Performance evaluation of lossless file compression in the cloud: a study based on Eucalyptus platform

Erico Medeiros[1] , **Erica Sousa**[1 2] , **Eduardo Tavares**[1] , **Paulo Maciel**[1]

[1]Federal University of Pernambuco – Center of Informatics, Brazil

[2]Federal Rural University of Pernambuco, Brazil

`emm2,etgs,eagt,prmm@cin.ufpe.br`

Abstract. *As cloud computing becomes more commonly adopted, a problem arises concerning storage of large files in the cloud. For infrastructure providers, the cost of continuingly acquiring storage devices may be prohibitive, and, thus, file compression techniques are very prominent in this context. File compression would not only reduce on-demand storage costs, but it would also reduce transmission bandwidth and storage time. This paper evaluates and compares the performance of virtualized cloud machines and unvirtualized machines regarding file compression.*

Resumo. *À medida que a computação em nuvem vem se tornando mais comumente adotada, surge um problema relativo ao armazenamento de grandes arquivos na nuvem. Para provedores de infraestrutura, o custo de contiguamente adquirir dispositivos de armazenamento pode ser proibitivo, tornando técnicas de compressão de arquivos mais proeminente neste contexto. Compressão de arquivos não somente reduziria custos de armazenamento sob demanda, mas também reduziria a utilização da largura de banda da rede e tempo de armazenamento. Este artigo avalia e compara o desempenho de máquinas virtuais na nuvem e máquinas não-virtualizadas considerando a compressão de arquivos.*

Keywords: Cloud Computing, Eucalyptus Platform, Performance evaluation, Lossless file compression.

1. Introduction

Cloud Computing is a paradigm that has been developed over the past few decades, which combines different technologies, such as grid, cluster and utility computing as well as virtualization [Velte et al., 2010, Chee and Franklin Jr, 2009]. Cloud computing usually adopts a pay-as-you-go mechanism, in the sense that the user will only pay for the effective amount of utilized resources. Additionally, such a paradigm commonly offers a simpler way to manage a company infrastructure by allowing clients to focus on business services rather than investing time and money on building a complex physical infrastructure. Indeed, with the pay-as-you-go mechanism, users are not concerned with the infrastructure maintenance, for instance, hardware acquisition and upgrades. Cloud service providers also establish guarantees on their service levels, which include refunds in case of failure. [Hugos and Hulitzky, 2010, Chorafas and Francis, 2011].

Lossless File compression is a technique that groups files in an archive, adopting a dictionary to substitute recurrent terms by a smaller tag. As a consequence, storage requirements are reduced. To recover the original files, the file archiver (tool for compression and uncompression) replaces the tags by the associated piece of data, recreating the original structure [Ozsoy and Swany, 2011].

Recently, some papers have studied file compression in cloud computing [Ozsoy and Swany, 2011, Miyamoto et al., 2009, Krintz and Calder, 2001, Hovestadt et al., 2011], but not specifically in the context of performance evaluation. Such an evaluation is important to understand whether file compression is feasible for cloud computing or not.

Performance evaluation of computational systems consists in a set of measure and modeling-based techniques to reason about a system efficiency. Essentially, performance measurement involves monitoring a system while it is under a workload. The workload can either be real or synthetic, in which the last is simpler to simulate specific utilization conditions [Lilja, D.J., 2005].

This paper presents a study regarding performance evaluation of lossless file compression in the cloud. Experiments are conducted using Eucalyptus platform in order to allow the comparison of cloud virtual machines and unvirtualized machines (i.e., physical machines) using a workload for the file compression. This work also considers distinct virtual machine types to analyze the respective behaviors whenever file compression is adopted.

This paper is organized as follows. Section 2 presents some related works. Section 3 introduces some basic concepts to better understand this work. Section 4 depicts the adopted methodology for performance evaluation. Section 5 presents experimental results and section 6 concludes this work.

2. Related Works

In the last years, some experiments have been conducted to evaluate the performance of cloud computing infrastructures. In [Iosup et al., 2010], the authors evaluated the performance of different public cloud providers adopting a specific benchmark. [Ostermann et al., 2010] also present a performance evaluation of cloud infrastructures, focusing on Amazon Elastic Cloud Computing Platform.

Other papers devoted attention to storage resources concerning public and private clouds for scientific application. In [Shafer, J., 2010], the Eucalyptus Platform is tested assuming a variety of configurations to determine its suitability for applications with high I/O performance requirements, such as the Hadoop MapReduce framework for data-intensive computing. In [Ghoshal et al.,], the authors evaluate I/O performance using IOR benchmarks, which is a set of tools for understanding the I/O performance of high-performance parallel file systems.

[He et al., 2010] provide a theoretical basis for prototypes and topology analysis for cloud storage. Cloud storage is related to storage services that can be utilized by users in a cloud environment. However, the authors do not introduce means to reduce network utilization and storage needs, such as compression and deduplication.

[Miyamoto et al., 2009] present an architecture to improve network performance in cloud computing. Although this paper does not describe file compression as a storage functionality, it shows that some services, such as data caching, firewall and protocol optimization, have to be adapted to the cloud infrastructure. Regarding storage, file compression is a useful method to reduce storage issues in the cloud. [Ozsoy and Swany, 2011] suggest that, due to negative performance effects, file compression is usually neglected and proposes CUDA (Compute Unified Device Architecture) in GPUs (graphics processing units) as a mean to reduce these effects, due to its high parallel processing capabilities. [Hovestadt et al., 2011] discuss the throughput reduction caused by shared I/O in Clouds. Moreover, they developed an adaptive file compression scheme to optimize I/O performance.

Different from previous works, this paper presents a study regarding performance evaluation of file compression in the cloud, comparing virtualized and unvirtualized machines through representative workloads.

3. Preliminaries

This section introduces important concepts and background to better understand this work.

3.1. Eucalyptus Platform

Eucalytpus is an open-source cloud computing platform that allows the creation of private clusters in enterprise datacenters [Amazon Web Services, 2011]. Eucalyptus provides API compatibility with the most popular commercial cloud computing infrastructure, namely, Amazon Web Services (AWS), and it allows management tools to be adopted in both environments. Eucalyptus is designed for compatibility across a broad spectrum of Linux distributions (e.g., Ubuntu, RHEL, OpenSUSE) and virtualization hypervisors (e.g., KVM, Xen), which are responsible for virtualization itself. This work adopts such a platform, since the respective source code is available and its architecture facilitates the measurement of prominent aspects common to many cloud platforms.

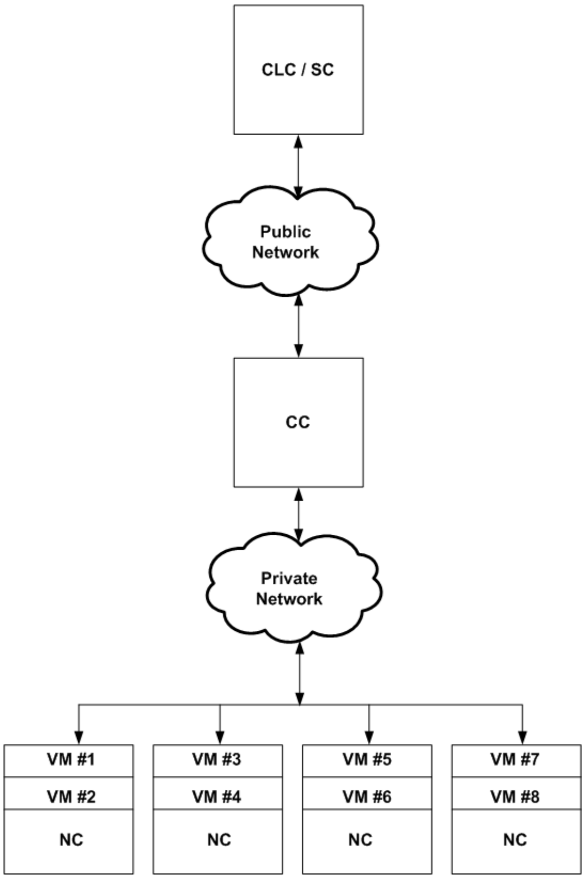

Figure 1: Eucalyptus Platform

Figure 1 depicts the Eucalyptus architecture. More specifically, Eucalyptus is composed of 5 major components that interact through webservice interfaces [D, J. and Murari, K. and Raju, M. and RB, S. and Girikumar, Y., 2010, Nurmi et al., 2009].

Cloud Controller (CLC) - CLC is the entry-point into the cloud for users and administrators. It queries node managers for information about resources, performs high-level scheduling decisions, and makes requests to cluster controllers.

Cluster Controller (CC) - CC acts as a gateway between the CLC and individual nodes in the data center. This component collects information about schedules and executions of virtual machines (VM) on node controllers, as well as it manages the respective virtual network.

Node Controller (NC) - NC contains a pool of physical computers that provide computational resources. Each machine contains a node controller service that is responsible for controling the execution, inspection, and termination of virtual machine (VM) instances. This component also configures the hypervisor and host OS as requested by CC [D, J. and Murari, K. and Raju, M. and RB, S. and Girikumar, Y., 2010, Nurmi et al., 2009].

Storage Controller (SC) - SC is a put/get storage service that implements Amazon's S3 interface, providing a mechanism for storing and accessing virtual machine images and user data.

Eucalyptus platform supports 5 different virtual machine (VM) types. Their respective default characteristics are presented in Table 1 [D, J. and Murari, K. and Raju, M. and RB, S. and Girikumar, Y., 2010, Nurmi et al., 2009], in which the rows represent virtual machine (VM) types and columns denote the associated resources.

Eucalyptus also offer the possibility of changing the characteristics of those virtual machine types, in order to offer suitability for diferent clients. The instantiation of Vitrual Machines (VMs) are requested through the Cloud Controller (CLC), which is responsible for gathering the resources and managing the instance.

Table 1: Virtual Machine Instances Types

VM type	CPU (Core)	Memory (MB)	Disk (GB)
m1.small	1	192	2
c1.medium	1	256	5
m1.large	2	512	10
m1.xlarge	2	1024	20
c1.xlarge	4	2048	20

3.2. Lossless File Compression

Lossless File compression is a technique that mainly search for redundant pieces of data in a file and creates a dictionary associating each redundant piece of data with a tag (i.e., a smaller value), which replaces the original data, in an attempt to reduce the file size. Algorithms like LZMA (Lempel-Ziv-Markov chain algorithm) or LZ77 (lossless data compression algorithm published in 1977) are usually adopted to handle these dictionaries. To uncompress, the file archiver needs to replace the dictionary tags found in a file and by its original piece of data [Krintz and Calder, 2001]. 7-zip tool (which utilizes the LZMA algorithm) is very well-known for its interesting compression ratios [Pavlov, 2012].

When applying Lossless file compression techniques in Cloud Computing, for instance, the users would pay half the price to store the same files when the adopted technique achieves a 50% compression ratio. Whenever the user requires to archive new files or recover some previously stored, the platform would only launch a virtual machine to uncompress the specified file and transfer it to the desired destination.

4. Performance Evaluation Methodology

This section presents the adopted methodology for performance evaluation concerning the Eucalyptus platform and compression techniques. Only general aspects concerning performance evaluation are presented. Specific tools and technical details regarding the experiments are only explained in the next section. Figure 2 shows the activity diagram of the methodology.

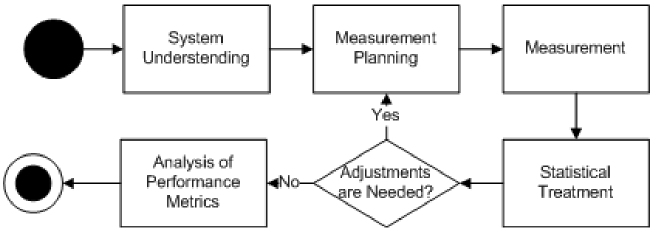

Figure 2: Methodology

The methodology consists of five activities, which are system understanding, measurement planning, measurement, statistical treatment and analysis of performance metrics. The methodology's first activity concerns understanding the system, its components, their interfaces and interactions. This activity should provide the set of metrics that should be evaluated. In this work, we are concerned with processor utilization, total delay time [Hovestadt et al., 2011] and system efficiency.

The second activity results in a document that describes how the measurement should be performed, the tools calibration, the frequency of data collection and how to store the measured data (presented further in this work).

The measurement activity consists of five steps, which were automated by bash [Stallman, 2012] scripts. (see Figure 3).

The first step instantiates the virtual machines. The second step starts the performance monitoring tool MPstat [Godard, S., 2004]. The third step configures the 7-zip compression tool on virtual machines. The fourth step executes the compression tool based on the adopted workload. Then, the platform terminates the virtual machines. Finally, report files with the measured data are created, and the measurement process is restarted.

In the measurement activity, the monitoring tools are started before the compression tool. When the compression tool ends, the monitoring tool is closed in order to create the report files.

The fifth activity applies statistical methods in measured data to provide information about the evaluated system.The final result is the mean (μ_D) and standard deviation (σ_D) of initially defined metrics.

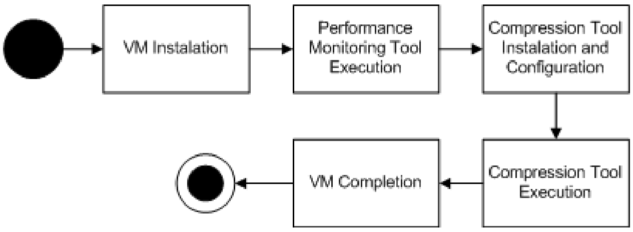
VM Instalation
Performance Monitoring Tool Execution
Compression Tool Instalation and Configuration
Compression Tool Execution
VM Completion

Figure 3: Measurement Activity

5. Experimental Results

This section presents the technical details regarding the adopted experiments and the respective results obtained from it.

This work adopts an architecture based on Figure 1 (i.e., Eucalyptus Architecture), taking into account 1 main controller (combination of cluster, cloud and storage controllers) and 4 node controllers. These 5 machines have the same configuration: Intel^R Core^{TM}2 Duo CPU E6550 2.33GHz, 2GB DDR2 RAM, 160 GB Hard Disk and 100MB ethernet interface.

We have adopted 7-zip tool to create the workload, as it offers a good compression ratio. Additionally, scripts were utilized to repeat the process of workload in order to automate the workload and data gathering. Table 2 depicts the VM types used by Eucalyptus (considering the architecture mentioned above), whereas Table 3 depicts the adopted scenarios, which represent different cloud demands concerning file compression. Scenario 5 represents an unvirtualized physical machine (with the same configuration of the previously mentioned architecture).

Table 2: VM types

VM type	Processor cores	MB RAM	GB Storage	Max machines
m1.small	1	512	10	8
c1.medium	1	1024	10	4
m1.large	2	1024	10	4
m1.xlarge	2	1536	10	4

Table 3: Scenarios

Scenario	VM type	Physical Machines needed	Used Machines
1	m1.small	5	8 (virtual)
2	c1.medium	5	4 (virtual)
3	m1.large	5	4 (virtual)
4	m1.xlarge	5	4 (virtual)
5	-	1	1 (physical)

In Table 3, Physical Machines Needed represent the number of physical machines necessary to proceed with the experiments. For scenarios 1-4, this value is equal to the number of machines that composed the cloud. In scenario 5, the value is one (hence only one machine is needed).

For the compression tool execution, 5 files were randomly generated using a C program and respectively had the following sizes: 200 bytes, 20 kilobytes, 2 megabytes, 191 megabytes and 763 megabytes (in a total of approximately 955 megabytes).

Basically, 7-zip offers 10 gradual levels of compression (0-9) where 0 represents no compression and 9 represents maximum compression. In an preliminary test using levels 1,3,5,7 and 9, levels 1 and 3 were approximately 70% faster than levels 5,7 and 9, but, for all levels, the archive total size was approximately 343MB, considering the files described above.

The fourth step of the measurement activity was executed as follows: each execution had a level 1 compression followed by further uncompression, level 3 compression and further uncompression. The time spent in the execution was measured and will be further treated as execution time.

Figure 4 shows the execution time of all scenarios and Figure 5 depicts processor utilization which was estimated using mpstat, a small utility for unix-like operating systems that details CPU statistics. In both figures, the horizontal axis represents the scenarios.

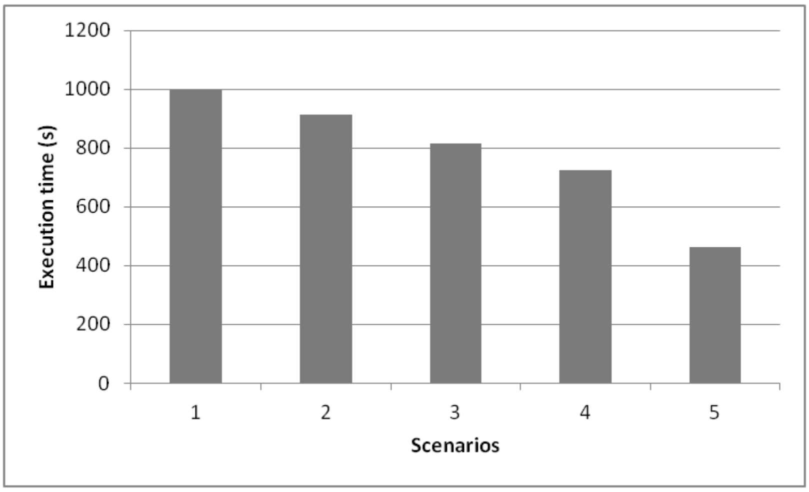

Figure 4: Execution Time

In Figure 4, lower values represent the best results. In this case, Scenario 5 (unvirtualized machine) is the best. The virtual machines, on the other hand, had a relatively bad performance compared to scenario 5 (60% to 120% longer execution time). Greater processor utilization mean a lesser idle processor time, representing a worthier scenario. In this case, scenarios 1 and 2 behaved better than scenarios 3,4 and 5.

To better assess scenarios 1 to 5, considering both metrics (execution time and utilization) we conceived a metric that takes in consideration number of executions in a defined time, amount of execution (machines) run simultaneously and the number of physical machines needed for the structure. Equation 1 was used to measure the benefit of each one of the scenarios.

$$\beta = \frac{\lambda \times smr}{pmn} \tag{1}$$

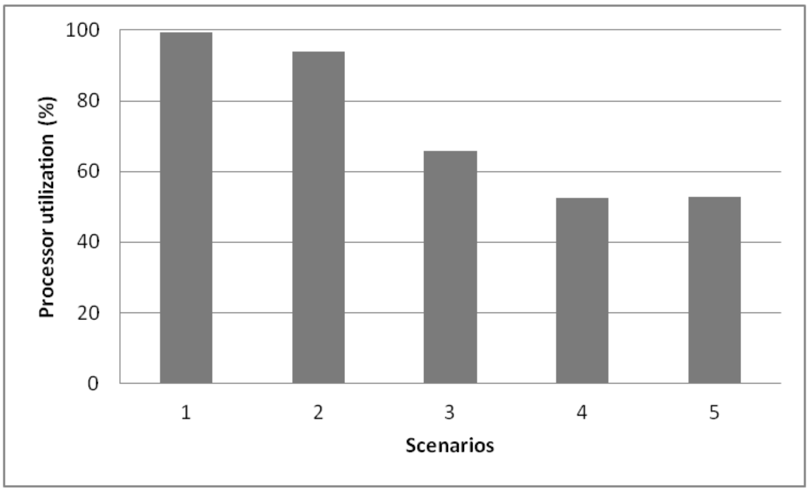

Figure 5: Processor Utilization

Where λ represents the amount of executions in a defined time for one machine. For this work, the time used was 1 hour. smr represents the number of simultaneously machines running, which is depicted in 3 as used machines. Then, $\lambda \times smr$ shows the total number of executions in a defined time for all machines of each scenario. pmn represents the number of physical machines needed for each scenario, also depicted in 3.

Table 4 depicts the approximate values found for λ and $\lambda \times smr$ for each scenario. Note: λ is based on experimental results of execution time.

Table 4: Scenarios considering 50 NCs

Scenario	λ	$\lambda \times smr$
1	3.611	28.888
2	3.939	15.756
3	4.409	17.636
4	4.962	19.848
5	7.782	7.782

The best scenario is the one with the better benefit. Figures 6 present β for all scenarios.

As it is possible to see in Figure 6, scenario 5 is the best option, closely followed by scenario 1. Table 5 shows theoretical values for smr and pmn, when using 50 node controllers instead of 4. Note that smr and pmn do not change for scenario 5, hence there is no virtualization.

Figure 7 depicts the new values of β calculated by the values in Table 5. It is important to notice that the the values of λ are kept.

When comparing scenario 1 in figures 6 and 7, it is possible to notice a raise in benefit. The reason is the $\frac{smr}{pmn}$ ratio, which changes from $8/5 = 1.6$ to $100/51 \approx 1.96$. This means that, as λ is kept as a constant for each scenario, the bigger the $\frac{smr}{pmn}$ ratio is, the better the scenario.

In other words, for each scenario, as the cloud infrastructure grows larger, the

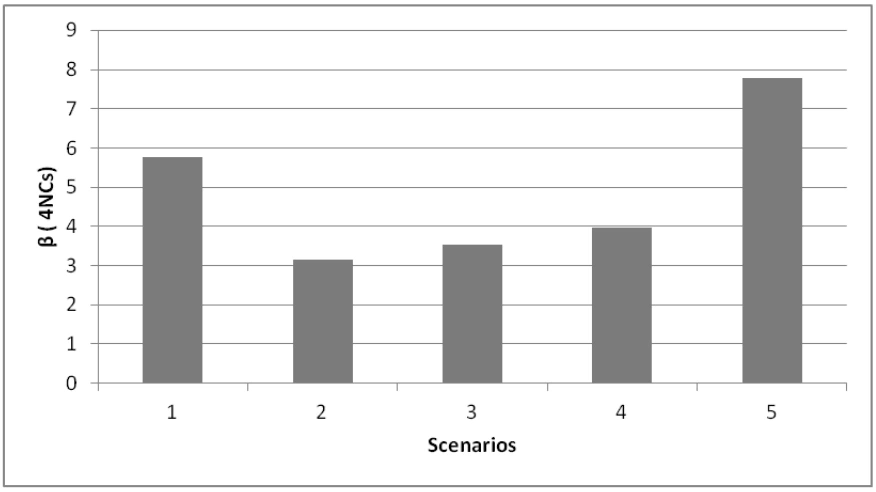

Figure 6: Benefit (4 NCs)

Table 5: Scenarios considering 50 NCs

Scenario	smr	pmn
1	100	51
2	50	51
3	50	51
4	50	51
5	1	1

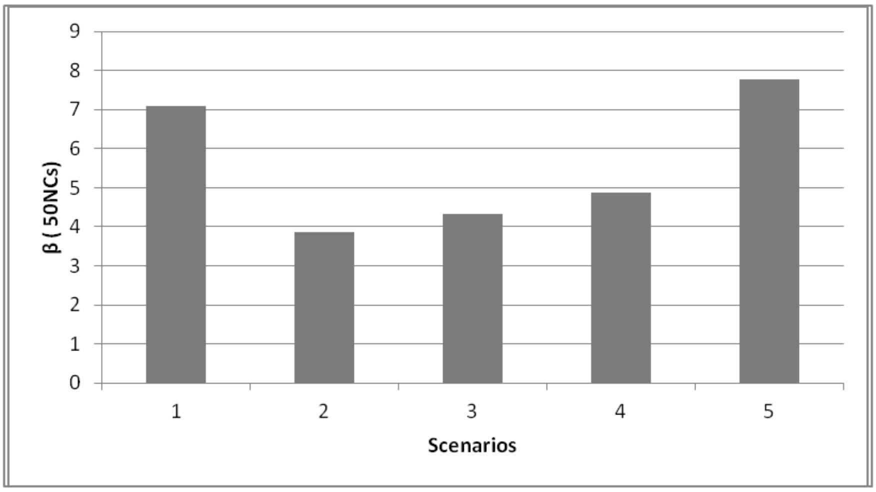

Figure 7: Benefit (50 NCs)

benefit level raises when the resources are fully used. This means that, in some scenarios, cloud performance can be equated to a unvirtualized physical machine. It is important to notice that cloud computing also offers scalability, thus reducing costs.

6. Conclusions and Future Works

This work presented a performance evaluation comparison of Cloud Virtual Machines and Unvirtualized Machines regarding compression.

Experimental results demonstrate that the performance of file compression may not be considerably affected in cloud infrastructures, when compared to unvirtualized machines. The adopted workload demonstrate that benefit usually improves whenever the number of virtual machines in a node controller is increased.

As a future work, we intend to evaluate other cloud platforms, such as Amazon Elastic Computing Cloud, and, also, to contemplate other metrics, such as cost savings due to the file compression.

References

Amazon Web Services (2011). Eucalyptus open-source cloud computing infrastructure - an overview. technical report, eucalyptus, inc.

Chee, B. and Franklin Jr, C. (2009). *Cloud computing: technologies and strategies of the ubiquitous data center*. CRC.

Chorafas, D. and Francis, T. . (2011). *Cloud computing strategies*. CRC Press.

D, J. and Murari, K. and Raju, M. and RB, S. and Girikumar, Y. (2010). Eucalyptus Beginner's Guide - UEC Edition.

Ghoshal, D., Canon, R., and Ramakrishnan, L. Understanding i/o performance of virtualized cloud environments.

Godard, S. (2004). *Sysstat:System performance tools for the Linux OS*.

He, Q., Li, Z., and Zhang, X. (2010). Study on cloud storage system based on distributed storage systems. In *Computational and Information Sciences (ICCIS), 2010 International Conference on*, pages 1332–1335. IEEE.

Hovestadt, M., Kao, O., Kliem, A., and Warneke, D. (2011). Evaluating adaptive compression to mitigate the effects of shared i/o in clouds. In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, pages 1042–1051. IEEE.

Hugos, M. and Hulitzky, D. (2010). *Business in the Cloud: What Every Business Needs to Know About Cloud Computing*. Wiley.

Iosup, A., Ostermann, S., Yigitbasi, N., Prodan, R., Fahringer, T., and Epema, D. (2010). Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing. *IEEE Transactions on Parallel and Distributed Systems*, pages 1–16.

Krintz, C. and Calder, B. (2001). Reducing delay with dynamic selection of compression formats. In *High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on*, pages 266–277. IEEE.

Lilja, D.J. (2005). *Measuring computer performance: a practitioner's guide*. Cambridge Univ Pr.

Miyamoto, T., Hayashi, M., and Tanaka, H. (2009). Customizing network functions for high performance cloud computing. In *Network Computing and Applications, 2009. NCA 2009. Eighth IEEE International Symposium on*, pages 130–133. IEEE.

Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., and Zagorodnov, D. (2009). The eucalyptus open-source cloud-computing system. In *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 124–131. IEEE Computer Society.

Ostermann, S., Iosup, A., Yigitbasi, N., Prodan, R., Fahringer, T., and Epema, D. (2010). A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing. pages 115–131.

Ozsoy, A. and Swany, M. (2011). Culzss: Lzss lossless data compression on cuda. In *Cluster Computing (CLUSTER), 2011 IEEE International Conference on*, pages 403–411. IEEE.

Pavlov, I. (april, 2012). 7-zip official website, http://www.7-zip.org/.

Shafer, J. (2010). I/o virtualization bottlenecks in cloud computing today. In *Proceedings of the 2nd conference on I/O virtualization*, pages 5–5. USENIX Association.

Stallman, R. (april,2012). Gnu bourne again shell website, http://www.gnu.org/software/bash/.

Velte, A., Velte, T., Elsenpeter, R., and Babcock, C. (2010). *Cloud computing: a practical approach*. McGraw-Hill.