

Available online at www.sciencedirect.com**ScienceDirect**

Procedia Technology 16 (2014) 649 – 658

Procedia
Technology

CENTERIS 2014 - Conference on ENTERprise Information Systems / ProjMAN 2014 - International Conference on Project MANagement / HCIST 2014 - International Conference on Health and Social Care Information Systems and Technologies

Performance of Jails versus Virtualization for Cloud Computing Solutions

Carlos Antunes^a, Ricardo Vardasca^{b,c*}

^aSchool of Technology and Management, Polytechnic Institute of Leiria, Morro do Lena- Alto do Vieiro 2411-901 Leiria, Portugal

^bLABIOMEP, IDMEC-FEUP, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias S/N 4200-465 Porto, Portugal

^cFaculty of Computing, Engineering and Science, University of South Wales, Trefforest, CF37 1DL, United Kingdom

Abstract

The existence of a virtualization layer within the cloud affects the resources optimization and the reduction of requirements of its implementation. In this publication the focus will be the use of jail environments, provided by the FreeBSD Operating System, which present a relevant set of features that can enhance the increase of performance. A set of data collection tests that allows measuring the degree of optimization obtainable in current models of cloud computing, based on the use of hypervisors tests, are presented. These tests proved that the increase in performance and optimization of resources is possible, bringing up the need to adapt the current models of cloud computing for the use of such solutions. However, this increase in performance, leads to a loss of flexibility of the usage of independent operating systems, which is not relevant to the model of cloud computing business.

© 2014 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Peer-review under responsibility of the Organizing Committee of CENTERIS 2014.

Keywords: Cloud computing; jail environments; optimization; performance.

* Corresponding author. Tel.: +351 22 041 4741; fax: +351 22 508 1445.

E-mail address: ricardo.vardasca@fe.up.pt

1. Introduction

Currently the vast majority of information systems are based on cloud computing, where applications and data are stored remotely. Resource consumption and performance in the execution of applications are matter of extreme importance with regard to computational systems and high availability solutions in particular. When referring about cloud solutions these attributes have an important and significant role in the nature of its business model [1].

Major part of the existent cloud computing solutions is based in virtualization. Which is a method to emulate the presence of a specific set of hardware [2], providing platform independent installation environments. Where each virtualization solution implements different ways of optimizing data access, either through mapped shares or via network protocols [3].

The requirements for implementation of any high performance solution are a major obstacle to its use, and indulging in cloud solutions, the investment required is huge. For this reason it is necessary to conduct studies and tests in order to increase the performance of the solutions and reduction of hardware [4-10].

Several solutions have been presented [4, 9], and in particular, container-based solutions, has undergone a constant reference to the future of the cloud, but the lack of measurements, combined with difficulty of presenting comparative amounts in real solutions has hampered the use of this types of solutions.

There is thus a clear need to adapt the current models used in the cloud, since the features of containers, and in particular of jail environments are not available in virtualized environments currently used in cloud computing solutions [9].

1.1. Jail environments

The jail environments are environments based on trapping processes, best known as containers and are essentially based on three features of the system: kernel, file structure and network address [2].

It may resemble the virtualized systems, however, it does not require any type of virtualization, and the access to resources is made concurrently by all running instances running, although, they are limited to their specific system files and are inflexible environments, because they are dependent on the type of base operating system [2].

The operating system should be the point of greatest gain in jails based systems, since the trapped systems are based on the sharing of the running kernel by the base system, thus eliminating the need for additional resources related to the implementation of a new system instance operating. Thus there is a significant saving of resources such as storage space, memory and processing.

Another pertinent point is required by the file structure of a jail, which is relatively lower than any actual operating system installation files space, being a simplified copy of the basic structure of a base FreeBSD system, requiring fewer storage resources.

The jail environments are based on optimizing the use of the base system, and focused mainly on two aspects, one related to the sharing of software among environments and the ability to map distributed areas [9].

The aim of this study is to measure the performance of data access, using up to 3 different protocols, which allows a comprehensive view of the behavior of the performed tests. This is a simple way of starting to research and to collect data for possible optimizations using open source software, since it requires less hardware resources.

2. Methodology

Aiming at the need of obtaining measurable values for preparation of a comparative analysis between virtualized and imprisoned environments at the level of performance and consumption, we begin with an analysis of high-level requirements.

Is shown in Table 1, a comparative analysis and summary of the requirements required by the solutions of implementing virtual environments.

From Table 1 it can be assumed that the use of trapped ambient will allow a considerable reduction in the level of requirements to implement this type of solution, and the resources consumed by the computer system will be much lower.

In order to evaluate the gain enabled by the use of jails environments and the performance of applications in jail environment, two scenarios were constructed for development of performance and resource consumption tests.

The HTTP protocol is main form of access and distribution of content on the Internet, the SSH protocol allows data transfer through secure channels and the NETBIOS protocol is a system capable of having remote files used among different operating systems. Due to these protocols being the most used in cloud computing solutions, they selected to be used within this study.

Table 1. Comparison of resources

Resources	Virtual machines	Jail services
Virtualization System	Yes	Base Operating System
Operating System	New	No
File structure	Full	Simplified
Virtualized drivers	Yes	No
Sharing data	Network and shares virtualized	Network or local
Additional drivers	Yes	No
Startup time	Startup of the Operating System	Less than 2 seconds

2.1. Scenarios

The development scenarios of the tests were built using the solutions currently used in commercial architectures, and using generic applications that allow us to prepare the respective tests:

Implementation in jails - Installation of applications in different environments jail in order to validate the performance changes (Fig. 1).

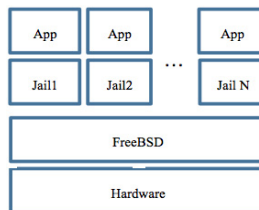


Fig. 1 – Scenario for testing based in Jail environment.

Deployment with Virtualization - Installing a system virtualization support, and implementation of applications in isolated virtualized environments (Fig. 2).

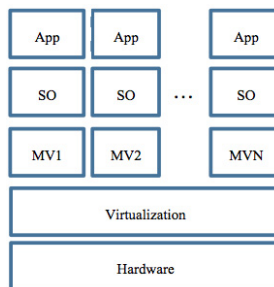


Fig. 2 – Scenario for testing based in virtualized environments

For construction of the tests the following material was used:

- Hardware - Server 2.0 with Xeon processor, 2GB RAM, 250GB HDD SATA1.0 and 100Mb Ethernet NIC.
- Operating Systems - Hypervisor 3.50 machines with Ubuntu 10.04lts, and FreeBSD 8.2 with jails.

Regarding the choice of applications, it was decided to choose three solutions, which were representative of resource use, and enabling a quick, clear and intuitive way to obtain measurable values:

- Applications – System remote access (SSH), Web Server (apache2) and remote file system (SAMBA).

2.2. Tests

Two different sets of tests were developed, the first focuses on the consumption of resources required by each of the respective solutions and applications, while the second set aims at obtaining access times to resources available for each of these applications.

For tests of consumption, the focus is not related to settings, and to avoid any kind of recovery, the applications were installed with the default settings, not having undergone any change except the definition of the appropriate network addresses. Consumption of CPU, RAM and disk space required for installing the environment and respective applications were evaluated at this point. It was also evaluated the startup time and availability of services. For these measurements the system tools *top* and *du* were used. In the case of virtual machines these values were confirmed with graphics provided by the hypervisor.

The accesses to resources tests were divided into three test groups in order to use three different protocols. The protocols to be used were SSH, HTTP and NetBIOS, and aims to analyze the different behaviors of data access. Each protocol will be tested in four different scenarios. First will be unique access to an environment, then access to a unique environment with both running, followed by concurrent access to a single environment and ends with concurrent access to both environments. For the development of tests two clients with operating systems Microsoft and Linux were implemented in order to concurrently access the respective features. Physically tests have been developed using a simple architecture shown in Fig. 3.

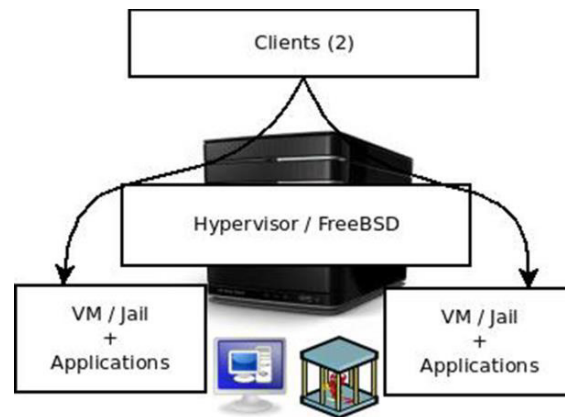


Fig. 3 – Physical scenario for tests

Each test was performed 20 times. Being presented the results in mean values, in order to discard values that have undergone the influence of external factors, such as interference at the communication level.

The tests are aimed at determining the ability of each environment to provide access to resources, simple access tests were performed with only one client, two concurrent accesses, and using two clients simultaneously accessing the same resources.

3. Results

Due to constraint of hardware required by the hypervisor, the tests were limited to the maximum use of two instances.

3.1. Resources consumption tests

To emphasize that what is intended with these tests is to measure resource consumption and analyze the performance of applications within the same system, and subsequently analyze differences in performance between systems. As a first test, it evaluates the consumption of resources required for each implementation.

From the analysis of representative charts (Fig. 4) and table 2, it can be seen that with increasing number of virtual environments with the usage of virtual machines it triggers the resource consumption. While with the use of ambient jail these resources are mainly channeled to the services provided, and the number of instances has a negligible impact on the computer system.

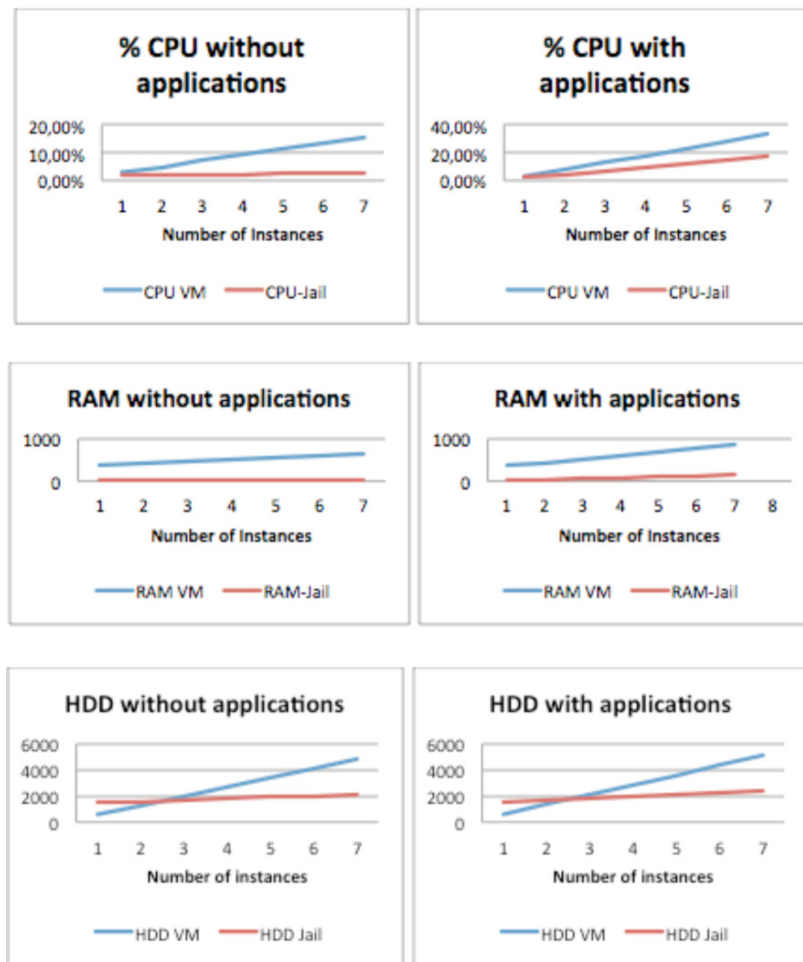


Fig. 4 – Resources consumption charts

Table 2. Consumptions resume.

	Virtual machines		
	CPU	Memory	HDD
Virtualization support system	~3%	~360MB	~600MB
Virtual Machine without services	~2%	~50MB	~700MB
Virtual Machine with services	~5%	~80MB	~750MB
	Jails		
FreeBSD (with source code and ports)	~2%	~20MB	~1500MB
Jail without services	~0.1%	~0.5MB	~100MB
Jail with services	~2.5%	~20MB	~150MB

Noted that the resources required to launch a jail environment are insignificant when compared with the requisites needed to launch a virtualized environment, increasing the fact that systems based on jails effecting competitors and direct access to devices not requiring additional virtualized drivers, enabling a performance boost in their usage.

From the analysis of the results obtained for time and resource consumption during startup of different environments (table 3), it can again be verified that the jails are more effective than the virtual machines, having a minimum release time accompanied by a irrelevant hardware consumption irrelevant that does not interfere with the performance of the equipment.

Table 3. Startup resources consumption.

	Startup time	CPU
Virtual Machine without services	~15s	~45%
Virtual Machine with services	~25s	~45%
Jail without services	~1,5s	~10%
Jail with services	~8s	~20%

The same is not true for virtual machines, which require a considerable time to boot as well as a significant amount of hardware resources, that may temporarily affect the stability of the hypervisor and virtual machines that are in use during the activity.

3.2. Data access performance tests

When referring to high availability, resource consumption is an important factor, where the availability of access to the data plays a major role, being one of the main reasons for implementing such solutions.

In this group of tests, the aim is to evaluate the availability of both scenarios cited above, but a trend of data access. The test is based on accessing data via SSH (Secure Shell), HTTP (Hypertext Transfer Protocol) and NetBIOS (Network Basic Input / Output System) based on the Apache 2.2.2 Samba OpenSSH services and features. Again, no additional settings to default installations were made. The intention was to monitor the impact of the number of environments, and not the configuration optimization carried out in each of the systems.

The testing is based on the access and copy of the data stored in the built environment. The first of these tests is based on access to a single file of approximately 450MB and the second access a file structure with approximately 2500 nodes and around 75MB of physical space.

The file type was chosen based on a set of assumptions that allow to have a vision of their impact without these represent a threat to the use of hardware, and they were:

- Minimum transfer time 15s;
- Minimum size 50MB total to download;
- Minimum number of nodes 5000;

- Diversified types of nodes:
 - Text files;
 - Binary files;
 - Folders;
 - Diversified sizes.

In the first test, data access through SSH, it consisted in access and copy data using the SSH protocol, making access and unique copy with only two active environments (table 4).

Table 4. Data access through SSH in a single machine.

		Exclusive access	
		Single file	File structure
Native Ubuntu		38.67	22.19
Hypervisor	VM1	128.15	37.82
	VM2		Stopped
Native FreeBSD		38.4	20.38
FreeBSD	Jail1	38.4	20.54
	Jail2		Stopped

Table 5 shows the values of the data access via SSH tests, with the second active machine are presented.

Table 5. Data access through SSH with 2 active machines.

		Exclusive access	
		Single file	File structure
Native Ubuntu		38.67	22.19
Hypervisor	VM1	137.2	38.25
	VM2		Active
Native FreeBSD		38.4	20.38
FreeBSD	Jail1	38.4	20.63
	Jail2		Active

It can be verified that, from the data presented in tables 4 and 5, that the performance worsens for the virtualized systems significantly relative to native system, and the simple fact that there a second active virtual machine causes a negative change in performance of data access, which despite has not been very significant, can be noticed. In the use of jails such behavior is not noticed, and the existence of a jail not cause any new variation, since the only difference is the existence of a new process.

Table 6 shows the results of data access and copy from multiple accesses. In this scenario there are two client machines with simultaneous access to the same data. Table 7 presents the results of concurrent and differentiated data access via SSH from 2 clients are presented.

As observed in the first group of accesses, the access to information located in different virtual machines have very significant implications in the performance of data access through SSH, traduced in results shown in tables 6 and 7. With the existence of a second client the access to a single file has presented an accentuated drop in performance.

With regard to the jail environments, the existence of an additional jail environment did not cause any interference on the performance of data access through SSH on the other jail environment. On the contrary, the existence of a second process resulted in an increase of the system performance.

Table 6. Concurrent data access through SSH.

		Concurrent access			
		Single file		File structure	
		Client 1	Client 2	Client 1	Client 2
Native Ubuntu		76.84	76.94	24.60	32.29
Hypervisor	VM1	256.5	255.31	70.04	70.07
	VM2			Stopped	
Native FreeBSD		76.49	76.48	23.29	30.69
FreeBSD	Jail1	76.47	76.48	23.19	30.69
	Jail2			Stopped	

Table 7. Concurrent data access through SSH with 2 active clients.

		Concurrent access			
		Single file		File structure	
		Client 1	Client 2	Client 1	Client 2
Native Ubuntu		76.84	76.94	24.60	32.29
Hypervisor	VM1	1353.01		80.61	
	VM2		1306.69		80,66
Native FreeBSD		76.49	76.48	23.29	30.69
FreeBSD	Jail1	69.00		23.34	
	Jail2		63.27		29.80

The next test is based on data access through HTTP and NetBIOS. To support this test, the apache and SAMBA services were installed with generic configuration. The tests were conducted using the *wget* utility. The results of the tests performed are shown in Tables 8, 9 and 10.

Table 8. Simple data access through HTTP and NetBIOS.

		Concurrent access			
		Single file		File structure	
		HTTP	NetBIOS	HTTP	NetBIOS
Native Ubuntu		38.01	38.05	217.38	26.90
Hypervisor	VM1	110.67	105.92	345.93	43.52
	VM2			Stopped	
Native FreeBSD		38.03	38.07	142.79	26.49
FreeBSD	Jail1	38.05	38.07	157.75	25.85
	Jail2			Stopped	

As was found with testing data access through SSH, availability of accessing data using the HTTP and NetBIOS protocols, brings evidence that the number of virtual machine instances negatively influences the overall performance of virtualized systems, being a limiting factor for the scalability of the solution.

With the use of jail environments, that impact was greatly reduced, reinforcing the concept of using jail environments in high availability solutions based on open source services. Presenting these as a great choice, allowing a great resource optimization without compromising the availability and reliability of the system.

Table 9. Simple data access through HTTP and NetBIOS with 2 active machines.

		Concurrent access			
		Single file		File structure	
		HTTP	NetBIOS	HTTP	NetBIOS
Native Ubuntu		38.01	38.05	217.38	26.90
Hypervisor	VM1	115.23	109.81	361.99	43.52
	VM2			Stopped	
Native FreeBSD		38.03	38.07	142.79	26.49
FreeBSD	Jail1	38.06	38.09	157.57	25.87
	Jail2			Stopped	

Table 10. Concurrent data access through HTTP and NetBIOS with 2 active clients.

		Concurrent access							
		Single file				File structure			
		HTTP		NetBIOS		HTTP		NetBIOS	
		Client 1	Client 2	Client 1	Client 2	Client 1	Client 2	Client 1	Client 2
Native Ubuntu		75.90	75.89	75.81	76.07	272.81	282.11	32.92	33.35
Hypervisor	VM1	602.65		579.34		1150.81		71.79	
	VM2		1177.73		200.52		1205.86		70.49
Native FreeBSD		76.05	76.04	76.07	76.08	184.07	224.13	30.74	31.09
FreeBSD	Jail1	75.80		75.74		210.33		31.64	
	Jail2		75.96		76.07		208.33		31.97

As a summary, is presented the table 11, which is comparison of performance, calculated through the coefficients of performance based on the values of the tests (values between 0 and 1 for performance improvement, values close to 1 for unchanged performance, and values above 1 for performance degradation).

Table 11. Performance comparison.

	Exclusive access		Concurrent access	
	VM1	VM1 + VM2	VM1	VM1 + VM2
Native Ubuntu	3 to 5	8 to 14	1.5 to 6	2.5 to 6
Native FreeBSD	J1	J1 + J2	J1	J1 + J2
	0.99 to 1.2	0.85 to 1.2	0.99 to 1.2	0.95 to 1.2

In table 11 are shown the comparisons between native and virtualized and jail environments implementation. In a very succinct way it can be stated that the use of virtualized environments translates into a loss of performance against the base machine, which tends to worsen with the increasing number of instances running. Similar trend is observed in the consumption of hardware resources required by the running instances.

Given this situation, it can be stated that the performance of a virtualized system will suffer a loss of performance greater than 50%, and that the number of virtualized instance running, negatively influences the performance of the computer system and can easily compromise the viability of the solution.

As opposite, it is observed that when using jails environments, there is no performance influence to the system, and in certain cases, it even increases the performance of the solution. It was also observed that the number of

instances of jail environments does not influence negatively the computer system, allowing flexibility of high availability solutions.

4. Discussion

As already mentioned, the virtual environment can never be considered a substitute for virtualized environments, due to operating system constraint, which is limited to the kernel used in the base system. However, for specific solutions, it may be a great contribution, especially in regard to the provision of access to data stored in remote areas. The jail environments, as all solutions based containers have a huge negative contribution, which is the loss of flexibility in the use of independent operating systems, but this is not relevant for the business model of cloud computing.

It can be concluded that the use of jails environments instead of virtualized systems for high availability solutions, given certain scenarios, can increase the performance of the solution in more than 50%, reducing the resources consumption in the same scale. Given that not all capabilities were tested by the jails, such as the sharing of binaries, it is still possible to optimize more jails based environments.

This study provides measurable results, which are important for developing high availability solutions based on jail environments. As future research it is suggested a development of a generic high availability model based on jails to access data stored remotely.

References

- [1] Strauss D. Containers - Not Virtual Machines, Are the Future Cloud. *The Linux Journal* 2013; 228:118-123.
- [2] Vouk M A. Cloud computing—issues, research and implementations. *CIT. Journal of Computing and Information Technology* 2008; 16(4):235-246.
- [3] Chaisiri S, Lee BS, Niyato, D. Optimization of resource provisioning cost in cloud computing. *Services Computing, IEEE Transactions on* 2012; 5(2):164-177.
- [4] Kamp P-H, and Watson RNM. Jails: Confining the Omnipotent Root. In *Proceedings of the 2nd International SANE Conference, Maastricht, The Netherlands, May 2000.*
- [5] Buyya R, Yeo CS, Venugopal S. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *10th IEEE International Conference on High Performance Computing and Communications (HPCC'08) 2008; 5-13.*
- [6] Grossman RL, Gu Y, Sabala M, Zhang W. Compute and storage clouds using wide area high performance networks. *Future Generation Computer Systems* 2009; 25(2):179-183.
- [7] Gu Y, Grossman RL. Sector and Sphere: the design and implementation of a high-performance data cloud. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 2009; 367(1897):2429-2445.
- [8] Napper J, Bientinesi P. Can cloud computing reach the top500?. In *Proceedings of the combined workshops on UnConventional high performance computing workshop plus memory access workshop 2009; 17-20.*
- [9] Antunes C, Vardasca R. Building Low Cost Cloud Computing Systems. *International Journal of Advanced Computer Science and Applications (IJACSA)* 2013; 4(5):48-52.
- [10] Vecchiola C, Pandey S, Buyya R. High-performance cloud computing: A view of scientific applications. In *IEEE 10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN) 2009; 4-16.*