Luís Miguel Teixeira da Silva

Bachelor of Science

Replication and Caching Systems for the support of VMs stored in File Systems with Snapshots

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Computer Science and Informatics Engineering

Adviser: Nuno Preguiça, Associate Professor,
NOVA University of Lisbon

Co-adviser: Pedro Medeiros, Associate Professor,
NOVA University of Lisbon

Examination Committee

Chairperson:	Name of the committee chairperson
Raporteurs:	Name of a raporteur
	Name of another raporteur
Members:	Another member of the committee
	Yet another member of the committee

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

March, 2018

Replication and Caching Systems for the support of VMs stored in File Systems with Snapshots

This document was created using the (pdf)LaTeX processor, based in the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Lorem ipsum.

Acknowledgements

The work presented in this document would never see the light of day if not for the collaboration of several people to whom I wish to manifest my profound gratitude and recognition.

Prof. Nuno Preguiça, Prof. Pedro Medeiros, Prof. Paulo Lopes

I also would like to acknowledge the following institutions for their hosting and financial support: *Departamento de Informática and Faculdade de Ciências e Tecnologia of the Universidade NOVA de Lisboa* (DI-FCT NOVA); the *NOVA Laboratory of Computer Science and Informatics* (NOVA LINCS) in particular the Computer Systems group; *SolidNetworks – Business Consulting, LDA* of the *Reditus S.A. Group*; and the funding provided through the *COMPETE2020 / PORTUGAL2020* program for the *iCBD* project (POCI-01-0247-FEDER-011467).

Abstract

Over the span of a few years, there were fundamental changes in the way computing power is handled. The heightening of virtualisation changed the infrastructure model of a *data centre* and the way physical computers are managed. This shift is the result of allowing for fast deployment of Virtual Machines (VMs) in a high consolidation ratio environment and with minimal need for management.

New approaches to virtualisation techniques are being developed at a never seen rate. Which leads to an exciting and vibrating ecosystem of platforms and services seeing the light of day. We see big industry players engaging in such problems as *Desktop Virtualisation* with moderate success, but completely ignoring the already present computation power in their clients, instead, opting for a costly solution of acquiring powerful new machines and software. There is still space for improvement and the development of technologies that take advantage of the onsite computation capabilities with minimum effort on the configuration side.

This thesis focuses on the development of mechanisms for the replication and caching of *VM* images stored in a conventional file system with the ability to perform snapshots. There are some particular items to address: like the solution needs to follow an entirely distributed architecture and fully integrate with a parallel implemented client-based Virtual Desktop Infrastructure (VDI) platform; needs to work with very large read-only files some of them resulting from the creation of snapshots while maintaining some versioning features. This work will also explore the challenges and advantages of deploying such system in a high throughput network, maintaining high availability and scalability properties while supporting a broad set of clients efficiently.

Resumo

Nos últimos anos, tem-se assistido a mudanças fundamentais na forma como a capacidade computacional é gerida. Com o grande aumento da utilização da virtualização a forma como são geridas as máquinas físicas e os modelos de infraestruturas de um centro de dados sofreram grandes alterações. Esta mudança é o resultado de uma procura por uma forma de disponibilizar rapidamente uma VM num ambiente altamente consolidado e com a mínima necessidade de intervenção para a sua gestão.

Estão a ser desenvolvidas novas abordagens às técnicas de virtualização a um ritmo nunca visto. O que leva à existência de um ecossistema altamente volátil com novas plataformas e serviços a serem criados a todo o momento. É possível apreciar a entrega de grandes empresas da industria das tecnologias de informação a problemas como a virtualização de desktops com algum sucesso, mas ignorando completamente o poder de computação que já está presente nos seus clientes. Optando ao em vez, por uma via de alto custo, adquirindo máquinas poderosas e vários softwares. Existe ainda espaço para melhores soluções e para o desenvolvimento de tecnologias que façam uso das capacidades de computação já se encontrem presentes com o mínimo de esforço na sua configuração.

Esta tese foca-se no desenvolvimento de mecanismos de replicação e caching para imagens de maquinas virtuais armazenadas num sistema de ficheiros convencional com a funcionalidade de fazer snapshots. Existem alguns pontos em particular a endereçar: a solução tem que seguir uma arquitectura distribuída e ser totalmente integrada numa solução client-based VDI; tem que funcionar com enormes ficheiros apenas de leitura alguns deles resultantes da criação de snapshots mantendo a característica de manutenção de versões. Este trabalho também incide na exploração dos benefícios de utilizar tal sistema numa rede com uma alta taxa de transferência de dados, em quanto mantem propriedades de alta disponibilidade e escalabilidade suportando um largo conjunto de clientes de forma eficiente.

Contents

List of Figures

List of Tables

Listings

Acronyms

IaaS	Infrastructure as a Service.
VDI	Virtual Desktop Infrastructure.
VM	Virtual Machine.

Chapter 1

Introduction

1.1 Context

The concept of virtualization, despite all the recent discussion, isn't new. In fact, this technology has been around since the 1960s [7], but not until the development of virtualization technologies for the x86 architecture [1] and the introduction of *Intel VT* [23] and *AMD SVM* [5] in the 2000s entered the mainstream as the go-to technology solution for server deployment across many production environments.
With efficient techniques that take advantage of all available resources, and a lowering price point on hardware, an opportunity for the advance of new application models and a revamp in the supporting infrastructure was generated.

However, companies realised that the cost to run a fully fledged *data centre* in-house is unreasonable and a cumbersome task. Not only taking into account the cost of the machines, but factoring in the many requirements like the cooling systems that take care of the heat generated by the running machines, physical security to protect the rooms, fire suppressing systems in case of emergency, people to maintain the infrastructure, all added, result in considerable costs on a monthly basis. Adding to this, the demand for instantaneous access to information and the extensive resources needed to store it does not stop growing.

This fact created an opening for a Infrastructure as a Service (IaaS) [17] model, outsourcing all the responsibilities of storing the data and providing the needed computation resources from third parties, which are experts in maintaining huge data centres and even provide all this in various geographic regions.

With major industry players following this trend, supporting more and more types of services and with an increasing number of customers joining this model, new ways to store the growing number of files have emerged.New file systems with a focus on reliability, consistency, performance, scalability, all in a distributed architecture are essential to a broad range of applications presenting a myriad of workloads.

1.2 Motivation

Virtualization is the pillar technology that allowed for the widespread of the IaaS cloud providers in a economy of scale model. These cloud providers, such as Amazon AWS [4], Microsoft Azure [18] and Google Cloud Platform [12], manage thousands of physical machines all over the globe, with the majority of the infrastructure being multi-tenant oriented.
The sheer magnitude of those numbers leads to an obvious problem. How to store efficiently all this data? Not only there is the need to store client generated data but also manage all the demands of the infrastructure and the many services offered. One approach taken by these companies was the development of their own storage solutions. For instance, Google uses BigTable distributed storage system [8], to store product specific data, and then serve it to users. This system relies on the Google File System underneath to provide a robust solution to store logs and data files, designed to be reliable, scalable and fault tolerance.

One characteristic in particular that stands out and is present in many of today's systems is the use of snapshots with copy-on-write techniques. The adoption of such methods allows for quick copy operations of large data sets but saving resources. At the same time it provides high-availability with read-only copies of the data always ready to use and allowing applications to continue execution of write operations simultaneously. All the above-mentioned properties joined with others such as replication and data distribution, to comprise the fundamentals to what is needed to run a highly distributed and scalable file system. For instance, the duplication of records across multiple machines, not only serves as a security net in case of a misfortune event avoiding having a single point of failure but can also be used to maximise availability and take advantage of network bandwidth.

One of these newer systems that have a significant adoption by the Linux community is the BTRFS [24]. At the start, this file system already adopts an efficient system of snapshots and it has as a primary design principle to maintain an excellent performance in a wide set of conditions. The combination of this file system with replication and partitioning techniques opens the way to a solution that serves the needs of an up to date storage system, consequently having the possibility of being easily integrated into an existing platform, serving a vast number of clients and presenting outstanding performance.

1.3 Project Presentation

This dissertation work is performed in the context of a larger project with the name iCBD, Infrastructure for Client-Based (Virtual) Desktop (Computing) [15], under development at Reditus S.A. in collaboration with DI - FCT/NOVA. The primary objective is to improve in a known model, the client-based Virtual Desktop Infrastructure, developing an infrastructure to support the execution, in a non-intrusive way, of virtualized desktops in conventional workstations.

1.3.1 iCBD Project

There are some leading-edge aspects of the iCBD project which sets it apart from other solutions that already exist. Such the adoption of a diskless paradigm with a remote boot, the way virtual machine images are stored in the platform and the support for a virtualized or native execution on any workstation, depending on the user's choice. [14]

The Remote boot support is offered by HTTP, TFTP, and DHCP servers, and in turn, the image repository servers manage the storage of the VMs templates and the production of instances based on them. To address the process of communication between workstations and the platform it is used the HTTP protocol, providing flexibility and efficiency in the communication of the messages. [2, 14, 16]

It is also interesting to briefly discuss some of the primary objectives of the project, being:

- Offer a work environment and experience of use so close to the traditional one, that there is no disruption for the users when they begin to use this platform.
- Enable centralized management of the entire infrastructure including servers in their multiple roles, storage and network devices from a single point.
- Complete decoupling between users and workstations in order to promote mobility.
- Support the disconnected operation of mobile workstations.

With all the above in account, there is a clear separation from other solutions previously and currently available. As far as we know, no other solution is so comprehensive in the use of the resources offered by workstations whether they are PCs, laptops or similar devices.

1.3.2 Previous Work

There have previously been two dissertations involved in this project. That work has centred in the creation of the instances of virtual machines, more specifically in the creation supported by native snapshot mechanisms of the file system where the templates are stored. This way instead of using the hypervisor itself as a method to provision full or thin clones the work is done by the file system snapshot system.

As is happening now, the two theses have followed two different paths in an attempt to determine which file system best suits these objectives. Being that one used a local file system, the BTRFS, and the other followed the object-based storage path, adopting the CephFS.

1.4 Project Contributions

This work, as a part of a bigger project and building on previous contributions, has as premise a couple of existing technologies in the file systems field to create a replicated and distributed environment capable of storing large files consisting mainly of *VMs* templates and golden images. This work not only focuses on storage management aspects, as also attends the need of being integrated into a larger infrastructure and coexist with a wide variety of other systems.

1.4.1 Main Expected Contributions

The main expected contributions are:

- The study, develop, and evaluate an implementation of a distributed and replicated BTRFS file system for VM storage.
- Implement a server-side caching solution in order to increase availability, improve response time, and enable better management of resources.
- Integrate the solutions described above with the work previously developed and the existing infrastructure
- And finally, carry out a series of tests that lead to a meaningful conclusion and that provide help in the design of the remaining platform.

A detailed view of the planning can be found in **Chapter** 3.3.

1.5 Document Structure

The remnant document is structured as follows:

- *Chapter* **?? Related Work** - This section presents existing technologies and theoretical approaches which were the target of study, such as, storage systems and several of its features, as well as several intrinsic characteristics of virtualization techniques.
- *Chapter 3* **Proposed Work** - In this chapter, there is a presentation of the work plan for the elaboration of this dissertation. Giving also an overview of the solution to develop on the duration of this thesis.

Chapter 2

iCBD - Infrastructure for Client-Based Desktop

The acronym iCBD stands for Infrastructure for Client-Based (Virtual) Desktop (Computing). Is a platform being developed by an R&D partnership between NOVA LINCS, the Computer Science research unit hosted at the Departamento de Informática of Faculdade de Ciências e Tecnologia of Universidade NOVA de Lisboa (DI-FCT NOVA) and SolidNetworks – Business Consulting, LDA part of the Reditus S.A. group. Where the primary goal is to achieve a particular kind of VDI infrastructure, a client based VDI, where client's computations are performed directly on the client hardware opposed to on big and expensive servers.

This chapter will address the central concepts and associated technologies encompassed in this project, particularly:

Section ?? overviews virtualization as a core concept and discuss in particular the data and metadata storage method.

Section ?? studies the principal characteristics of a file system, with emphasis on snapshot techniques and replication and consistency models.

2.1 The Concept

2.2 The Architecture

2.2.1 Boot Layer

2.2.2 Client Layer

2.2.3 Storage Layer

CHAPTER 3

Proposed Work

This chapter firstly presents the technologies that will serve as the basis for the development of the work proposed in the following section. Then is given an overview of the solution and finally, there is a presentation of a detailed action plan.

3.1 Existing Technologies

This section briefly lists the main technologies the work will be built upon.

BTRFS is a modern file system built from scratch, initially designed by Oracle Corporation, based on the copy-on-write principle. This file system aims at implementing advanced features while also focusing on fault tolerance, repair and easy administration. But its most interesting feature is its ability to create snapshots of directories and files. Btrfs uses B + trees as the main structure for storing the data. Everything in this file system, from inodes, to data, directories, along with others, is an object in the B + tree.

Memcached [13] is an open source, high-performance, distributed memory object caching system. It acts as an in-memory key-value store for small arbitrary data (such as strings or objects) from results of database calls, API calls, or page rendering. This tool is constituted by four major components: 1) Client software that has a list of available memcached servers; 2) A client-based hashing algorithm, which chooses a server based on the key; 3) Server software, which stores values with their keys into an internal hash table; 4) Least Recently Used cache which determines when to throw out old data.

3.2 Proposed Solution

This work focuses on the problematic of storing virtual machines in a context of virtual desktop infrastructure, in this sequence the general idea is to introduce a replication and caching system with basis on a local file system. Moreover, this solution should integrate correctly with the existing platform using the part of the infrastructure which is already functional. There are three fundamental steps to the realisation of this project towards accomplishing all the goals.[1]

First, there is a need for some preliminary evaluation. As already mentioned, this project has been underway for quite some time. In this sense, now that it is the moment to change from a single-node paradigm to a multi-node one, even having in mind a multi-site distribution, the demand for a thorough study is in order. This examination is intended to understand in detail the current operation of the storage infrastructure, and what happens when users use the platform asking for files and put some pressure on the system. This phase is of great importance since its results will be decisive for the following design phases and general shape of the solution.

Addressing the second requirement originates the need to tackle the challenge of introducing a caching system to storage servers. In this design, a client of the storage system will first check if what is being requested is within the cache system and only in case of a cache miss will the request be routed to the layer below. The difficulty here is that this system must put up with reasonably large files resulting from the consolidation of snapshots in a distributed environment. There are some solutions currently in the market such as Memcached, where support is more focused on making small files available, but there is the possibility of tweaking the system for the sake of getting it to work with bigger files.

There are some relevant issues when talking about a single-node solution, such as low fault tolerance, a general scaling difficulty, and limited availability. Here the approach focuses on the idea of replicating a BTRFS-based file system and at the same time ensuring the properties just mentioned. Two methodologies can be followed to complete this objective. One is to use an existing middleware and integrate it with the different components of the infrastructure. The second is to build from scratch a solution that fulfils this requirement. At the moment it is not yet decided which of the approaches to take, but this will be one of the main objectives of the design phase of the solution.

We plan to evaluate our work in a setting that includes servers running on a platform provided by Reditus S.A and featuring the complete iCBD solution. Thus being able to conduct a series of benchmarks in a near real-world setting. Such test can be grouped into three parameter categories to study, the required bandwidth, latency presented and the number of input/output operations per second (IOPS). Comparing these categories

[1] At the time of writing, these are the goals that make sense to us. Still, the architecture of this solution is dependent on consultation with Reditus S.A, a discussion that will only occur after the deadline of this document.

according to various types of operations (sequential reading, random reading, sequential writing, random writing) in order to simulate multiple types of workload that may occur in a deployment of this platform. To this end, we expect to be able to use an open-source tool called Flexible I/O Tester (fio) [6] that mitigates the need of writing tailored test cases and allows the use of job files containing the operating environment that needs to be tested. In addition, it is necessary to evaluate the behaviour of the solution in a virtualized environment with several nodes, including fault tolerance correction and general availability.

3.3 Work Plan

In this section the work plan for the elaboration of the dissertation is described. The work plan is divided in four phases and a final one for writing the dissertation. The work is to be carried out in the period between the last week of February and the penultimate week of September. Table 3.1 shows the proposed durations for each of the tasks and Figure **??** depicts a Gantt chart displaying a summary of the described schedule, better explaining how dates overlap.

Task	Time (Weeks)
1) Preliminary Evaluation	3
2) Caching Feature	11
i) Design	4
ii) Implementation	5
iii) Evaluation	2
3) Replication Feature	13
i) Design	4
ii) Implementation	7
iii) Evaluation	2
4) Final Evaluation	6
i) Final Tests	4
ii) Result Validation	2
5) Writing	10

Table 3.1: Work plan summary.

Bibliography

[1] O. Agesen, A. Garthwaite, J. Sheldon, and P. Subrahmanyam. "The Evolution of an x86 Virtual Machine Monitor." In: *SIGOPS Oper. Syst. Rev.* 44.4 (Dec. 2010), pp. 3–18.

[2] N. Alves. "Linked clones baseados em funcionalidades de snapshot do sistema de ficheiros." Master's thesis. Universidade NOVA de Lisboa, 2016.

[4] *Amazon Web Services (AWS) - Cloud Computing Services*. 2017. URL: https://aws.amazon.com/ (visited on 02/05/2017).

[5] AMD. *AMD-V Nested Paging*. Tech. rep. 2008, pp. 1–19. URL: http://developer.amd.com/wordpress/media/2012/10/NPT-WP-1%201-final-TM.pdf.

[6] J. Axboe. *Flexible I/O Tester*. 2017. URL: https://github.com/axboe/fio (visited on 02/15/2017).

[7] J. P. Buzen and U. O. Gagliardi. "The Evolution of Virtual Machine Architecture." In: *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition* (1973), pp. 291–299.

[8] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. "Bigtable: A distributed storage system for structured data." In: *7th Symposium on Operating Systems Design and Implementation (OSDI '06), November 6-8, Seattle, WA, USA* (2006), pp. 205–218.

[12] *Google Cloud Platform*. 2017. URL: https://cloud.google.com/ (visited on 02/05/2017).

[13] D. Interactive. *Memcached*. 2017. URL: https://github.com/memcached/memcached (visited on 02/15/2017).

[14] P. Lopes. *Proposta de Candidatura ao programa P2020*. Tech. rep. DI-FCT/NOVA, Reditus S.A, 2015, pp. 1–26.

[15] P. Lopes, N. Preguiça, P. Medeiros, and M. Martins. "iCBD: Uma Infraestrutura Baseada nos Clientes para Execução de Desktops Virtuais." In: *Proceedings CLME2017/VCEM 8º Congresso Luso-Moçambicano de Engenharia / V Congresso de Engenharia de Moçambique* (2017), pp. 13–18.

[16] E. Martins. "Object-Base Storage for the support of Linked-Clone Virtual Machines." Master's thesis. Universidade NOVA de Lisboa, 2016.

[17] P. Mell and T. Grance. “The NIST definition of Cloud Computing.” In: *NIST Special Publication* 145 (2011), p. 7.

[18] *Microsoft Cloud Computing Platform and Services*. 2017. URL: https://azure.microsoft.com/ (visited on 02/05/2017).

[23] M. Righini. *Enabling Intel Virtualization Technology Features and Benefits*. Tech. rep. 2010, pp. 1–9. URL: https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/virtualization-enabling-intel-virtualization-technology-features-and-benefits-paper.pdf.

[24] O. Rodeh, J. Bacik, and C. Mason. “BTRFS: The Linux B-Tree Filesystem.” In: *ACM Transactions on Storage* 9.3 (2013), pp. 1–32.