing marketplace, multiple cloud providers are available with the same functions but with different QoS attributes. This may result beneficial in terms of better QoS, available choices for low price resource selection and introduction of interactive and creative services. However, it may pose the issue of vendor selection as cloud market is growing at rapid pace. The characterization and evaluation of cloud providers is the first step to understand what instances are more appropriate for a particular set of applications. Providing a benchmark for investigating fair performance assessment of implementation decisions under different deployment environments is necessary to compare current variety of cloud providers. At minimum, we need to understand what a *low* network performance means in terms of network bandwidth and other IaaS related metrics to estimate how long it might take to transfer data to/ across a cloud provider [10].

Different industry standard tools are available to benchmark physical computing machines. However, these tools are not widely used to benchmark virtual machines, being deployed on IaaS platforms. Mostly, cloud benchmarking data is available through commercial websites but these websites provide no details about the parameters used or whether these are for a single machine over time, averages across several, or best or worst values [10, 15, 16, 19]. Very few studies have been conducted on performance benchmarking of virtual machines but to the best of our knowledge, the core issue of cost involved in such evaluation is yet to be addressed in this area of research.

The aim of this present study is to improve the decision accuracy while choosing a cloud provider for a given set of user preferences. The evaluation process helps to measure instance configuration in an objective way to find the most suitable ones based on benchmarking to compare the actual performance in contrast to the stated QoS which may vary over time. In this paper, we present performance evaluation and benchmarking results of three major cloud providers: Google, Amazon and Rackspace. These providers are selected since they are the most popular Infrastructure-as-Service (IaaS) cloud providers. In particular, we are interested to address the following two research questions: Can industry standard benchmark tools be used to compare and evaluate cloud services

for a particular set of quality attributes in contrast to the stated QoS by cloud providers? and are these performance results consistent with incurred cost (VM cost per hour)? The second research question is very significant and needs to be addressed as scientific applications are implemented as multi-threaded heterogeneous distributed programs or meta-computations, designed to run in parallel [12]. Cloud computing is a promising platform for large-scale scientific applications where hundreds or thousands of computational resources, in the form of virtual machines, can be leased on the fly. These virtual machines can be inter-connected by using Message Passing Interface (MPI), Parallel Virtual Machine (PVM), OpenMP etc. Since cost and deadline are two key considerations in such an environment [21, 29], the analysis of price vs. performance of different cloud providers is critical to allow scientific community to find the most cost effective virtual machines that fit their application needs as well as reduce overall cost.

This paper is organized as follows: Section 2 reviews the literature for relevant benchmarking studies. Section 3 formulates our research methodology. Section 4 presents results and discussion. Finally, we present conclusion and possible future directions.

2. Related Work

Benchmarking is the process of measuring services and products based on performance metrics to industry best practices and standards [1, 5, 30]. It can be measured based on certain indicators resulting in a metric form performance that can be comparable to other processes and products of similar nature. Cloud vendors offer heterogeneous types of resources i.e., computational, storage and network services etc. with different level of quality of service. Before ranking a particular cloud provider, it is necessary to benchmark its performance based on industry best standards to compare the actual performance in contrast to the stated QoS which may vary over time. Different benchmarking techniques and methodologies have been proposed in the literature.

Garg et al. [9] opined to evaluate cloud providers in an objective way to find the most suitable ones. The proposed framework SMICloud, based on ISO SMI [17], is a cloud services evaluation framework that addresses the issue of cloud provider selection based on benchmarking of services and feedback from user experiences. ISO SMI KPIs are used as an assessment tool to compare and evaluate cloud services. The novelty of their work is the classification of requirements where users can specify requirements as essential or non-essential. By using different QoS metrics, they addressed the issues of measurement of SMI attributes and ranking of cloud providers based on these attributes. As a case study, different cloud vendors were compared using Analytic Hierarchy Process (AHP) by investigating performance heuristics from past researches. The evaluation data was collected from the studies conducted earlier. Based on these measures, cloud vendors were ranked according to services they provide.

Migration of legacy systems to current state of the art is a challenging issue for industry practitioners and enterprise developers. CloudCmp [20] is a framework aimed to estimate the cost and performance of legacy systems while porting and deploying it over the cloud. Three phases were involved in this process: In the service benchmarking phase, six cloud vendors (including Google AppEngine, Amazon AWS, Microsoft Azure, GoGrid and Rackspace) were selected considering the web application deployment features necessary for cloud computing services. Four types of service benchmarking were applied including elastic compute cluster, persistent storage, intra-cloud networking and wide-area delivery networking. Every cloud provider was assessed for performance and associated cost by running a collection of benchmarking test application. Since portability was a key concern in this environment, cloud services were tested using SPECjvm2008 Java benchmarking toolkit. The performance of individual services was tested by measuring starting and finishing time

while cost effectiveness was measured by the cost per task. Server response time was measured by the time a VM was requested and when the VM was provisioned. However, this metric is limited considering the fact that not all services allow scaling request in this way.

Z. Zheng et al. [38] argued to measure qualitative values of services before any comparison is being made. They ranked different services based on prediction of qualitative values. In traditional model of evaluation of service components on stand-alone systems, different service calls can be invoked to predict the overall performance; however, this may not be a case with cloud computing which involves different layers of abstraction between a service call and the response. Measuring performance at server end could be a reasonable solution to calculate performance values for different parameters which may be close to what providers normally claim. However, the case may vary with qualitative values as these values are based on user preferences and evaluation of such parameters can be tested from client side. The proposed approach, CloudRank, applies qualitative values of service prediction from client side. There are two types of users in the proposed design; normal users with different requirement levels and active users who rank the system based on certain parameters. A data repository is used to store user information along with suggested ranking of services. The system comprises of three components: a similarity computation measures similarity values for all active users based on comparison from data repository. The classification component clusters users based on obtained similarity values. Finally, the ranking component executes ranking algorithm and shows the final results to the user.

CloudStone [28] is a UC Berkeley open source project aimed at characterizing workload of social network websites using web 2.0 applications. The goal is to provide a benchmark for investigating fair performance assessment of implementation decisions under different deployment environments. The framework offers various AMI's (virtual machine image files compatible with Amazon's Elastic Compute Cloud) to facilitate the workflow. Cloudstone is based on three components: Olio, a work load generator and social event calendar that may support thousands of online users. A collection of open source tools for database and metric formulation for experimenting Olio on Amazon EC2 and a set of parameters for calculating the cost per month of applications under deployment.

Considering our current study scope, none of the above benchmarking frameworks covers all aspect of IaaS benchmarking metrics as well as price-performance analysis and hence this study is undertaken with aim to cover all metrics of IaaS price-performance benchmarking which is a first step in this direction.

3. Materials and methods

Based on the literature, we evaluated the three IaaS cloud providers using the following categories of metrics: Compute, Memory Hierarchy, Network and Storage. In this study, we have added few more metrics as our model differs from those, proposed earlier [5, 10, 19, 35]:

Compute: Speed of VM (CPU), CPU cores, execution time;

Memory Hierarchy: Memory read/modify/write access time;

Network: available bandwidth, latency, throughput;

Storage: input/output operations per second, average access time, throughput, latency.

Most of the cloud vendors guarantee performance of compute metrics at hypervisor level and these metrics are not reflected in service level agreements. On the other hand, network metrics are mostly reflected in SLA at datacenter level which are not specific to any particular user or consumer [23].

Since simulation results and data extraction from the studies conducted earlier do not present the actual figures to benchmark a particular cloud vendor, so in this study, we evaluated different cloud vendors on the real testbed experiments. Our experiments were conducted on different 64-bit Linux VM instances, given in Table 1, leased from the three cloud providers. Necessary statistical data for benchmarking was collected after a series of experiments conducted in a time span of one month. Multiple datacenters of the three cloud providers were used to calculate average performance estimation. However, an exception was Rackspace North Virginia datacenter which was not used in these experiments due to its poor network performance.

Table 1: List of VM instances used for experimentation

| Ref. ID | Cloud Provider | VM instance | vCPU | Memory (GB) | Cost (US\$) | Storage (GB) |
|---------|----------------|------------------|------|-------------|-------------|--------------|
| A4GB | Amazon | t2.medium | 2 | 4 | 0.052 | EBS |
| A15GB | Amazon | m3.xlarge | 4 | 15 | 0.266 | 2 x 40 SSD |
| RS 4GB | Rackspace | 4GB Standard | 2 | 4 | \$0.24 | 160 RAID |
| R7.5GB | Rackspace | 7.5GB Compute v1 | 4 | 7.5 | \$0.23 | 50 ESB |
| G3.8GB | Google | n1-standard-1 | 1 | 3.75 | \$0.050 | 10240 SSD |
| G13GB | Google | n1-highmem-2 | 2 | 13 | \$0.126 | 10240 SSD |

Tables 2–5 show the list of benchmarking tools used in our experiments:

Table 2: Network benchmarking tools

| Tool | Supported Metrics |
|--------------------|-------------------------------|
| iperf [18] | Network bandwidth, throughput |
| ping | Network latency |
| speedtest-cli [24] | Network bandwidth |

Table 3: CPU benchmarking tools

| Tool | Supported Metrics |
|---------------------------|--------------------|
| Dacapo [32] | CPU execution time |
| SPECjvm2008 [36] | CPU execution time |
| Phoronix Test Suites [22] | CPU execution time |

Table 4: Storage benchmarking tools

| Tool | Supported Metrics |
|---------------|---------------------------|
| iozone [34] | Throughput, Disk I/O |
| bonnie++ [31] | Storage latency, Disk I/O |

Table 5: Memory benchmarking tools

| Tool | Supported Metrics |
|-----------------|--------------------------------------|
| Cachebench [33] | Memory read/modify/write access time |
| Ubench [14] | Memory throughput |

3.1. Ranking Methodology

We consider a system model which consists of n cloud providers. Each cloud provider offers m type of VM instances, each of which offers different CPU cores, disk storage and memory size. To compare performance of VMs, we need to assign weights to each top level evaluation metric (W_j) and associated benchmarking tools (w_j) taking into account their relative importance of each benchmarking experiment j . Each VM instance is compared term by term correspondingly for k set of benchmarking tools for evaluation of benchmarking metrics. We compute the performance score for each benchmarking experiment j . However, since a benchmarking tool may consist of l number of benchmarking test cases, the performances score (P_j) for each of the experiment is calculated as follows:

$$P_j = \sum_{b=1}^l \text{GeometricMean}(\text{test_case_score}). \quad (3.1)$$

The price factor should also be incorporated in Eq. (3.1) as a VM with high utilization cost will usually provide better performance results so performance comparison should be made on equal ground. Hence, we introduce ρ , the price-performance ratio, so that VMs with different configuration and pricing schemes may be accurately compared:

$$\rho_j = P_j / C, \quad (3.2)$$

where C represents the VM cost, given in Table 1. Finally, overall score for a particular benchmarking experiment is computed as follows:

$$(\alpha_j \times \rho_j) + (100 - \alpha_j) \times P_j, \quad (3.3)$$

where α_j modifier is the sensitive factor of price-performance analysis and may vary for different experiments. For each benchmark, the utility values of VM instance v_{ij} are computed that depicts the performance of a particular VM with respect to a particular benchmarking result. The utility value is normalized so that $\sum_{j=1}^k v_{ij} = 1$ and $v_{ij} \in [0, 1]$. The overall aggregation of utility values V_{ij} is then calculated as a sum of all v_{ij} for every evaluation metric. Finally, the utility function $f(V_{ij})$ associated with VM instance i is constructed as follows:

$$f(V_{ij}) = \sum_{j=1}^k v_{ij} * W_j. \quad (3.4)$$

VM instances with higher value of utility functions are ranked higher as compared to instances with lower score.

4. Results and Discussions

In this section, we present our benchmarking results based on the four categories of metrics discussed in earlier sections.

4.1. CPU Benchmarking

CPU Processing is a common performance benchmark for applications involving intensive workload. Processing performance is much dependant on clock speed, number of CPU cores and type of hardware. It is generally believed that VM instances with large number of CPU cores provide much better performance in terms of server execution and response time. However, our experimentation

results somehow depict that no significant performance gain was observed between VM instances with 2 vCPU and 4 vCPU compute instances. A maximum of 10-15% of performance variation was found between switching small instances to comparatively better server grade hardware. For instance, Decapo CPU benchmarking resulted in only 8% of performance gain while switching from Amazon t2.medium to Amazon m3.xlarge. Figures 1–3 present benchmarking results of Decapo, SPECJVM and Phoenix test suites. Experimentation results demonstrate that Amazon m3.xlarge outperforms all other instances in terms of CPU performance.

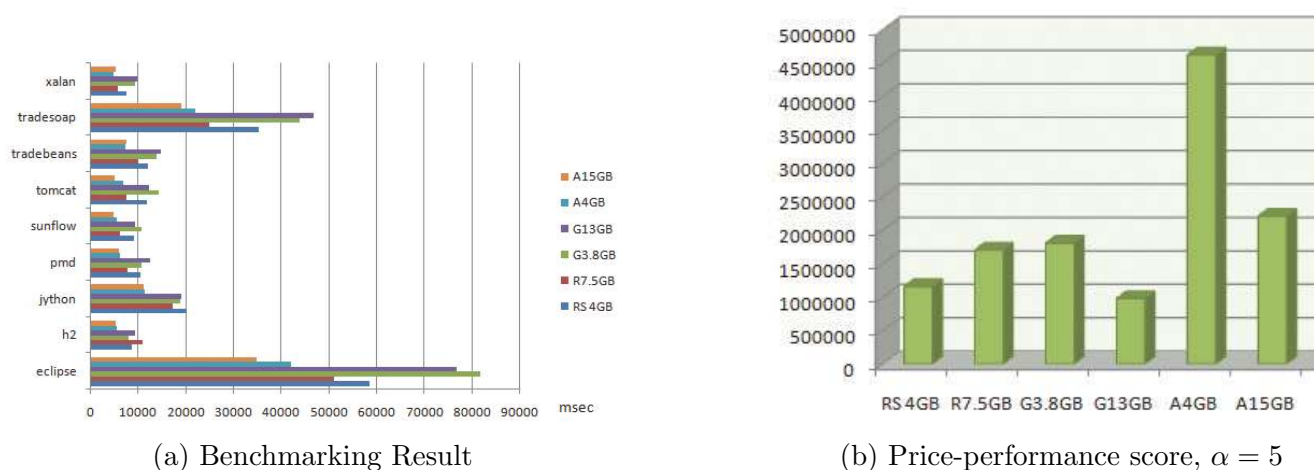


Figure 1: Benchmarking result of Decapo test suite

A15GB outperformed other VM instances in terms of performance (as shown in Figure 1a) but with relatively higher cost as compared to A4GB. However, the performance results of A4GB are much more promising than other VM instances when the price vs. performance score is taken into consideration (shown in Figure 1b). Scientific applications, due to their inherently multi-threaded distributed nature, can significantly reduce computational cost by leasing such inexpensive VM instances instead of investing in relatively expensive VM instances with partial gain in overall performance.

The price-performance comparison of all VMs, based on Figure 1b, is given in Table 6.

Table 6: Decapo price-performance analysis ($\alpha = 5$)

| VM | RS 4GB | R7.5GB | G3.8GB | G13GB | A4GB | A15GB |
|---------------|--------|--------|--------|--------|-------|-------|
| RS 4GB | 1:1 | 1.5:1 | 1.6:1 | 1: 1.2 | 4:1 | 1.9:1 |
| R7.5GB | 1.5:1 | 1:1 | 1.1:1 | 1:1.8 | 2.7:1 | 1.3:1 |
| G3.8GB | 1:1.6 | 1:1.1 | 1:1 | 1:1.9 | 2.6:1 | 1.2:1 |
| G13GB | 1.2:1 | 1.8:1 | 1.9:1 | 1:1 | 4.8:1 | 2.3:1 |
| A4GB | 1:4 | 1:2.7 | 1:2.6 | 1:4.8 | 1:1 | 1:2.1 |
| A15GB | 1:1.9 | 1:1.3 | 1:1.2 | 1:2.3 | 2.1:1 | 1:1 |

In Table 6, each score is a ratio between two cloud providers. For instance, the score 1.5:1 of second column in first row shows that RS 4GB performance result is 1.5 times lower than R7.5GB. The most extreme ratio was between A4GB and G13GB with 1:4.8.

Scientific applications are inherently CPU and memory intensive and they often push computing resources to its full potential [8]. The sub-benchmarks of SPECjvm2008 are components of real world applications intended to test overall performance of modern CPU architecture and memory sub-system. Figure 2 depicts benchmarking results in ops/min (operations per minute) of three different test cases: Compiler for throughput measurement to reflect overall performance of the system, SciMark workload to test JVM code optimization targeted at L2 cache and memory subsystem and Crypto sub-benchmark with AES workload for performance testing of encryption and decryption protocols. The most extreme ratio, as shown in Table 7, was between A4GB and RS 4GB with 1:5.3.

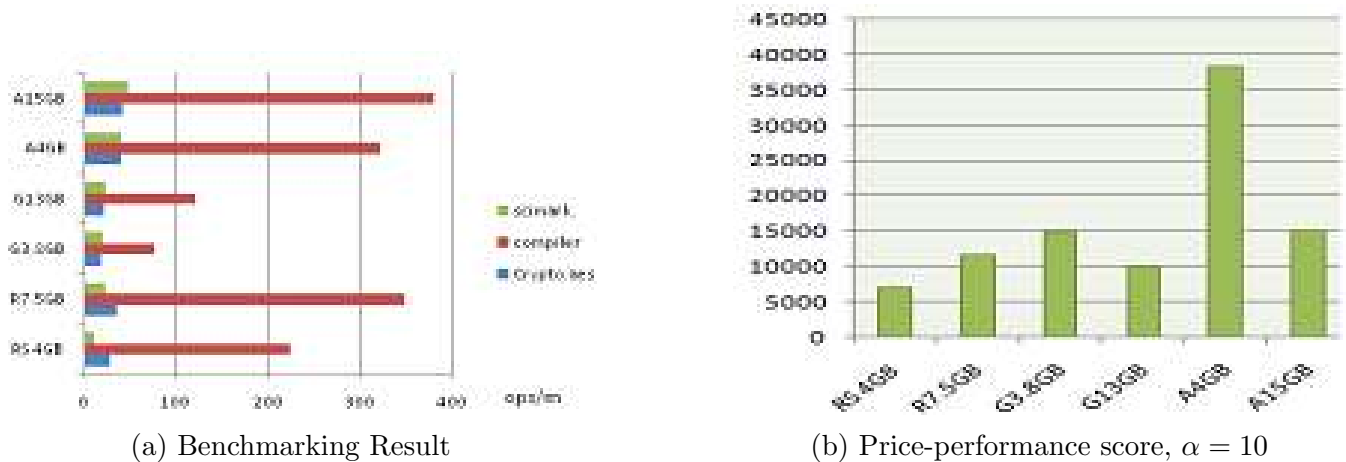
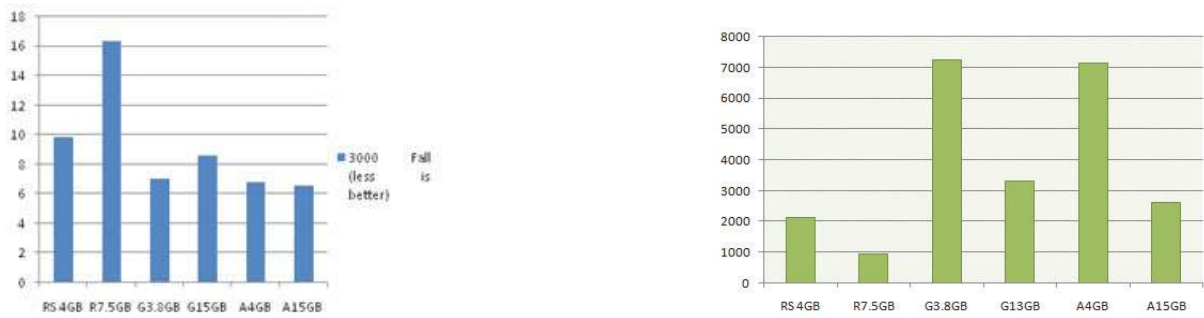


Figure 2: Performance Classifications of three SPECJVM test cases

Table 7: SPECJVM price-performance analysis ($\alpha = 10$)

| VM | RS 4GB | R7.5GB | G3.8GB | G13GB | A4GB | A15GB |
|---------------|--------|--------|--------|-------|-------|-------|
| RS 4GB | 1:1 | 1.6:1 | 2.1:1 | 1.4:1 | 5:1 | 2.1:1 |
| R7.5GB | 1:1.6 | 1:1 | 1.3:1 | 1:1.2 | 3.2:1 | 1.3:1 |
| G3.8GB | 1:2.1 | 1:1.3 | 1:1 | 1:1.5 | 2.6:1 | 1:1 |
| G13GB | 1:1.4 | 1.2:1 | 1.5:1 | 1:1 | 3.8:1 | 1.5:1 |
| A4GB | 1:5.3 | 1:3.2 | 1:2.6 | 1:3.8 | 1:1 | 1:2.5 |
| A15GB | 1:2.1 | 1:1.3 | 1:1 | 1:1.5 | 2.5:1 | 1:1 |

The Phoronix Test Suite is built for benchmarking real world applications and hardware performance comparisons including processor, system, memory and graphic performance. Bullet is a physics engine to simulate virtual environment that incorporates laws from the physical world. Figure 3 presents the experimentation results of 3000 fall bullet physics engine. The score of such experiments are inverted as VMs with better hardware configuration complete the simulation in relatively lesser time. The results are interesting and somehow differ from the earlier two CPU experiments as G3.8GB performed slightly better than A4GB in price-performance analysis. As shown in Table 8, the worst ratio was G3.8GB vs. R7.5GB with a performance variation of 1:7.7.



(a) Benchmarking Result of 3000 fall bullet Phoronix test suite

(b) Price-performance score, $\alpha = 2$

Figure 3: Performance classifications of three Phoronix test cases

Table 8: Phoronix price-performance analysis ($\alpha = 2$)

| VM | RS 4GB | R7.5GB | G3.8GB | G13GB | A4GB | A15GB |
|---------------|--------|--------|--------|-------|-------|-------|
| RS 4GB | 1:1 | 1:2.2 | 3.4:1 | 1.6:1 | 3.4:1 | 1.2:1 |
| R7.5GB | 2.2:1 | 1:1 | 7.7:1 | 3.5:1 | 7.6:1 | 2.8:1 |
| G3.8GB | 1:3.4 | 1:7.7 | 1:1 | 1:2.2 | 1:1 | 1:2.8 |
| G13GB | 1:1.6 | 1:3.5 | 2.2:1 | 1:1 | 2.2:1 | 1:1.3 |
| A4GB | 1:3.4 | 1:7.6 | 1:1 | 1:2.2 | 1:1 | 1:2.8 |
| A15GB | 1:1.2 | 1:2.8 | 2.8:1 | 1.3:1 | 2.8:1 | 1:1 |

4.2. Network Benchmarking

Network performance determines how fast a VM can communicate with other clients over a network. Unlike most cloud storage and database applications, which are comparatively more tolerant to delay and jitter, media applications are far more demanding. The network delay with large jitter degrades user received media quality of latency sensitive multimedia applications [3, 6]. Hence, cloud providers with better network performance in terms of throughput vs. latency play a key role in the success of such business applications. We evaluated the network performance of the three cloud providers using iperf, speedtest-cli and ping. The results are depicted in Figure 4.



(a) Benchmarking Result

(b) Price-performance score, $\alpha = 5$

Figure 4: Network performance of VM Instances

Table 9: Network price-performance analysis ($\alpha = 5$)

(a) Download analysis

| VM | RS 4GB | R7.5GB | G3.8GB | G13GB | A4GB | A15GB |
|--------|--------|--------|--------|-------|-------|-------|
| RS 4GB | 1:1 | 1.4:1 | 1:3.4 | 1:2.3 | 1.5:1 | 1.4:1 |
| R7.5GB | 1:1.4 | 1:1 | 1:4.8 | 1:3.3 | 1:1 | 1:1 |
| G3.8GB | 3.4:1 | 4.8:1 | 1:1 | 1.5:1 | 5:1 | 5:1 |
| G13GB | 2.3:1 | 3.3:1 | 1:1.5 | 1:1 | 3.4:1 | 3.3:1 |
| A4GB | 1:1.5 | 1:1 | 1:5 | 1:3.4 | 1:1 | 1:1 |
| A15GB | 1:1.4 | 1:1 | 1:4.9 | 1:3.3 | 1:1 | 1:1 |

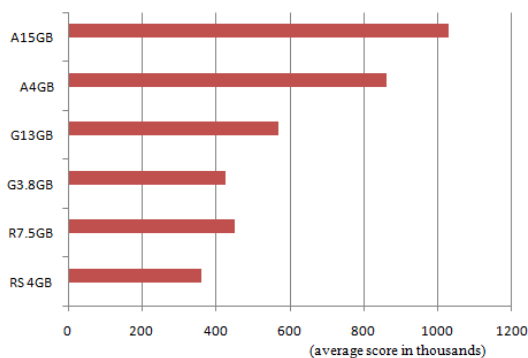
(b) Upload analysis

| VM | RS 4GB | R7.5GB | G3.8GB | G13GB | A4GB | A15GB |
|--------|--------|--------|--------|-------|-------|-------|
| RS 4GB | 1:1 | 1.2:1 | 1:3.1 | 1:6.1 | 1.6:1 | 1.1:1 |
| R7.5GB | 1:1.2 | 1:1 | 1:3.8 | 1:7.6 | 1.3:1 | 1:1.2 |
| G3.8GB | 3.1:1 | 3.8:1 | 1:1 | 1:2 | 5:1 | 3.3:1 |
| G13GB | 6.1:1 | 7.6:1 | 2:1 | 1:1 | 10:1 | 6.6:1 |
| A4GB | 1:1.6 | 1:1.3 | 1:5 | 1:10 | 1:1 | 1:1.5 |
| A15GB | 1:1.1 | 1.2:1 | 1:3.3 | 1:6.6 | 1.5:1 | 1:1 |

As shown in Table 9, both Amazon and Rackspace provide much better network performance with notably higher download/upload speed. Google VM instances are relatively poor in terms of network performance and latency. The most extreme ratios 1:5, between G3.8GB and A4Gb in case of download and 1:10, between G13GB and A4Gb in case of upload were obtained in our calculation results.

4.3. Memory Benchmarking

Memory is the core element that determines the processing speed of an application. Many scientific and business applications are extremely dependant on memory system performance. Under heavy memory contention, the memory latency may increase two or three times. Thus, the increasing performance gap between processors and memory systems imposes a memory bottleneck for such applications [13]. We evaluated the three cloud providers using Ubench and Cachebench benchmarking tools. Experimentation results are presented in Figures 5 and 6. The results suggest that Google and Amazon VM instances are well optimized for memory related operations. However, as summarized in Table 10, Rackspace instances are far below in the memory related functions and operations. Likewise CPU benchmarking, no major performance degradation was observed between VM instances with less number of cores and memory.



(a) Benchmarking Result

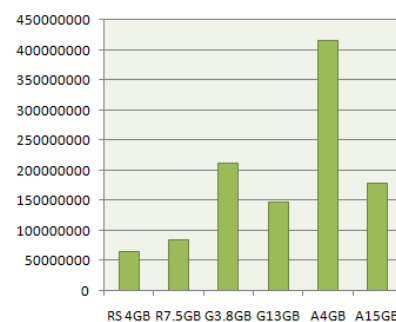
(b) Price-performance score, $\alpha = 2$

Figure 5: Ubench memory benchmarking score

Table 10: Ubench price-performance analysis ($\alpha = 2$)

| VM | RS 4GB | R7.5GB | G3.8GB | G13GB | A4GB | A15GB |
|--------|--------|--------|--------|-------|-------|-------|
| RS 4GB | 1:1 | 1.3:1 | 3.2:1 | 2.2:1 | 6.3:1 | 2.7:1 |
| R7.5GB | 1:1.25 | 1:1 | 2.6:1 | 1.8:1 | 5:1 | 2.2:1 |
| G3.8GB | 1:3.2 | 1:2.6 | 1:1 | 1:1.6 | 2:1 | 1:1.2 |
| G13GB | 1:2.2 | 1:1.8 | 1.5:1 | 1:1 | 2.9:1 | 1.2:1 |
| A4GB | 1:6.3 | 1:5 | 1:2 | 1:2.9 | 1:1 | 1:2.3 |
| A15GB | 1:2.7 | 1:2.2 | 1.2:1 | 1:1.2 | 2.3:1 | 1:1 |

Cachebench experiments were performed for benchmarking memory system performance using read, write, rmw, handread, handwrite, handrmw, memset and memcpy operations. From Figure 6, it is apparent that Amazon and Google VM instances have very identical memory performance while the performance of Rackspace VM instances was degraded during the memory related operations.

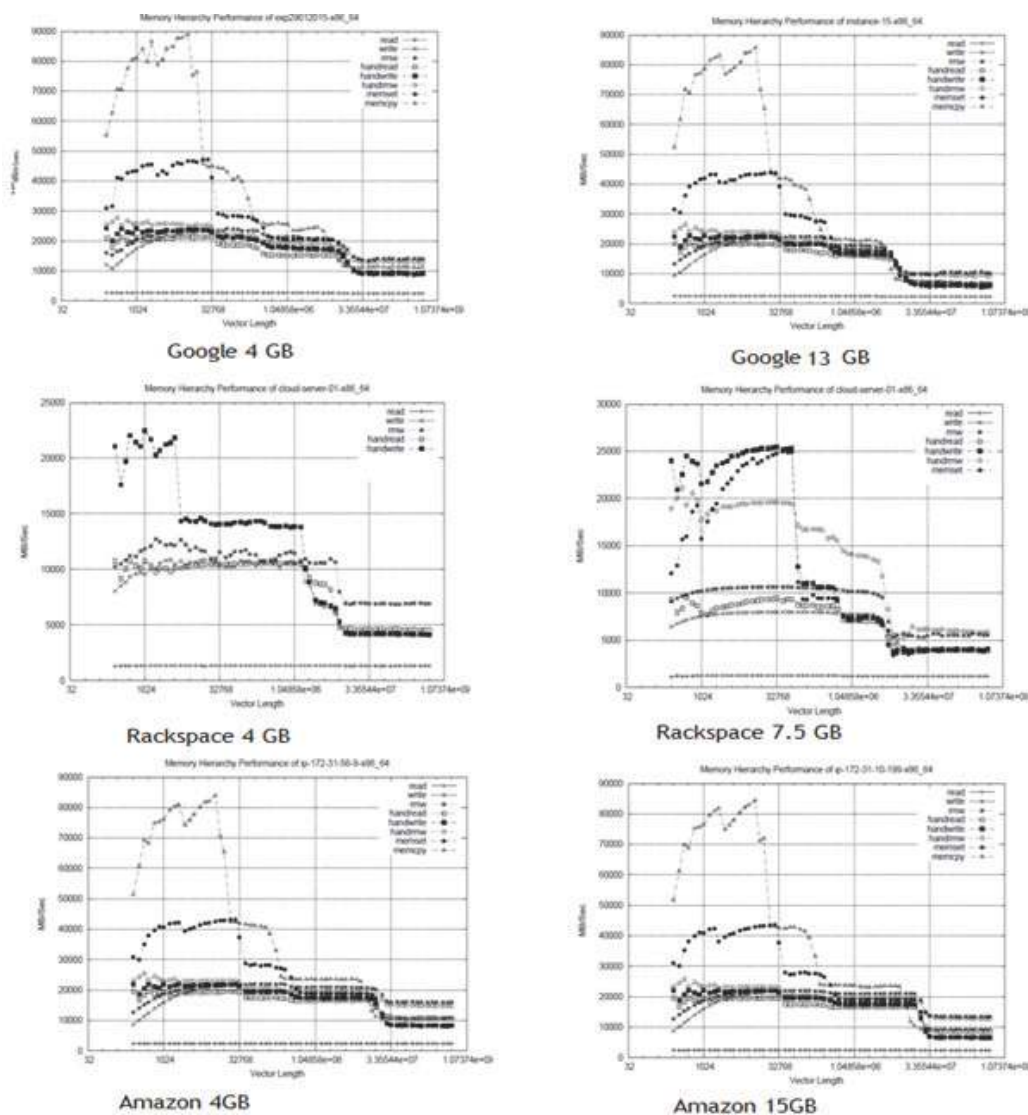


Figure 6: Cachebench results of three cloud providers

4.4. Storage Benchmarking

Considering the nature of cloud computing environment where scientific applications and business workflows continuously involve storage I/O operations, it is important to determine how quickly storage devices allow applications to interact with data or files. Experimentations results, as shown in Figure 7, reveal a steady IO performance for both Rackspace and Amazon, however, performance variation of Google is a matter of discussion for future studies. Our second sets of experiments with Bonnie suggest that Rackspace VM instances outperformed Amazon and Google. Furthermore, the two Google VM instances were failed to complete IO operations for file size greater than 4 GB. It clearly shows that such instances are not built for data intensive cloud applications.

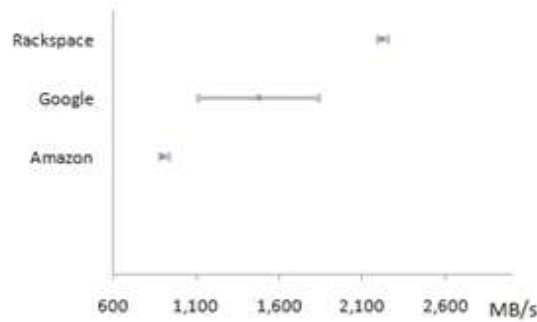


Figure 7: IOZone benchmarking results of 4GB VM instances

For I/O benchmarking through bonnie++, we provided data size double of the size of RAM. However, an exception was Google13GB instance which was failed on such data size and we had to restrict the experiment using 8 GB data size. As presented in figure 8, Rackspace VM instances provided better performance results while Amazon instances usage for CPU was lesser during the same operations. Google VM instances were ranked lower in all cases.

| Sequential Output | | | | | | Sequential Input | | | | | | Random Seeks | |
|---|-------|---------|------|----------|-------|------------------|-------|---------|------|----------|-------|--------------|------|
| Per Char | Block | Rewrite | | Per Char | Block | Per Char | Block | Rewrite | | Per Char | Block | Seeks | |
| E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | MB | %CPU |
| | | 226745 | 75 | 100036 | 26 | | | 314383 | 37 | 1075 | 25 | | |
| Latency | | 325ms | | 279ms | | | | 79900ms | | 71644ms | | | |
| Rackspace 4GB | | | | | | | | | | | | | |
| Sequential Output | | | | | | Sequential Input | | | | | | Random Seeks | |
| Per Char | Block | Rewrite | | Per Char | Block | Per Char | Block | Rewrite | | Per Char | Block | Seeks | |
| E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | MB | %CPU |
| | | 55444 | 4 | 37811 | 2 | | | 78127 | 2 | 4202 | 39 | | |
| Latency | | 436ms | | 508ms | | | | 140ms | | 10823ms | | | |
| Amazon 4GB | | | | | | | | | | | | | |
| Sequential Output | | | | | | Sequential Input | | | | | | Random Seeks | |
| Per Char | Block | Rewrite | | Per Char | Block | Per Char | Block | Rewrite | | Per Char | Block | Seeks | |
| E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | MB | %CPU |
| | | 73435 | 10 | 43280 | 4 | | | 152284 | 6 | 498 | 7 | | |
| Latency | | 723ms | | 1030ms | | | | 36906ms | | 58684ms | | | |
| Google 4 GB | | | | | | | | | | | | | |
| Sequential Output | | | | | | Sequential Input | | | | | | Random Seeks | |
| Per Char | Block | Rewrite | | Per Char | Block | Per Char | Block | Rewrite | | Per Char | Block | Seeks | |
| E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | MB | %CPU |
| | | 104997 | 8 | 48108 | 3 | | | 77852 | 2 | 4379 | 39 | | |
| Latency | | 220ms | | 5413ms | | | | 154ms | | 7730ms | | | |
| Rackspace 7.5 GB | | | | | | | | | | | | | |
| Sequential Output | | | | | | Sequential Input | | | | | | Random Seeks | |
| Per Char | Block | Rewrite | | Per Char | Block | Per Char | Block | Rewrite | | Per Char | Block | Seeks | |
| E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | MB | %CPU |
| | | 201343 | 29 | 106881 | 25 | | | 301401 | 30 | 1070 | 19 | | |
| Latency | | 150ms | | 1832ms | | | | 206ms | | 65568ms | | | |
| Amazon 15 GB | | | | | | | | | | | | | |
| Sequential Output | | | | | | Sequential Input | | | | | | Random Seeks | |
| Per Char | Block | Rewrite | | Per Char | Block | Per Char | Block | Rewrite | | Per Char | Block | Seeks | |
| E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | E:MB | %CPU | MB | %CPU |
| | | 90824 | 15 | 91496 | 8 | | | 5667894 | 99 | 5612 | 117 | | |
| Latency | | 445ms | | 433ms | | | | 203ms | | 7619ms | | | |
| Google 13 GB (tested on 8 GB file size) | | | | | | | | | | | | | |

Figure 8: Bonnie file system benchmarking results

4.5. Ranking Cloud Providers for Scientific Computing

For overall price-performance ranking of the three cloud providers, we used the additive weighting scheme, discussed in Section 3.1. Scientific applications are usually hardware dependant and may be compute and I/O intensive [25, 37], compute and bandwidth intensive [7, 27] or compute and memory intensive [2, 11, 39]. Based on the research studies, weights were assigned to each evaluation metric as given in Table 11.

Table 11: List of Metrics and associated weights

| Metric | Top level weight(W_j) | Benchmark (j) | Second level (w_j) |
|---------|---------------------------|---------------|------------------------|
| CPU | 0.4 | Phoronix | 0.4 |
| | | SPECjvm | 0.4 |
| | | dacapo | 0.2 |
| Memory | 0.3 | Cachebench | 0.6 |
| | | Ubench | 0.4 |
| Network | 0.15 | Upload | 0.4 |
| | | Download | 0.4 |
| | | Ping | 0.2 |
| Storage | 0.15 | Bonnie | 0.6 |
| | | IOZone | 0.4 |

For all VMs, the utility value against each benchmark is obtained by multiplying the relative normalized score with the relative second level weight of the benchmark w_j . For every CPU metric, the score is aggregated, resulting in cumulative utility value (V_j). The utility function is then calculated by multiplying top level weight (W_j) by cumulative utility value. Finally, total ranking score is calculated as a sum of utility functions for all evaluation metrics. The ranking scores for all VMs are given in Table 12.

Table 12: List of Metrics and associated weights

| Metric | Benchmark | VM ₁ | v _{1j} | V _{1j} | f(V _{1j}) | VM ₂ | v _{2j} | V _{2j} | f(V _{2j}) | VM ₃ | v _{3j} | V _{3j} | f(V _{3j}) |
|---------|------------|-----------------|-----------------|-----------------|---------------------|-----------------|-----------------|-----------------|---------------------|-----------------|-----------------|-----------------|---------------------|
| CPU | Phoronix | 0.09 | 0.04 | 0.09 | 0.036 | 0.04 | 0.02 | 0.1 | 0.04 | 0.31 | 0.12 | 0.21 | 0.08 |
| | SPECjvm | 0.07 | 0.03 | | | 0.12 | 0.05 | | | 0.15 | 0.06 | | |
| | Dacapo | 0.09 | 0.02 | | | 0.14 | 0.03 | | | 0.14 | 0.03 | | |
| Network | Download | 0.16 | 0.06 | 0.18 | 0.027 | 0.23 | 0.09 | 0.23 | 0.035 | 0.05 | 0.02 | 0.06 | 0.009 |
| | Upload | 0.18 | 0.07 | | | 0.23 | 0.09 | | | 0.06 | 0.02 | | |
| | Ping | 0.23 | 0.05 | | | 0.25 | 0.05 | | | 0.08 | 0.02 | | |
| Memory | Cachebench | 0.09 | 0.05 | 0.07 | 0.021 | 0.07 | 0.04 | 0.07 | 0.021 | 0.28 | 0.17 | 0.24 | 0.072 |
| | UBench | 0.06 | 0.02 | | | 0.08 | 0.03 | | | 0.19 | 0.07 | | |
| Storage | Bonnie | 0.23 | 0.14 | 0.22 | 0.033 | 0.19 | 0.11 | 0.18 | 0.027 | 0.08 | 0.05 | 0.12 | 0.018 |
| | Iozone | 0.21 | 0.08 | | | 0.18 | 0.07 | | | 0.18 | 0.07 | | |
| Total | | | | | 0.117 | | | | 0.123 | | | | 0.179 |

| Metric | Benchmark | VM ₄ | v _{4j} | V _{4j} | f(V _{4j}) | VM ₅ | v _{5j} | V _{5j} | f(V _{5j}) | VM ₆ | v _{6j} | V _{6j} | f(V _{6j}) |
|---------|------------|-----------------|-----------------|-----------------|---------------------|-----------------|-----------------|-----------------|---------------------|-----------------|-----------------|-----------------|---------------------|
| CPU | Phoronix | 0.14 | 0.06 | 0.12 | 0.05 | 0.31 | 0.12 | 0.35 | 0.14 | 0.11 | 0.04 | 0.14 | 0.06 |
| | SPECjvm | 0.10 | 0.04 | | | 0.39 | 0.16 | | | 0.16 | 0.06 | | |
| | dacapo | 0.08 | 0.02 | | | 0.37 | 0.07 | | | 0.18 | 0.04 | | |
| Network | Download | 0.07 | 0.03 | 0.05 | 0.008 | 0.24 | 0.10 | 0.26 | 0.039 | 0.24 | 0.10 | 0.22 | 0.033 |
| | Upload | 0.03 | 0.01 | | | 0.30 | 0.12 | | | 0.20 | 0.08 | | |
| | Ping | 0.03 | 0.01 | | | 0.23 | 0.04 | | | 0.18 | 0.04 | | |
| Memory | Cachebench | 0.16 | 0.10 | 0.15 | 0.045 | 0.26 | 0.16 | 0.31 | 0.093 | 0.15 | 0.09 | 0.15 | 0.045 |
| | UBench | 0.13 | 0.05 | | | 0.38 | 0.15 | | | 0.16 | 0.06 | | |
| Storage | bonnie | 0.06 | 0.04 | 0.06 | 0.009 | 0.25 | 0.15 | 0.24 | 0.036 | 0.18 | 0.11 | 0.18 | 0.027 |
| | iozone | 0.04 | 0.02 | | | 0.22 | 0.09 | | | 0.17 | 0.07 | | |
| Total | | | | | 0.112 | | | | 0.308 | | | | 0.165 |

Where RS 4GB, R7.5GB, G3.8GB, G13GB, A4GB and A15GB are represented as VM1, VM2, . . . , VM6 respectively. The results clearly indicate that A4GB outperformed all other VMs in terms of

price-performance evaluation metrics, followed by G3.8GB, A15GB, R7.5GB, 4GB, and G13GB. Our final results are somehow different from the benchmarking studies conducted earlier [10, 26] as the computational cost, being the driving factor of cloud computing model, was not considered in these studies and hence the performance evaluation results may not be on equal ground.

5. Conclusion & Future Work

Quality of Service is a broad topic in the field of computer networks and distributed systems. Although various QoS standards are available for other fields of computer networks, much research is still needed to define QoS standards and metrics in the field of cloud computing. This study was undertaken to provide detailed insights into VM instance configuration of three major cloud providers. For future work, an automated broker based QoS ranking framework can be integrated with multiple IaaS cloud providers to filter QoS requirements and then calculate providers' ranking based on performance of cloud infrastructure along with previous job experience to select the 'right cloud provider' for a particular set of quality attributes required to satisfy incoming job requirements.

Disclosure Policy

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

- [1] G. Aceto, A. Botta, W. De Donato, A. Pescapè, *Cloud monitoring: A survey*, *Comput. Networks*, **57** (2013), 2093–2115. 2
- [2] R. Bader, M. Brehm, R. Ebner, H. Heller, L. Palm, F. Wagner, *TeraFlops Computing with the Hitachi SR8000-F1: From Vision to Reality*, *High Performance Computing in Science and Engineering*, Springer Berlin Heidelberg, (2002), 3–8. 4.5
- [3] S. K. Barker, P. Shenoy, *Empirical evaluation of latency-sensitive application performance in the cloud*, *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, (2010), 35–46. 4.2
- [4] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, *Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility*, *Future Gener. Comput. Syst.*, **25** (2009), 599–616. 1
- [5] T. Chauhan, S. Chaudhary, V. Kumar, M. Bhise, *Service level agreement parameter matching in cloud computing*, *World Congress on Information and Communication Technologies (WICT)*, (2011), 564–570. 2, 3
- [6] L. Cheng, C. L. Wang, *QoS Scheduling for Latency-sensitive Cloud Applications*, **2011** (2011), 10 pages. 4.2
- [7] CIO WebBusiness, December 1, 1997. 4.5
- [8] A. Fries, J. P. i de Mora, R. Sirvent, *Java-based communication in a High Performance Computing environment*, *EAS Public. Ser.*, **45** (2010), 103–106. 4.1
- [9] S. K. Garg, S. Versteeg, R. Buyya, *A framework for ranking of cloud computing services*, *Future Gener. Comput. Syst.*, **29** (2013), 1012–1023. 2
- [10] L. Gillam, B. Li, J. O'Loughlin, A. P. S. Tomar, *Fair benchmarking for cloud computing systems*, *J. Cloud Comput. Adv. Syst. Appl.*, **2013** (2013), 6 pages. 1, 3, 4.5
- [11] N. K. Govindaraju, S. Larsen, J. Gray, D. Manocha, *A memory model for scientific algorithms on graphics processors*, *Proceedings of the ACM/IEEE SC Conference*, (2006). 4.5
- [12] P. T. Homer, *Constructing Scientific Applications from Heterogeneous Resources*, Ph.D. Dissertation, Technical Report 94-33, Department of Computer Science, University of Arizona, (1994). 1
- [13] C. Hristea, D. Lenoski, J. Keen, *Measuring memory hierarchy performance of cache-coherent multiprocessors using micro benchmarks*, *ACM/IEEE 1997 Conference on Supercomputing*, (1997). 4.3
- [14] https://archive.org/download/tucows_69604_Ubench/ubench-0.32.tar.gz 5
- [15] <http://blog.cloudharmony.com/2010/06/disk-io-benchmarking-in-cloud.html> 1
- [16] <http://cloudspectator.com> 1

- [17] Service Measurement Index Framework Version 2.1 [Online], http://csmic.org/downloads/SML-Overview_TwoPointOne.pdf 2
- [18] <https://iperf.fr> 2
- [19] A. Lenk, M. Menzel, J. Lipsky, S. Tai, P. Offermann, *What are you paying for? Performance benchmarking for infrastructure-as-a-service offerings*, IEEE International Conference on Cloud Computing (CLOUD), (2011), 484–491. 1, 3
- [20] A. Li, X. Yang, S. Kandula, M. Zhang, *CloudCmp: comparing public cloud providers*, Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, (2010), 1–14. 2
- [21] M. Malawski, G. Juve, E. Deelman, J. Nabrzyski, *Cost-and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds*, Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, **22** (2012), 1–11. 1
- [22] Phoronix Test Suite Suites - <http://openbenchmarking.org> 3
- [23] Practical Guide to Cloud Service Level Agreements Version 2.0 [Online], <http://www.cloud-council.org/deliverables/CSCC-Practical-Guide-to-Cloud-Service-Agreements.pdf> 3
- [24] pypi.python.org/pypi/speedtest-cli 2
- [25] D. A. Reed, *Scalable Input/Output: achieving system balance*, The MIT Press, U.S.A., (2003). 4.5
- [26] C. B. Rodamilans, A. Baruchi, E. T. Midorikawa, *Experiences Applying Performance Evaluation to Select a Cloud Provider*, Proceedings of recent advances in computer engineering, communications and information technology, (2014). 4.5
- [27] S. Sindhu, S. A. Mukherjee, *A Dynamic List Scheduling Algorithm for Scheduling HPC Application in a Cloud Environment*, [Online] retrieved from http://searchdl.org/public/book_series/AETS/7/178.pdf. 4.5
- [28] W. Sobel, S. Subramanyam, A. Sucharitakul, J. Nguyen, H. Wong, A. Klepchukov, S. Patil, A. Fox, D. Patterson, *Cloudstone: Multi-platform, multi-language benchmark and measurement tools for web 2.0*, Proceedings of the 1st Workshop on Cloud Computing, (2008). 2
- [29] R. Van den Bossche, K. Vanmechelen, J. Broeckhove, *Cost-optimal scheduling in hybrid IaaS clouds for deadline constrained workloads*, IEEE 3rd International Conference on Cloud Computing (CLOUD), (2010), 228–235. 1
- [30] C. Vazquez, R. Krishnan, E. John, *Cloud Computing Benchmarking: A Survey*, Proceedings of the International Conference on Grid Computing and Applications (GCA), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing, (2014). 2
- [31] www.coker.com.au/bonnie++ 4
- [32] www.dacapobench.org 3
- [33] www.icl.cs.utk.edu/projects/llcbench/cachebench.html 5
- [34] www.iozone.org 4
- [35] www.forbes.com/sites/joemckendrick/2014/07/28/ibm-microsoft-surge-ahead-of-amazon-in-cloud-revenues-analysts-estimate 3
- [36] www.spec.org/jvm20083
- [37] D. Zhao, Z. Zhang, X. Zhou, T. Li, K. Wang, D. Kimpe, I. Raicu, F. S. Fusion, *Toward supporting data-intensive scientific applications on extreme-scale high-performance computing systems*, IEEE International Conference on Big Data, (2014), 61–70. 4.5
- [38] Z. Zheng, X. Wu, Y. Zhang, M. Lyu, J. Wang, *QoS Ranking Prediction for Cloud Services*, IEEE Trans. Parallel Distrib. Syst., **24** (2012), 1213–1222. 2
- [39] Z. Zhong, V. Rychkov, A. Lastovetsky, *Data partitioning on heterogeneous multicore platforms*, IEEE International Conference on Cluster Computing (CLUSTER), (2011), 580–584. 4.5