

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Title of the Dissertation

Name of the Author

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Name of the Supervisor (Title)

June 8, 2012

Title of the Dissertation

Name of the Author

Mestrado Integrado em Engenharia Informática e Computação

June 8, 2012

Abstract

Here goes the abstract written in English.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vehicula lorem commodo dui. Fusce mollis feugiat elit. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec eu quam. Aenean consectetur odio quis nisi. Fusce molestie metus sed neque. Praesent nulla. Donec quis urna. Pellentesque hendrerit vulputate nunc. Donec id eros et leo ullamcorper placerat. Curabitur aliquam tellus et diam.

Ut tortor. Morbi eget elit. Maecenas nec risus. Sed ultricies. Sed scelerisque libero faucibus sem. Nullam molestie leo quis tellus. Donec ipsum. Nulla lobortis purus pharetra turpis. Nulla laoreet, arcu nec hendrerit vulputate, tortor elit eleifend turpis, et aliquam leo metus in dolor. Praesent sed nulla. Mauris ac augue. Cras ac orci. Etiam sed urna eget nulla sodales venenatis. Donec faucibus ante eget dui. Nam magna. Suspendisse sollicitudin est et mi.

Fusce sed ipsum vel velit imperdiet dictum. Sed nisi purus, dapibus ut, iaculis ac, placerat id, purus. Integer aliquet elementum libero. Phasellus facilisis leo eget elit. Nullam nisi magna, ornare at, aliquet et, porta id, odio. Sed volutpat tellus consectetur ligula. Phasellus turpis augue, malesuada et, placerat fringilla, ornare nec, eros. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus ornare quam nec sem mattis vulputate. Nullam porta, diam nec porta mollis, orci leo condimentum sapien, quis venenatis mi dolor a metus. Nullam mollis. Aenean metus massa, pellentesque sit amet, sagittis eget, tincidunt in, arcu. Vestibulum porta laoreet tortor. Nullam mollis elit nec justo. In nulla ligula, pellentesque sit amet, consequat sed, faucibus id, velit. Fusce purus. Quisque sagittis urna at quam. Ut eu lacus. Maecenas tortor nibh, ultricies nec, vestibulum varius, egestas id, sapien.

Phasellus ullamcorper justo id risus. Nunc in leo. Mauris auctor lectus vitae est lacinia egestas. Nulla faucibus erat sit amet lectus varius semper. Praesent ultrices vehicula orci. Nam at metus. Aenean eget lorem nec purus feugiat molestie. Phasellus fringilla nulla ac risus. Aliquam elementum aliquam velit. Aenean nunc odio, lobortis id, dictum et, rutrum ac, ipsum.

Ut tortor. Morbi eget elit. Maecenas nec risus. Sed ultricies. Sed scelerisque libero faucibus sem. Nullam molestie leo quis tellus. Donec ipsum. Nulla lobortis purus pharetra turpis. Nulla laoreet, arcu nec hendrerit vulputate, tortor elit eleifend turpis, et aliquam leo metus in dolor. Praesent sed nulla. Mauris ac augue. Cras ac orci. Etiam sed urna eget nulla sodales venenatis. Donec faucibus ante eget dui. Nam magna. Suspendisse sollicitudin est et mi.

Phasellus ullamcorper justo id risus. Nunc in leo. Mauris auctor lectus vitae est lacinia egestas. Nulla faucibus erat sit amet lectus varius semper. Praesent ultrices vehicula orci. Nam at metus. Aenean eget lorem nec purus feugiat molestie. Phasellus fringilla nulla ac risus. Aliquam elementum aliquam velit. Aenean nunc odio, lobortis id, dictum et, rutrum ac, ipsum.

Ut tortor. Morbi eget elit. Maecenas nec risus. Sed ultricies. Sed scelerisque libero faucibus sem. Nullam molestie leo quis tellus. Donec ipsum.

Resumo

O Resumo fornece ao leitor um sumário do conteúdo da dissertação. Deverá ser breve mas conter detalhe suficiente e, uma vez que é a porta de entrada para a dissertação, deverá dar ao leitor uma boa impressão inicial.

Este texto inicial da dissertação é escrito no fim e resume numa página, sem referências externas, o tema e o contexto do trabalho, a motivação e os objectivos, as metodologias e técnicas empregues, os principais resultados alcançados e as conclusões.

Este documento ilustra o formato a usar em dissertações na Faculdade de Engenharia da Universidade do Porto. São dados exemplos de margens, cabeçalhos, títulos, paginação, estilos de índices, etc. São ainda dados exemplos de formatação de citações, figuras e tabelas, equações, referências cruzadas, lista de referências e índices. É usado texto descartável, *Loren Ipsum*, para preencher a dissertação por forma a ilustrar os formatos.

Seguem-se umas notas breves mas muito importantes sobre a versão provisória e a versão final do documento. A versão provisória, depois de verificada pelo orientador e de corrigida em contexto pelo autor, deve ser publicada na página pessoal de cada estudante/dissertação, juntamente com os dois resumos, em português e em inglês; deve manter a marca da água, assim como a numeração de linhas conforme aqui se demonstra.

A versão definitiva, a produzir somente após a defesa, em versão impressa (dois exemplares com capas próprias FEUP) e em versão eletrónica (6 CDs com "rodela" própria FEUP), deve ser limpa da marca de água e da numeração de linhas e deve conter a identificação, na primeira página, dos elementos do júri respetivo. Deve ainda, se for o caso, ser corrigida de acordo com as instruções recebidas dos elementos júri.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vehicula lorem commodo dui. Fusce mollis feugiat elit. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec eu quam. Aenean consectetur odio quis nisi. Fusce molestie metus sed neque. Praesent nulla. Donec quis urna. Pellentesque hendrerit vulputate nunc. Donec id eros et leo ullamcorper placerat. Curabitur aliquam tellus et diam.

Ut tortor. Morbi eget elit. Maecenas nec risus. Sed ultricies. Sed scelerisque libero faucibus sem. Nullam molestie leo quis tellus. Donec ipsum. Nulla lobortis purus pharetra turpis. Nulla laoreet, arcu nec hendrerit vulputate, tortor elit eleifend turpis, et aliquam leo metus in dolor. Praesent sed nulla. Mauris ac augue. Cras ac orci. Etiam sed urna eget nulla sodales venenatis. Donec faucibus ante eget dui. Nam magna. Suspendisse sollicitudin est et mi.

Phasellus ullamcorper justo id risus. Nunc in leo. Mauris auctor lectus vitae est lacinia egestas. Nulla faucibus erat sit amet lectus varius semper. Praesent ultrices vehicula orci. Nam at metus. Aenean eget lorem nec purus feugiat molestie. Phasellus fringilla nulla ac risus. Aliquam elementum aliquam velit. Aenean nunc odio, lobortis id, dictum et, rutrum ac, ipsum.

Ut tortor. Morbi eget elit. Maecenas nec risus. Sed ultricies. Sed scelerisque libero faucibus sem. Nullam molestie leo quis tellus. Donec ipsum.

Acknowledgements

Aliquam id dui. Nulla facilisi. Nullam ligula nunc, viverra a, iaculis at, faucibus quis, sapien. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur magna ligula, ornare luctus, aliquam non, aliquet at, tortor. Donec iaculis nulla sed eros. Sed felis. Nam lobortis libero. Pellentesque odio. Suspendisse potenti. Morbi imperdiet rhoncus magna. Morbi vestibulum interdum turpis. Pellentesque varius. Morbi nulla urna, euismod in, molestie ac, placerat in, orci.

Ut convallis. Suspendisse luctus pharetra sem. Sed sit amet mi in diam luctus suscipit. Nulla facilisi. Integer commodo, turpis et semper auctor, nisl ligula vestibulum erat, sed tempor lacus nibh at turpis. Quisque vestibulum pulvinar justo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam sed tellus vel tortor hendrerit pulvinar. Phasellus eleifend, augue at mattis tincidunt, lorem lorem sodales arcu, id volutpat risus est id neque. Phasellus egestas ante. Nam porttitor justo sit amet urna. Suspendisse ligula nunc, mollis ac, elementum non, venenatis ut, mauris. Mauris augue risus, tempus scelerisque, rutrum quis, hendrerit at, nunc. Nulla posuere porta orci. Nulla dui.

Fusce gravida placerat sem. Aenean ipsum diam, pharetra vitae, ornare et, semper sit amet, nibh. Nam id tellus. Etiam ultrices. Praesent gravida. Aliquam nec sapien. Morbi sagittis vulputate dolor. Donec sapien lorem, laoreet egestas, pellentesque euismod, porta at, sapien. Integer vitae lacus id dui convallis blandit. Mauris non sem. Integer in velit eget lorem scelerisque vehicula. Etiam tincidunt turpis ac nunc. Pellentesque a justo. Mauris faucibus quam id eros. Cras pharetra. Fusce rutrum vulputate lorem. Cras pretium magna in nisl. Integer ornare dui non pede.

The Name of the Author

*“You should be glad that bridge fell down.
I was planning to build thirteen more to that same design”*

Isambard Kingdom Brunel

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation and Objectives	2
1.3	Dissertation Structure	2
2	State of the Art	5
2.1	Virtualization and Virtual Machines	5
2.1.1	Hypervisors	8
2.2	Grid Computing	8
2.3	Cloud Computing	12
2.3.1	Utility Computing	15
2.4	Grids VS. Clouds	16
2.5	Developments, Applications and Services	17
2.5.1	AWS - Amazon Web Services	18
2.5.2	Google Cloud - Google's App Engine	19
2.5.3	Microsoft Azure	20
2.6	FEUP's Computing System	20
2.6.1	OpenNebula	20
2.6.2	OpenStack	21
2.6.3	Clusters	21
2.7	ISO Images	23
2.8	Introdução	24
2.9	Secção Exemplo	24
2.9.1	Sub-secção Exemplo	25
2.9.2	Sub-secção Exemplo	25
2.10	Resumo ou Conclusões	26
3	The problem and the approaches	27
3.1	Problem Description	27
3.2	Context - A birdseye view on a bigger project	27
3.3	Secção Exemplo	28
3.3.1	Exemplo de Figura	29
3.3.2	Exemplo de Tabela	29
3.4	Secção Exemplo	31
3.5	Resumo e Conclusões	31

CONTENTS

4	Implementação	33
4.1	Secção Exemplo	33
4.2	Mais uma Secção	33
4.3	Resumo ou Conclusões	34
5	Conclusões e Trabalho Futuro	35
5.1	Satisfação dos Objectivos	35
5.2	Trabalho Futuro	36
A	Loren Ipsum	37
A.1	O que é o <i>Loren Ipsum</i> ?	37
A.2	De onde Vem o Loren?	37
A.3	Porque se usa o Loren?	38
A.4	Onde se Podem Encontrar Exemplos?	38
	References	39

List of Figures

3.1	Arquitectura da Solução Proposta	29
-----	--	----

LIST OF FIGURES

List of Tables

3.1	Uma Tabela Simples	30
3.2	Uma Tabela Mais Complicada	31

LIST OF TABLES

Abbreviations

CICA	Centro de Informática Prof. Correia de Araújo
CPU	Central Processing Unit
FEUP	Faculdade de Engenharia da Universidade do Porto
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
PHP	PHP: Hypertext Processor
VD	Virtual Disk
VM	Virtual Machine
VO	Virtual Organization
VW	Virtual Workspace
WWW	<i>World Wide Web</i>

Chapter 1

2 Introduction

High performance computing describes the ability of using parallel processing in order to perform
4 advanced application programs with a great deal of efficiency, reliability and quickness. [?]

In this document is presented a study that analyzes two community projects on creating and
6 managing private Cloud infrastructures (along with some of the technologies that support these
projects) and if and how they can be implemented in FEUP - Faculty of Engineering of the Uni-
8 versity of Porto.

This first chapter introduces a brief technological and situational context, as well as the mo-
10 tivation behind the choice of this subject and the objectives set. The document's structure is also
shown.

12 1.1 Context

Leonard Kleinrock (part of the team that developed Arpanet, an early seed for the Internet) said in
14 1969:

“As [...] computer networks [...] grow and become sophisticated, we will probably see
16 the spread of ‘computer utilities’ which, like present electric and telephone utilities,
will service individual homes and offices around the country.” [BYV⁺09]

18 Confirming Kleinrock's prediction, computing is migrating in a direction where people de-
velop software for an incredible amount of people so it can be used as a service, instead of running
20 said software on their personal computers. Different providers such as Amazon, Google, IBM
and Sun Microsystems are now establishing data centers dedicated to hosting Cloud Computing¹
22 applications spread around the world in order to ensure redundancy and reliability in case one of
the datacenters fails.

24 User requirements for Cloud services are complex and varied, so service providers need to
know they can be flexible when delivering those services at the same time they keep the users
26 clear from the infrastructure on which those services stand.

¹Using multiple server computers via a digital network as if they were a single computer.

Computing services are available instantly when anyone needs them and the consumers only required to pay the providers when they actually access and use those resources. Consumers no longer have the need to invest in and maintain complex IT infrastructures and software developers are facing new challenges. They must create custom made software that will be used as a service, instead of the traditional practice of installing the software in the users' machines. Some people state this is the era of pervasive computing, where computation and information are available all the time. [McF]

Having this in mind, FEUP has started developing a private cloud project at its Informatics center (CICA - Centro de Informática Prof. Correia de Araújo). As it will be discussed in greater detail in Chapter 3 - [The problem and the approaches](#) - this document reports the work realized on the front-end of the private cloud project.

1.2 Motivation and Objectives

Some computing infra-structures, namely Grids² and Clusters³, can be rather inflexible when compared to Clouds, as the latter are supposed to allow the user to take advantage of a myriad of services, and not just computing power. [Sun]

However, as powerful as these infra-structures can be, they can be deemed useless if people who need to work with them, cannot do it because they have no knowledge of the technologies. As such, this type of issue causes a lack of growth in the use of FEUP's computing system.

This project aims at increasing the usability of the current computing system that exists at FEUP and with this, increase its usage and stop the lack of growth. In order to achieve this goal, it was purposed that a web portal would be developed which would simplify the access to the system. This portal would have a list of software packets and Linux distributions that the user could choose from and create an ISO image which would run the investigator's computing job.

1.3 Dissertation Structure

This Dissertation is structured as follows:

Chapter 1: “[Introduction](#)”

Chapter 2: “[State of the Art](#)”

Chapter 3: “[The problem and the approaches](#)”

Chapter 4: “[Implementação](#)”

Chapter 5: “[Conclusões e Trabalho Futuro](#)”

Chapter 1 (this chapter) - Introduction Chapter 2 - State of the art Chapter 3 - Problem statement

Chapter 4 - Implementation Chapter 5 - Conclusions

²Distributed systems that are loosely coupled, heterogeneous and geographically dispersed and act together to perform very large tasks.

³Group of linked computers working closely together as if they were a single machine.

Introduction

Current Grid Computing infrastructures are generally not very flexible when it comes to the users' needs. As such, whenever it is required, the user must adapt its code to the infrastructures specifications. On the other hand, Cloud Computing is associated with an extreme flexibility allowing the infrastructure to adapt itself to the users' requirements. Another aspect present in Cloud Computing but non-existent in Grid Computing is the Quality of Service factor, where a user can submit a job according to a certain cost or deadline. Furthermore, there is also the elasticity component, something that is not available in Grid Computing technologies but is inherent to Cloud Computing, and is one of its flagships that may be able to cross over to grid infrastructures. Having this in mind, this project consisted in the development of a web environment for the creation of virtual high performance computing infrastructures and the analysis of the impact these environments have in the usage of the said infrastructures.

This document covers the work undergone regarding the study, design and implementation of the project titled:

Web System For Creating And Managing Virtual High Performance Computing Environments throughout the course of two six month semesters, the first dedicated to the study of the subject and the second to the implementation of the solution.

Introduction

Chapter 2

2 State of the Art

This chapter presents a bibliographic review on the subjects covered by this project. Firstly, the concepts of Virtualization, Grid and Cloud Computing are presented, as well as a comparison between these two areas. There is also an analysis on existing developments, applications and projects on the area of Cloud and Grid Computing, with greater focus on two community driven projects (OpenStack and OpenNebula), since they are currently in use at FEUP. Some of the computing technologies in use at FEUP are also presented and discussed.

2.1 Virtualization and Virtual Machines

Throughout this project, Virtual Machines (VMs) are mentioned in great amount and as such, they deserve a special section.

Certain problems arise when the requirements of different virtual organizations (VOs) that need to use the same resources are in conflict or are incompatible with site policies. The software available on clusters cannot guarantee isolation of different communities and maintain resource availability while ensuring good utilization of those said resources. This is where Virtual Machines (VMs) come into play. They are emulations of lower layers of computer abstractions on behalf of the higher layers and allow the isolation of the applications from the hardware and neighbour VMs and customizing the platform so it suits the user's needs. [FFK⁺06, ZKFF05]

Virtualization benefits include an improvement in fault isolation and independence from guest VMs, performance isolation and simplifying the migration of VMs across different physical machines. These benefits enable VMs to share pools of platform and data center resources. [NS07]

The ability to serialize and migrate the state of a VM paves the way for better load balancing and improved reliability that cannot be achieved with traditional resources. Deploying virtual clusters - set of VMs configured to behave as a cluster and intended to be scheduled on a physical resource at the same time [ZKFF05] - of diverse topologies requires the ability to deploy many VMs in a coordinated manner so that sharing of infrastructure, such as disks and networking, can be properly configured. This can become more costly than the deployment of single VMs.

In order to understand more, it is necessary to define virtual workspaces (VWs). These are an aggregation of an execution environment and the resources allocated to that specific environment. It is described by the workspace metadata, containing all the information needed for deployment. An atomic workspace, representing a single execution environment, specifies the data that must be obtained and the deployment information that must be configured on deployment. It is also needed to specify a requested resource allocation, something that describes how much of each resource should be allocated to the workspace.

These atomic workspaces can then be combined to form what is called a virtual cluster. Foster et al propose an aggregate workspace that contains one or more workspace sets - atomic workspaces with the same configuration. Cluster descriptions can be defined in ways that atomic workspaces can be constructed flexibly into more complex structures, organizing at the same time the infrastructure sharing between the virtual nodes. This deployment enables the user to specify different resource allocations for different members of aggregates defined like this. Foster et al consider that the trade-off they obtained is acceptable, as the slowdown suffered was of about 5% and considering that virtual machines offer unprecedented flexibility in terms of matching clients to resources. [FFK⁺06]

Zhang et al also approached the virtual cluster theme in an article written with both Foster and Freeman, where they combine virtual clusters with Grid technology. The authors only considered two types of node within the cluster: head-nodes and worker nodes. They optimized the loading of the virtual images through image cloning (only transferring one image for all the worker nodes and one image for the head-node, therefore cloning all the worker node images at either staging or deployment time) and they considered that the cost of virtual cluster deployment and management is a good justification for expecting that they may be used for VOs for large groups of short jobs and single long-running jobs. They also found that the cost of running batch jobs in a virtual cluster was very acceptable. [ZKFF05]

Katarzyna Keahey and Tim Freeman introduce the term *contextualization* in order to describe the process of quickly deploying fully configured images and adapt them to their deployment context, for single VMs. The authors understand *contextualization* as the process of adapting an appliance to its deployment context (an appliance defining an environment as an abstraction independent of its deployment). They are deployed dynamically and are potentially associated with a different context. According to the authors, they can also fulfill three different roles:

1. Appliance providers - they configure environments, maintain them and guarantee their consistency;
2. Resource providers - they provide resources with limited configuration requirements that are designed to support appliances but no longer to provide end-user environments for multiple communities;
3. Appliance deployers - they coordinate the mapping of appliances onto available resource platforms and information exchange between groups of appliances to enable them to share information. [KF08]

There are different types of approaches, since providers can have the applications running inside VMs or provide access to the VMs as a service (Amazon Elastic Compute Cloud), enabling the users to install their own applications. With virtualization, companies are trying to save power by getting the most of what they consume. Running several operating systems inside one machine, they can run independently and CPU idle time is kept to a minimum.[\[Wei07\]](#)

Virtualized computing clusters offer the advantage of being able to transform themselves to the user's needs. However, as pointed out by Nishimura et al, previous work has shown that the system does not scale when increasing the number of VMs and their detailed configuration is not allowed. To counter this issue, Nishimura et al propose a new way of managing virtual clusters so that a flexible and fully-customizable system integration by creating VMs on-the-fly is achieved.

The authors also propose the creation of *virtual disk caches* (VD caches), in order to reduce software installation time. This VD cache is created when a user requests it and is automatically destroyed to keep the total cache size within the given space. What the authors did was that when an installation request is made by the user, the system selects physical resources to host a virtual cluster for the request, instantiates a set of VMs and installs the operating system and other requested software to them. The experiments conducted by the authors using a prototype implementation showed that installing a 190-node virtual cluster can be done in 40 seconds, indicating that the installation of a 1000-VM could be done in under two minutes.[\[NMM07\]](#)

Nathuji and Schwan addressed the issue of integrating power management mechanisms and policies with the virtualization techniques deployed in virtual environments. They propose the *VirtualPower* approach which aims to control and synchronize the effects of the power management policies applied by the VMs to the virtualized resources.

The authors propose this approach having in mind the current limitations in battery capacities and the power delivery and cooling limitations existent in data centers when they try to handle the constant demands of performance and scalability. The authors state that their approach can exploit the hardware power scaling and the methods that control the power consumption of the underlying platforms. It takes guest VMs' power management policies and coordinates them through the system in order to achieve the objectives.

The power management actions are encoded as a set of rules, these being based on a set of mechanisms which serve as a base to implement the power management methods. This approach aimed to present guest VMs with a set of power states and then use the state changes requested by the VMs as inputs to virtualization-level management policies, including those to use specific platforms and their power management capabilities, along with policies that take into consideration goals derived from the applications running through the whole system and from global constraints, such as rack-level limitations on maximum power consumption.

Their findings showed that it is possible to respond to specific power management goals and policies implemented in guest VMs without a need for application specificity to be established at the virtualization level. [\[NS07\]](#)

It is extremely important to discuss *OpenNebula*, as this project will interact with it and as such, it is approached in a later section. [2.6.1](#)

2.1.1 Hypervisors

An hypervisor is a piece of software that emulates the functioning of certain hardware, a process called *Hardware Virtualization*. KVM - Kernel-based Virtual Machine - an open-source virtualization software¹ is used on the back-end of the project..

The following features for KVM were identified:

- Virtualization using hardware virtualization extensions, such as Inter-VT and AMD-V, thus enabling faster virtualization;
- Symmetric Multi Processor emulation - enables multiprocessor hardware emulation;
- Live migration of VMs between hosts, allowing VM relocation without downtime;
- Paravirtualized networking and block devices, which enables faster emulation of those devices. [Car11]

2.2 Grid Computing

Buyya et al believe that Grid computing facilitates the sharing, selection and aggregation of geographically dispersed resources, be it supercomputers, storage systems, data sources or even special assets owned by organizations for solving large-scale resource-intensive problems in different areas of expertise, and that was Grid computing's motivation. Buyya also created a definition for "Grid" at the 2002 Grid Planet conference held in San Jose, United States:

“A Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed 'autonomous' resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements.”[BYV⁺09]

Ian Foster, one of the most revisited authors regarding Grid computing, states that the “Grid” must be looked upon in respect of the applications it contains, the business value it generates and the scientific results it is capable of returning, instead of its architecture. Carl Kesselman and Ian Foster wrote the following definition in their book “The Grid: Blueprint for a New Computing Infrastructure”:

“A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities.”[KF98]

Foster and Steve Tuecke redefined the definition, this time referring social and policy issues, affirming that Grid computing is related to resource sharing and problem solving in a coordinated

¹Nuno Cardoso compared XEN and KVM, but came to the conclusion that neither offered an advantage over the other, so KVM was chosen due to its simplistic installation process.

manner and that these occur in dynamic, multi-institutional virtual organizations, the aspect to remember being the power to do something with the result. The authors also stated that they are preoccupied with the “direct access to computers, software, data and other resources.”

As such, Foster proposes (as pointed out by the title of his article) a three point checklist that defines what a Grid system should be:

- The Grid should coordinate resources that are not subject to centralized control – Integration and coordination of both users and resources that live within different domains;
- The Grid should use standard, open, general-purpose protocols and interfaces, as this will allow the establishment of dynamic resource-sharing arrangements and the creation of something more than an agglomerate of incompatible and non-interoperable distributed systems;
- The Grid should deliver nontrivial qualities of service, such as response time, throughput, availability, security, co-allocation of multiple resource types to meet complex user demands, resulting in the utility of the combined system to be greater than just the sum of its parts.

Foster also states that the Web is not a Grid, as though its general-purpose protocols support the access to distributed resources; they do not coordinate their use to deliver qualities of service.

Some large-scale Grid deployments inside the scientific community abide by the three points described by Foster, such as NASA’s Information Power Grid and the TeraGrid, which will link major U.S. academic sites, as they integrate resources from several institutions, use open and general-purpose protocols (Globus Toolkit, which will be discussed in further details later on this report) to negotiate and manage sharing and they address multiple dimensions of the quality of service, such as security, reliability and performance.[Fos02]

Stockinger started a survey where he contacted over 170 Grid researchers globally spread in order to obtain a general feel on how the Grid was being defined. The results showed that the Grid infrastructure should provide a set of capabilities, such as:

- Description of available resources, what they are capable of doing and how they are connected;
- Visibility into the state of resources, including notifications and logging of significant events and state transitions;
- Assurance of the quality of service across an entire set of resources for the lifetime of their use by an application;
- Provision, life-cycle management and decommissioning of allocated resources;
- Accounting and auditing of the service;
- Security.

The results also showed that a Grid should have a set of characteristics, including:

State of the Art

- Collaboration - sharing resources in a distributed manner;
- Aggregation - the Grid is more than just the sum of all parts; 2
- Virtualization - Services are provided in a way that the complexity of the infrastructures is hidden from the end-user through the creation of an abstract "layer" between clients and resources; 4
- Heterogeneity; 6
- Decentralized control, Standardization and Interoperability - supporting Ian Foster's definition; 8
- Access transparency - users should be able to access the infrastructure without having to preoccupy themselves how they are doing it; 10
- Scalability;
- Reconfigurability; 12
- Security - specially since the systems are often spread through multiple administrative domains. [Sto07] 14

The members of the EGEE (Enabling Grids for E-science Project) also state that their Grid abides by some of the characteristics mentioned above, namely "decentralized control", "heterogeneity" and "collaboration" [B08]. Their Grid is described in greater detail in the "Grids VS Clouds" section below. 16 18

Bote-Lorenzo et al also identified some core Grid characteristics that coincide with Stockinger and Ian Foster's definitions. These include scalability, heterogeneity, resource coordination and dependable, consistent and pervasive access. They propose the following definition for a Grid: 20

"... large scale geographically distributed hardware and software infrastructure composed of heterogeneous networked resources owned and shared by multiple administrative organizations which are coordinated to provide transparent, dependable, pervasive and consistent computing support to a wide range of applications. these applications can perform either distributed computing, high throughput computing, on-demand computing, data-intensive computing, [...]"[BLDGS04] 22 24 26

Baker et al say that the Grid has evolved from something static and carefully configured, to what has been witnessed in the past years, where it became a seamless and dynamic virtual environment, capturing the attention from the industry and thus making an impact on the Grid's architecture and protocols and standards. 28 30

The authors also describe a few standards and organizations that have been actively present in the Grid's environment over the past years. These include the Global Grid Forum (GGF), a 32

community-driven set of groups which goal is to develop standards and best practices for wide-area distributed computing. The GGF creates a group of documents that provide some information to the Grid community, dividing its efforts into several categories, including architecture, data and security.

The authors also approach the World Wide Web Consortium (W3C), an international organization created to promote common and interoperable protocols. This organization was responsible for creating the first Web Services specifications in 2003, such as SOAP and the Web Services Description Language (WSDL). According to the authors, the most important Grid standard to appear recently is the Open Grid Services Architecture (OGSA), which goal is to define a common, standard, and open architecture for Grid-based applications. It was announced by the GGF at the Global Grid Forum in 2002 and in March 2004 it was declared by the GGF to be the flagship architecture.[\[BAFB05\]](#)

Iosup, Dumitresco and Epema analyzed four Grid implementations and the differences on their workload:

- Firstly, they covered the LHC Computing Grid, which testbed has 25,000 (twenty five thousand) CPUs and 3 PetaBytes of storage. Jobs are managed and routed to resources via a Resource Broker, which tries to conduct the job matchmaking and balance the workloads at the global level. The site used by the authors had around 880 CPUs;
- Secondly, they looked at the Grid3 testbed, representing a multivirtual organization environment that sustains production level services required by various physics experiments. It is composed by more than 30 sites with 4500 (four thousand five hundred) CPUs;
- Thirdly, they analysed the TeraGrid system - used for scientific research - which has over 13,6 TeraFLOPS of computing power and can store 450 TeraBytes of data;
- Finally, they reviewed the DAS-2 environment, which has 400 CPUs spread over five Dutch Universities and its workload ranges from single CPU jobs to very complex ones. These can be submitted either via the local resource managers or to Grid interfaces that communicate with them.

They discovered that while Grid research focuses on complex application types, most of the applications encountered were extremely easy to run in parallel (embarrassingly parallel applications).

The authors identified two large problems, a scale (origin and size of the data that must be collected) and a methodological (missing components of the information) problem. In order to address the first problem, the information should ideally come from three different sources:

- Local and Grid scheduler - without these logs, job arrival and dependency information can be lost and an analysis of site-related performance metrics cannot be done;
- Grid AAA (authentication, authorization and accounting) modules - these modules provide the information regarding the link between jobs and their owners;

- Monitoring systems - without the information these systems provide, it is impossible to understand how the applications are running within the Grid and to quantify the system utilization. 2

The authors concluded that a small number of VOs and users control the workload in terms of submitted jobs and consumed resources, system evolution can appear at the system, VO and user level and should be considered when provisioning resources. [IDE⁺06] 4 6

Malcolm Atkinson from the National e-Science Center in the United Kingdom, says the following: 8

“With Web Services we allow a thousand flowers to bloom. With a Grid we organize the planting and growth of a crop of plants to make harvesting easier.” [Sto07] 10

Iosup et al end their article with the following quote:

“[...] conclude that Grids are not yet utilized at their full capacity.” [IDE⁺06] 12

which serves as a conclusion for this section.

2.3 Cloud Computing 14

Similarly to the Grid, many definitions arise when one talks about the Cloud. Presently, it is considered normal to obtain access to content spread over the Internet without a reference to the hosting infrastructure that lies underneath it. This infrastructure is made of data centers that are being monitored by service providers. 16 18

Buyaa et al state that Cloud computing extends this paradigm in where the capabilities of the applications are viewed as complex services that can be accessed over a network. The authors also believe that the Cloud is an infrastructure from where businesses and users can access applications from anywhere in the world anytime they want. Cloud computing’s services need to be reliable, scalable and sufficiently autonomic to support omnipresent access, dynamic discovery and they need to support composability, as they must permit to be reassembled and selected in any order to comply to the user’s requirements. 20 22 24

The authors have a definition of their own: 26

“A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiation between the service provider and consumers.” [BYV⁺09] 28 30

They believe that Clouds are the new datacenters with hypervisor technologies such as VMs, with services provided on-demand as a personalized resource collection in order to meet the 32

service-level agreement, which should be established *à priori* with a “negotiation” and accessible as a composable service via Web Service technologies.

Vaquero et al state that the paradigm of Cloud computing shifts the infrastructure to the network in order to reduce the costs that are normally associated with the management of hardware and software resources.

Having in mind Gartner’s Hype Cycle², Vaquero et al state that Cloud computing is now in its first stage – Positive Hype – mixing every definition that appears into an overly general term that confuses every single person. The same thing that happened to Grids can be applied here. There are no widely accepted definitions (Foster’s being the most accepted one) and a clear definition can help transmit what it actually is and how businesses can reap benefits from it.

There are many Cloud definitions, but they all focus on certain technological aspects. Thus, Vaquero et al try to analyze all the features of Cloud computing in order to reach a clearer definition.

The authors try to distinguish the different actors and scenarios that can arise:

The actors:

Service Providers make services accessible to Service Users through Internet-based Interfaces. The computing infrastructure is offered “as a service” by the Infrastructure Providers, moving computing resources from the SPs to the IPs, in order to give the firsts flexibility and reduced costs.

The scenarios:

- Infrastructure as a Service – IPs are responsible for the management of a large set of computing resources, such as storing and processing capacity. If they use virtualization, they can split, assign and dynamically resize the resources to build ad-hoc systems as the customers (SPs) demand, by deploying the software stacks that run their services.
- Platform as a Service – Clouds offer an additional abstraction layer – they can provide the software platform where systems run on. The sizing of the hardware resources demanded by the execution of the services is made in a transparent manner. The applications developed are run on the provider’s infrastructure and are delivered through the Internet from the provider’s servers. The Google Apps Engine is a very good example.
- Software as a Service – Cloud systems can host many services that users can be interested in, such as online word processors or even Google Apps. [KG09, VRMCL08]

In the article written by Vaquero et al, many Cloud definitions are gathered. Markus Klems states that the key elements for the Cloud are immediate scalability and the optimization of resources usage, these being brought provided by increased monitoring and automation of resources

²Graphic representation of the maturity, adoption and social application of specific technologies. It has five phases: 1 Product launch that generates interest; 2 - Frenzy of publicity generates over-enthusiasm and unrealistic expectations; 3 - Technologies fail to meet expectations and become unfashionable; 4 - Press stopped to cover the technologies, but some businesses continue to experiment and understand its benefits; 5 - Benefits become widely demonstrated and accepted. Technology becomes stable and evolves.

management. Jeff Kaplan and Reuven Cohen prefer to focus on the business model, paying more attention to the collaboration and pay-as-you-go and reducing the costs of investment. Douglas Gourlay and Kirill Sheynkman define the Cloud as being simple virtualized hardware and software, combined with monitoring and provisioning technologies. [GKC⁺, VRMCL08]

McFedries believes that the basic unit of the Cloud is nothing other than data centers - huge collection of clusters - that can offer a large supply of computing power and storage simply by using whatever resources they can spare.[McF]

Kevin Hartig defines Cloud Computing as being able to access resources and services needed to perform certain tasks with needs that are constantly changing. The application or user requests access from the cloud rather than on a specific endpoint of the network or a resource. The Cloud becomes a virtualization of resources that is both self maintainable and manageable, a view also shared by Jan Pritzker, who focuses his definition on virtualization and on-demand resource allocation. Other authors such as Reuven Cohen, Praising Gaw, Damon Edwards and Ben Kepes (to name a few) are strong believers that Cloud computing is nothing more other than a buzz word, grouping concepts such as deployment, load balancing, provisioning and data and processing outsourcing.[GKC⁺]

Having this in mind, Vaquero et al believe that the Cloud is a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services), these being dynamically reconfigured to adjust to a variable load (scaling) also allowing for an optimum resource utilization. The pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized Service-Level Agreements. The authors also state the set of features that resemble this minimum definition would be scalability, pay-per-use utility model and virtualization. [VRMCL08]

Brian Hayes states in his article that even though the future of Cloud computing is still unclear, there are a few directions in which it can go. One of those directions is Web based services, such as Google Docs or even Photoshop Express. Salesforce.com also offers a variety of online applications and its slogan is actually "No software!".

As mentioned earlier in the report, Amazon.com also ventured into this new paradigm, offering data storage and computing capacity, each of these services being able to expand and contract as the users need (elasticity) and Google has its App Engine, providing hosting on Google server farms.

There is great concern in terms of scalability in the Cloud, as it might be necessary to organize resources so that the program runs flawlessly even though the number of concurrent users increases might arise. Hayes also mentions that Cloud computing raises questions in terms of privacy, security and reliability, since personal documents are being delivered to a third-party service.[Hay08]

Nicholas Carr writes in his book that a shift is happening, where the Cloud is becoming similar (if not equal) to the electric grid, as we can connect to the Cloud and get data, storage space and processing power cheaply and instantly (Utility Computing). [Car09]

Aaron Weiss writes in his article that the Cloud is robust, even self-healing, as it has many sources from where to get the power to recover from whatever accident occurs. Weiss states

that the Cloud is also very power consuming, as roughly 50 percent of the energy it consumes comes from the cooling process alone. Giants such as IBM and Microsoft are also scouting locations where the hydroelectric power is cheaper and greener, so they can establish their cluster centers.[Wei07]

2.3.1 Utility Computing

Computing is going in such a direction that the services that are made available to the user are being done in such a way that computing is becoming equal to traditional utilities such as water, gas, electricity and telephone services.[BYV⁺09]

In an article published by InfoWorld (formerly the Intelligent Machines Journal), Utility computing is mentioned as being a form of Cloud computing, where storage and virtual servers are being offered and can be accessed on demand, such as the services offered by Amazon.com, Sun or IBM. [B08]

IBM Global Services provide the following definition for Utility Computing:

“Utility computing is the on demand delivery of infrastructure, applications, and business processes in a security-rich, shared, scalable, and standards-based computer environment over the Internet for a fee. Customers will tap into IT³ resources - and pay for them - as easily as they now get their electricity and water.” [Rap04]

Utilities have the following characteristics:

- Necessity - Users depend on utility services to fulfill their day-to-day needs. It takes time for distribution networks to spread and costs to decline, as it also takes time for users to adapt to the service. Once they do, the service may grow in importance as users begin to find new ways to reap benefits from it;
- Reliability - The service must be readily available when and where the user requires it, as a temporal or intermittent loss of service may cause several issues to the user. Redundancy must be built into production capacity in order to make up for hypothetical service failures;
- Usability - Users have a “plug-and-play” mentality and they need to feel at ease with whatever feature they are using;
- Utilization rates - Utilities are driven by a necessity to carefully manage utilization rates. User demands for utility services may vary over time and across service regions. This may lead to spikes in utilization of the service and under-utilization in off-peak periods. Service providers must have in mind that how the service is billed may influence how users use that service;

³Information Technologies

- Scalability - As production capacity grows, the unit cost of production shrinks. It might be expected that as the demand for the service rises, the quality of service may decline or vice-versa [Rap04].

Bhattacharya and Vashistha state that utility based computing allows computing resources to be available for a customer on demand, as the customers subscribe to the services of the utility provider and only pay for the quantum of the resources used. This allows any customer to cut down on IT infrastructure spendings as they can simply subscribe to the provider's services and use the computing resources at will, only paying for as much as they use. Typical measures of usage include metered CPU hours and memory space usage [BV08].

Ross and Westerman write in their article that utility computing relies on several important technical capabilities to deliver what it promises - services available on-demand. The authors believe that for most firms, the impact of utility computing will be on the extent and nature of outsourcing. The benefits that can be obtained only enhance the current benefits of IT and business processes outsourcing: lower cost, variable capacity and increased strategic focus. On demand capacity leads to firms to invest less in computing capacity. Advances in autonomic computing may reduce the number of people needed to monitor operations and thus reduce labor costs.

The authors believe that firms will be able to do more with less and will be able to allocate their most strategic resources to their most strategic opportunities [RW04].

2.4 Grids VS. Clouds

As one knows, Grids and Clouds share a few goals, such as reducing computing costs and increasing flexibility and reliability through the use of third-party operated hardware.

Vaquero et al lay out a very comprehensive list of features and discuss the similarities and differences between them. The list includes resource sharing, heterogeneity, virtualization, security, the offer of high level services such as metadata search, the awareness of architecture, dependencies and platform, software workflow, scalability and self management, standardization, payment model and quality of service. The list is shown in Figure ?? which is in Appendix ??.

The authors also believe that Grids are meant to be user friendly, virtualized and automatically scalable utilities, something that steps into the Clouds' path, but they still need to be able to incorporate virtualization techniques in order to obtain some advantages already present in the use of Clouds, like migrability and hardware level scalability.[VRMCL08]

A few members of the Enabling Grids for E-science (EGEE, now part of the European Grid Infrastructure) performed a comparative analysis on Grids and Clouds, focusing two implementations of both: the EGEE project for Grid and the *Amazon Web Service* (AWS) for Cloud, using metrics such as performance, scale, ease of use, costs and functionality, amongst others. The Grid in use by the EGEE runs on gLite, an open source software which had development funding from the EGEE, described in a later section of this document, as it is used in some extent by FEUP's cluster system.

When comparing both EGEE Grid and the *Amazon Web Service*, the authors of the analysis encounter a set of differences and similarities:

- The AWS does not expose how they operate their data centers and how they implement the user interfaces, execute the user requests and maintain their accounting, its back-end is still a grey area;
- The EGEE Grid exposes both user interface as well as the resource interface to permit providers to connect their resources. The AWS hides this second interface;
- The authors assume that on the resource side, both systems work in similar manner, as both cases require a queueing mechanism whether the data center is dispatching a grid job via a batch system or is requested to instantiate a new virtual machine;
- The greatest benefit of the Cloud proposed by Amazon is its interfaces and usage patterns, focused on simplicity;
- Both services are not fail-proof, but the authors consider that a centralized Cloud might not be able to provide the resilience that the distributed nature of EGEE does;
- Grids are typically used for job execution - limited duration execution of a program, part of a larger set of jobs, consuming or producing a significant amount of data. Clouds, even though they support a job usage pattern, they seem to be more often used for long-serving services;
- Amazon bills users for computing resources usage with a minimum of one hour usage. This stops being efficient when dealing with a large number of small jobs;
- Elasticity in the Grid is made by adding worker nodes at a site or adding new sites;
- The complexity in the Cloud is kept server-side, which makes its entry point very low, something that is still considered a goal to achieve for Grids. [B08]

2.5 Developments, Applications and Services

With the shift of the computing industry towards a provision of Platform as Service and Software as a Service, consumers can access resources on-demand without having to preoccupy themselves with time and location, Buyya et al believe that there will be an increasing number of Cloud platforms being developed. [BYV⁺09] One of those platforms is *OpenNebula*, an open-source tool kit for Cloud management. In this report, the Amazon computing service (2.5.1) will also be analysed, as well as *Google's App Engine* (2.5.2) and *Microsoft Azure* (2.5.3). *Rackspace Hosting* and *NASA's OpenStack*

2.5.1 AWS - Amazon Web Services

The *Amazon Web Services* consist of several components, but only two will be taken into consideration in this document, as they are the most relevant to the work discussed: *Amazon's Simple Storage System* (2.5.1.1) and *Amazon's Elastic Computing Cloud* (2.5.1.2).

2.5.1.1 Amazon's Simple Storage Service

The core service for the *Amazon Web Services* is the *Amazon's Simple Storage Service*, that gives the user the power to store large amounts of data in a reliable way which does not hinder its availability. Data is accessed through protocols such as SOAP⁴ and REST⁵, while also being able to be accessible via normal web browsers. The storage model runs on a two-level hierarchy, where the users can create *buckets* and place data *objects* in those buckets. Strings are used as keys for both buckets and objects, thus being able to be easily incorporated in URLs. Users are charged 15 US cents per Gigabyte per month, each user being able to have up to 100 buckets and each can hold up to 5GB of data.[Haz08]

2.5.1.2 Amazon's Elastic Computing Cloud

Physically speaking, the *Elastic Computing Cloud* (EC2) is a large number of computers on which Amazon provides time to paying customers, these computers being spread all over the United States. EC2 is based on the XEN virtualization technology, which allows one physical computer to be shared by several virtual ones, each with its own operating system.

Through the use of virtualization, the users create an image of their software environment using the tools provided. This will be used to create an instance of a machine in Amazon's Cloud. Customers can freely choose configuration templates for their instance and they can create and destroy the instances at will, enabling the software to scale itself to the amount of computing power it needs.[B08, Haz08]

Amazon has released *Elastic IPs* (Static IPs for Dynamic Cloud Computing), which allows the assignment of static IPs to dynamic resources that are deployed via EC2, as well a service that enables users to request EC2 instances to be geographically distributed, as a response to the demand for EC2 IP addresses in a static range for application range for applications like email service hosting, as well as providing a safety net in case the operations of an Amazon Web Services data center go awry.

Amazon provides a variety of ways of requesting the EC2 instances, namely through the use of Web Services, supporting Buyya et al's Cloud definition previously mentioned in the document.

Amazon has also introduced its own performance unit named "EC2 Compute Unit". Since Amazon ventured into the Utility computing field, model it follows differs from the traditional way

⁴Simple Object Access Protocol - Used to exchange information in the implementation of Web Services in computer networks.

⁵Representational State Transfer - Style of software architecture for distributed hypermedia such as the World Wide Web.

developers were formatted to think about CPU resources. Instead of renting a certain processor for several months or years, it is now rented by the hour. One EC2 Compute Unit provides the CPU equivalent of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.[[Ama](#)]

2.5.2 Google Cloud - Google's App Engine

The *Google Cloud's* official name is *App Engine* or *Appengine*. It gives developers the ability to run web applications on Google's infrastructure, the same that is being used by *Google* for *GMail* and *Google Docs*. The Cloud appears to be a platform accessible over the Internet with limitless hardware, the latest software and abundant storage for deploying web applications. The *App Engine* has the following features:

- Automatic horizontal scaling and load balancing;
- APIs⁶ for authenticating users with Google Accounts and for sending emails. No system administration is needed by the user to set up or allow access to these APIs;
- Fully featured Eclipse developed environment that simulates *Google App Engine* on the localhost for development and testing;
- Persistent storage and support for transactions and queries using the standard JDO⁷ and JPA⁸ APIs;
- Generous free quotas, which allow small universities to have access to the same hardware and software as large industries. Each user can have 10 applications created, each with 10 versions, which totals an effective development environment of 100 applications. A free account supports six and a half CPU hours a day, with 1GB of stored data and sending email to 2000 recipients a day and a max of 5 million page views a month;
- It is free, with no contracts to sign, no hardware expense and no system administration costs for maintaining, updating, patching or backing up *App Engine*;
- Eclipse plug-in available for Apple, Linux and Windows, which allows standard debugging using Eclipse debug tools. It provides menu based functionality to automatically upload the application to the Google App Engine;
- Requires no system administration;
- Simple web based, user friendly console.[[HP10](#)]

⁶Application Programming Interface

⁷Java Data Objects

⁸Java Persistence API

2.5.3 Microsoft Azure

Microsoft Azure platform is a cloud computing platform which offers a set of cloud computing services similar to those offered by Amazon Web Services. *Windows Azure Compute* (Microsoft's counterpart to Amazon's EC2), only supports Windows virtual machines and offers a limited variety of instance types when compared with Amazon's EC2. Its instance type configurations and cost scales up linearly from small to extra large and its instances are available in 64 bit x86_64 environments.

It has been speculated that the clock speed of a single CPU core in *Azure's* terminology is approximately 1.5 GHz to 1.7 GHz.[GWQF10] *Windows Azure* enables developers to build, host and scale applications in Microsoft datacenters, not requiring upfront expenses, long term commitment and users only pay for the resources they use. *Windows Azure* relieves the user from the effort of configuring load balancing and failover, is designed to let developers build applications that are continuously available, even if they need software updates and hardware failures occur.[Mic]

2.6 FEUP's Computing System

In this section FEUP's computing system is analysed in detail. The cluster system and the technologies it uses in its management are described and detailed. FEUP's cluster system currently uses three different technologies, *Moab*, *gLite* and *Condor*. The Cloud creation and management tool is also described, as it is a vital part in FEUP's computing system and is the connecting link between the front-end and the back-end.

2.6.1 OpenNebula

OpenNebula is an academic project developed by the *Universidad Complutense Madrid*. It was created in order to control several hosts and machines in a unified environment. It has been released as a software package in *Ubuntu 9.04* and is compatible with both *Xen* and *KVM*, two *platform virtual machines* that emulate the whole physical computer machine. It has its own command-line tools and gives the user different configuration scripts which enable a simple and flexible way to design and manage running virtual machines. *OpenNebula's* architecture is shown in ?? [Pin10].

OpenNebula does not have a built-in utility to create VMs from scratch, but its templates allow the VMs to boot an ISO image, leaving the user with just creating an empty hard disk image.

It provides monitoring capabilities which become rather useful when there is a need to troubleshoot, scale or control resource allocation scenarios. *OpenNebula* exports drivers that communicate directly with the hypervisor (*KVM* - 2.1.1) and return useful data, such as the amount of CPU used, reserved and used memory and network traffic.[CGH09]

Figure ?? sums up the pros and cons of using *OpenNebula*.

2.6.2 OpenStack

OpenStack is a global collaboration of developers and cloud computing technologists producing the open source cloud computing platform for public and private clouds. The aim of this project is to deliver solutions for all types of clouds by being simple to implement, massively scalable and filled with features.

It was founded by *Rackspace Hosting* and NASA⁹ and it has grown to be a global software community of developers collaborating on a standard open source cloud operating system.

OpenStack's mission is to enable any organization to create and offer cloud computing services running on standard hardware.

All of its code is available under the *Apache 2.0* license and as such, anyone can run it, build applications on it or submit changes back to the project.

OpenStack has three major components:

- **Compute** - Designed to provision and manage large networks of virtual machines, creating a redundant and scalable Cloud computing platform. It has built-in control panels, APIs required to create and manage a Cloud, including running instances, managing networks and controlling access through users and projects;
- **Swift** - *OpenStack* Object Storage - Designed to create redundant and scalable object storage using clusters or standardized servers to store PetaBytes of accessible data. It is a long-term storage system for a permanent type of static data that can be retrieved, leveraged and then updated if necessary, such as virtual machine images, photo storage, email storage and backup archiving;
- **Glance** - *OpenStack* Image Service - Provides discovery, registration and delivery services for virtual disk images. It relies on a REST¹⁰ interface for querying information. It currently allows uploads of private and public images which include those created via KVM, VMWare and VirtualBox.

It is a project still in its early stages, as it was first released in October 2010 and is only in its third version [CC].

This project is under close surveillance by CICA as it is viewed as a possible substitute for *OpenNebula* (2.6.1).

2.6.3 Clusters

Three different technologies are currently in use by FEUP's Cluster system: *Moab*(2.6.3.1), *gLite*(2.6.3.2) and *Condor*(2.6.3.3).

⁹North American Space Agency

¹⁰Representational State Transfer

2.6.3.1 Moab Cluster Suite

Moab Cluster Suite is a proprietary tool for high performance computing systems, developed by the company *Cluster Resources*. It has built-in modules for work management, Cluster administration and monitoring, report creation. It is composed by three essential components:

- **Moab Workload Manager** - scheduling and workload management engine;
- **Moab Cluster Manager** - graphical interface for Cluster administration, monitoring and report analysis;
- **Moab Access Portal** - web based portal for job management and submission, directly focused on the end-user.

A resource manager supplies the system with basic functionalities for initiating, stopping, canceling or monitoring jobs. *Moab Workload Manager* uses a resource manager's services to get information about the state of the resources and the node workload. It is also used to manage jobs and to send information on how they should be run and it can be configured to manage more than one resource manager simultaneously. Its composing nodes can be split into three groups:

1. **Master node** - Manages the resources;
2. **Submissive/Interactive nodes** - Allow users to manage and submit jobs into the system;
3. **Computing nodes** - Execute the submitted jobs.

It is also possible to split the nodes into two groups - source and destination nodes. The first ones are the nodes where there users, portals or other systems can submit their jobs and the latter ones are where the jobs are executed. Jobs originate in a source node and are transferred to the destination nodes. Decisions are made in the source nodes, so it is possible to choose which nodes will execute the submitted jobs. Moab also allows for the establishment of connections between several Grid systems, which permit access to additional resources.[\[Pin10, Res\]](#)

2.6.3.2 gLite

gLite consists in a set of components designed with the objective of building a Grid computing infrastructure for resource sharing developed by the project EGEE (Enabling Grids for E-science), also mentioned in an earlier section. *gLite* is based on four core concepts:

- **Computing Element (CE)** - Set of local computational resources, namely a Cluster. It is composed by three components:
 - **Grid Gate** - generic interface for the Cluster who receives jobs and submits it to the Local Resource Management System (LRMS);
 - **Local Resource Management System (LRMS)** - Sends the jobs to the worker nodes for execution;

– **Worker Nodes** - Cluster nodes where jobs are executed.

- 2 • **Storage Element (SE)** - Supplies access to data storage resources;
- 4 • **Information Service (IS)** - Resource research is made through this component, which is also responsible for supplying information regarding resources and their state;
- 6 • **Workload Management System (WMS)** - Receives jobs from users, appropriately allocates a CE, saves jobs states and gets the final results.[[Pin10](#), [CER](#)]

2.6.3.3 Condor

8 *Condor* is a free and open-source workload management system, developed by the *Condor Research Project*. It has built-in job queueing mechanisms, scheduling and priority policies, resource
10 monitoring and management. Users submit jobs to *Condor*, which puts them in a queue, chooses when and where to execute them based on defined policies, carefully monitors their progress and
12 informs the user when the jobs are finished. *Condor* can manage dedicated nodes or harness the CPU energy wasted in workstations that are turned on but unused. If the system detects the
14 machine became suddenly unavailable, *Condor* can migrate the state of the job into a different machine and resume work. It offers an extremely flexible structure to assign resources to jobs, al-
16 lowing these to have specific requisites and resource preferences, as well as enabling the resources to specify preferences over jobs to execute. Each machine from *Condor* can play several roles:

- 18 • **Central Manager** - Machine that collects information and makes the negotiation between resources and resource requests. All resource requests go through the *Central Manager*.
20 There can only be one *Central Manager* in a *Condor* infrastructure;
- 22 • **Execute** - Machine which executes jobs, therefore allowing the network to take advantage of its resources. Any machine can be configured to take this role;
- 24 • **Submit** - Machine responsible for the job reception and submission to the *Central Manager*. Any machine can be configured to take this role;
- 26 • **Checkpoint Server** - Machine which stores checkpoint files for all submitted jobs.[[Pin10](#), [Tea](#)]

2.7 ISO Images

28 unsure.....

Ubuntu's VM Builder? talvez ver mais.

Neste capítulo é descrito o estado da arte e são apresentados trabalhos relacionados para mostrar o que existe no mesmo domínio e quais os problemas em aberto. Deve deixar claro que existe uma oportunidade de desenvolvimento que cobre alguma falha concreta .

O capítulo deve também efectuar uma revisão tecnológica às principais ferramentas utilizáveis no âmbito do projecto, justificando futuras escolhas.

2.8 Introdução

Neste capítulo é ilustrada a utilização de macros \LaTeX para definir entradas no índice remissivo e são feitas diversas referências bibliográficas, usando-se texto de um artigo apresentado na Conferência XATA2006 [?].

Nos últimos tempos têm surgido diversas soluções, apresentadas por empresas do sector Automação de Sistemas para a disponibilização de sistemas *SCADA/DMS* na *Web*.

Aliquam sollicitudin facilisis sapien. Mauris tincidunt tristique diam. Mauris sollicitudin pede at tellus varius volutpat. Integer vel leo. Nunc massa diam, egestas eu, venenatis at, porttitor ac, sapien. Sed magna elit, vulputate in, lacinia sed, lobortis ac, urna. Proin cursus massa id risus. Vestibulum libero. Curabitur venenatis augue. Mauris eu libero eget lectus tempus tempor. In tincidunt, justo in varius adipiscing, ipsum enim gravida massa, eget ornare ante lacus id est. Praesent vitae est ut elit convallis convallis. Aenean tincidunt, purus id consectetur volutpat, sem leo pulvinar libero, nec semper sem purus ultricies nibh [?].

Fusce risus mi, tristique eu, consectetur id, auctor sed, elit. Donec laoreet. Duis consectetur interdum libero. Etiam eu orci. In eu arcu. Fusce luctus diam eget lectus. Duis interdum lacus sed ligula. Proin vestibulum felis eget lacus. Vivamus vestibulum, tellus ut congue viverra, mauris lacus tempor turpis, eu congue nisi magna at dolor. Ut molestie vehicula libero. Praesent in neque sed risus tempus ornare. Donec hendrerit, erat eu semper aliquam, pede nulla dapibus risus, ut pretium orci pede et neque. Etiam eget tortor a metus convallis viverra. Quisque eget nisi sed orci facilisis interdum. Aliquam non felis.

2.9 Secção Exemplo

Scalable Vector Graphics é uma linguagem em formato XML que descreve gráficos de duas dimensões. Este formato padronizado pela W3C (*World Wide Web Consortium*) é livre de patentes ou direitos de autor e está totalmente documentado, à semelhança de outros W3C standards [?].

Sendo uma linguagem XML, o SVG herda uma série de vantagens: a possibilidade de transformar SVG usando técnicas como XSLT, de embeber SVG em qualquer documento XML usando *namespaces* ou até de estilizar SVG recorrendo a CSS (*Cascade Style Sheets*). De uma forma geral, pode dizer-se que SVGs interagem bem com as actuais tecnologias ligadas ao XML e à Web, tal como referido em [?, ?].

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a eros. Phasellus non nulla
 2 non massa venenatis convallis. In porta. Mauris quis magna. Proin mauris eros, aliquet id, eleifend
 vitae, semper quis, erat. Aliquam id lectus non odio dignissim blandit. Vestibulum porttitor arcu ut
 4 ligula. Nunc quis erat. Curabitur ipsum tortor, ornare vitae, dapibus pretium, hendrerit sed, urna.
 Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Phasellus
 6 bibendum, nulla eget varius aliquam, tortor nulla sollicitudin quam, vel vestibulum nisl magna at
 sem. Aliquam velit sapien, ultrices viverra, tempus quis, ultrices at, dui. Aliquam sit amet justo.
 8 Quisque tristique, metus eu iaculis sagittis, urna leo bibendum diam, a ultricies sem diam a
 augue. Mauris consectetur, libero vel euismod tincidunt, nisi metus viverra ante, quis pretium
 10 sapien odio nec risus. Nunc semper auctor nulla¹¹.

2.9.1 Sub-secção Exemplo

12 Batik é um conjunto de bibliotecas baseadas em *Java* que permitem o uso de imagens *SVG* (vi-
 sualização, geração ou manipulação) em aplicações ou *applets* [?]. O projecto Batik destina-se a
 14 fornecer ao programador alguns módulos que permitem desenvolver soluções específicas usando
SVG [?].

16 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc eu nulla. Pellentesque vitae
 nibh ultrices quam iaculis convallis. Aliquam purus eros, varius eget, volutpat sodales, imperdiet
 18 nec, lacus. Curabitur in elit sed sem rutrum posuere. Class aptent taciti sociosqu ad litora torquent
 per conubia nostra, per inceptos himenaeos. Duis sem. Praesent ultricies odio vel sapien. Integer
 20 faucibus malesuada libero. Cras semper, dolor id ullamcorper varius, magna risus volutpat felis,
 id pellentesque nulla ante at erat. Integer sodales.

22 Quisque sit amet odio. In at risus sit amet turpis interdum posuere. Maecenas iaculis vehicula
 sem. Ut leo arcu, malesuada vel, imperdiet id, dignissim a, purus. Duis eleifend, lectus non vene-
 24 natis dignissim, risus libero imperdiet mi, nec gravida massa libero sed mauris. Nullam lobortis
 libero non sapien. Integer convallis iaculis erat. Morbi dictum. Ut ultrices pellentesque velit. Cras
 26 ac ante. Etiam in neque tincidunt lacus gravida vehicula. Proin et nisi.

Vivamus non nunc nec risus tempor varius. Quisque bibendum mi at dolor. Aliquam con-
 28 sectetur condimentum risus. Aliquam luctus pulvinar sem. Duis aliquam, urna et vulputate
 tristique, dui elit aliquet nibh, vel dignissim magna turpis id sapien. Duis commodo sem id quam.
 30 Phasellus dolor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos
 himenaeos.

2.9.2 Sub-secção Exemplo

Loren ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet sem. Maecenas
 34 eleifend facilisis leo. Vestibulum et mi. Aliquam posuere, ante non tristique consectetur, dui
 elit scelerisque augue, eu vehicula nibh nisi ac est. Suspendisse elementum sodales felis. Nullam
 36 laoreet fermentum urna.

¹¹Exemplo de nota de rodapé.

Duis eget diam. In est justo, tristique in, lacinia vel, feugiat eget, quam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Fusce feugiat, elit ac placerat fermentum, augue nisl ultricies eros, id fringilla enim sapien eu felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed dolor mi, porttitor quis, condimentum sed, luctus in. 2 4

2.10 Resumo ou Conclusões 6

No final do capítulo deverá ser apresentado um resumo com as principais conclusões que se podem tirar. 8

Vivamus non nunc nec risus tempor varius. Quisque bibendum mi at dolor. Aliquam consectetur condimentum risus. Aliquam luctus pulvinar sem. Duis aliquam, urna et vulputate tristique, dui elit aliquet nibh, vel dignissim magna turpis id sapien. Duis commodo sem id quam. Phasellus dolor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. 10 12

Chapter 3

2 The problem and the approaches

In this chapter the problem will be described and justified, as well as

4 3.1 Problem Description

The issue and the approach taken to solve it. Este capítulo deve começar por fazer uma apresentação detalhada do problema a resolver¹ podendo mesmo, caso se justifique, constituir-se um capítulo com essa finalidade.

8 Deve depois dedicar-se à apresentação da solução sem detalhes de implementação. Dependendo do trabalho, pode ser uma descrição mais teórica, mais “arquitetural”, etc.

10 3.2 Context - A birdseye view on a bigger project

Currently, FEUP’s computing infrastructures are only accessed by those who have the technical
12 knowledge to interact with the system. These people are technicians whose area of expertise encompasses outsourcing computing resources to perform computing jobs. If someone from an area
14 unrelated to the computing system wants to perform any operation in it, that someone must contact the said technicians and waste valuable time for both parties cutting through red tape. Having this
16 in mind, CICA has started developing a project that reduces the ammount of knowledge necessary to perform the said computing operations. The project aims at simplifying the whole process and
18 to make FEUP’s computing infrastructures more accessible to the academic community, without the users having to spend time learning about the technologies and how the system actually works.
20 This document focuses only on the front-end of the project, the back-end having already been developed by former MIEIC student Nuno Cardoso as part of his Master Thesis. In order to better
22 understand the full scope of the issue, Figure 1 shows the full system as it should function, through the means of an hypotetic and yet plausible use case scenario:

24 «insert figure 1»

¹Na introdução a apresentação do problema foi breve.

Firstly, a researcher of a specific field of study wants to conduct a more complex operation that involves greater computing efforts than his/her home and work computer. As such, the researcher proceeds to enter the designed system through a web page where he/she can:

- Chose a suitable work environment for her computational needs according to a set of predefined parameters;
- Create his/her own work environment according to the specifications he/she provides the system with.

The system will then automatically create a virtual environment (image containing all the information needed), which will be passed onto the second part of the project where a virtual cluster will be created according to that virtual environment. Finally a username and password combination should be returned so that the researcher can enter the created environment and perform his/her operations.

3.3 Secção Exemplo

Neste capítulo apresentam-se exemplos de formatação de figuras e tabelas, equações e referências cruzadas.

Apresenta-se de seguida um exemplo de equação, completamente fora do contexto:

$$CIF_1 : \quad F_0^j(a) = \frac{1}{2\pi i} \oint_{\gamma} \frac{F_0^j(z)}{z-a} dz \quad (3.1)$$

$$CIF_2 : \quad F_1^j(a) = \frac{1}{2\pi i} \oint_{\gamma} \frac{F_0^j(x)}{x-a} dx \quad (3.2)$$

Na Equação 3.2 lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse tincidunt viverra elit. Donec tempus vulputate mauris. Donec arcu. Vestibulum condimentum porta justo. Curabitur ornare tincidunt lacus. Curabitur ac massa vel ante tincidunt placerat. Cras vehicula semper elit. Curabitur gravida, est a elementum suscipit, est eros ullamcorper quam, sed cursus velit velit tempor neque. Duis tempor condimentum ante.

Phasellus imperdiet, orci vel pretium sollicitudin, magna nunc ullamcorper augue, non venenatis dui nunc quis massa. Pellentesque dolor elit, dapibus venenatis, viverra ultricies, accumsan cursus, orci. Aliquam erat volutpat. Mauris ornare tristique leo. Maecenas eros. Curabitur velit nunc, tincidunt vitae, dictum posuere, pulvinar nec, diam. In suscipit mauris a nunc. Pellentesque gravida. Morbi quam lacus, pretium eget, tincidunt vulputate, interdum sed, turpis. Curabitur quis est. Sed lectus lorem, congue vel, dignissim laoreet, blandit a, nisi. Aenean nunc ligula, tincidunt eu, hendrerit vel, suscipit non, erat. Aliquam gravida. Integer non pede. In laoreet augue id leo. Mauris placerat.

A arquitetura do visualizador assenta sobre os seguintes conceitos base [?]:

- **Componentes** — Suspendisse auctor mattis augue *push*;

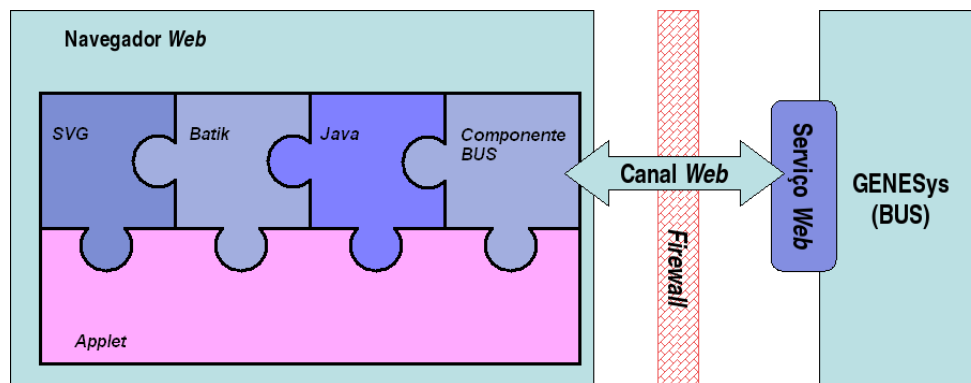


Figure 3.1: Arquitetura da Solução Proposta

- **Praesent** — Sit amet sem maecenas eleifend facilisis leo;
- **Pellentesque** — Habitant morbi tristique senectus et netus.

3.3.1 Exemplo de Figura

É apresentado na Figura 3.1 um exemplo de figura flutuante.

Loren ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet sem. Maecenas eleifend facilisis leo. Vestibulum et mi. Aliquam posuere, ante non tristique consectetur, dui elit scelerisque augue, eu vehicula nibh nisi ac est. Suspendisse elementum sodales felis. Nullam laoreet fermentum urna.

Duis eget diam. In est justo, tristique in, lacinia vel, feugiat eget, quam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Fusce feugiat, elit ac placerat fermentum, augue nisl ultricies eros, id fringilla enim sapien eu felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed dolor mi, porttitor quis, condimentum sed luctus.

3.3.2 Exemplo de Tabela

É apresentado na Tabela 3.1 um exemplo de tabela flutuante e na Tabela 3.2 um exemplo de tabela flutuante, um pouco mais complicada.

Integer quis pede. Fusce nibh. Fusce nec erat vel mi condimentum convallis. Sed at tortor non mauris pretium aliquet. In in lacus in dolor molestie dapibus. Suspendisse potenti. Pellentesque sagittis porta erat. Mauris sodales sapien id augue. Nam eu dolor. Donec sit amet turpis non orci rhoncus commodo. Etiam condimentum commodo libero.

Mauris pede. Curabitur faucibus dictum nibh. Proin tincidunt diam vitae mauris. Sed hendrerit dolor vel ipsum. Nullam dapibus. Vivamus tellus diam, egestas sit amet, vulputate non, vulputate id, eros. Nunc sit amet nibh eget nibh imperdiet ornare. Cras vehicula mattis ipsum. Sed diam arcu, semper at, gravida vitae, fermentum et, nulla. Aenean massa orci, tristique nec, rutrum id,

The problem and the approaches

Table 3.1: Uma Tabela Simples

Acrónimo	Significado
ADT	<i>Abstract Data Type</i>
ANDF	<i>Architecture-Neutral Distribution Format</i>
API	<i>Application Programming Interface</i>

fringilla eget, erat. Curabitur nulla ipsum, aliquam sed, rutrum vitae, semper quis, ante. Fusce at nunc in dolor condimentum tempor. Duis sit amet massa.

2

Curabitur convallis nulla quis risus. Nulla mollis porttitor purus. Fusce ultricies odio at ligula pellentesque suscipit. Nulla velit libero, blandit a, aliquet quis, hendrerit id, arcu. Phasellus porttitor porttitor purus. Suspendisse velit tortor, fringilla sit amet, commodo a, ultrices et, mi. Donec eu metus in erat ornare adipiscing. Praesent varius mi ac nunc. Vestibulum leo lacus, elementum in, vestibulum sit amet, hendrerit at, justo. Sed sit amet neque. Donec libero risus, commodo sit amet, dignissim ut, tincidunt a, eros. Ut non lacus quis tortor mattis ullamcorper. Vivamus consequat augue vel erat. Sed tincidunt. Sed leo eros, ornare a, pulvinar non, mattis quis, nibh. Aliquam faucibus mi ac nisi.

4

6

8

10

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Duis aliquet, libero sit amet ornare viverra, augue erat interdum dolor, vitae tincidunt lorem erat a lacus. Sed lectus nisi, auctor in, hendrerit a, molestie vel, lectus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Duis lacinia tempor dui. Vivamus rhoncus, tellus a viverra dignissim, pede dui adipiscing odio, non faucibus metus mi gravida eros. Nullam a tellus ut velit elementum tempus. Aenean rutrum convallis tellus. Vestibulum nulla ante, dapibus ut, lobortis ut, varius sed, nisl. Fusce lobortis. Sed ac lorem. Nulla tincidunt nulla eget leo. Maecenas ac lectus eu neque ultrices pharetra. Curabitur a risus nec arcu placerat tempor. Suspendisse magna nisl, viverra a, adipiscing eget, ornare ultricies, ligula. Maecenas eu ligula vitae eros convallis dignissim.

12

14

16

18

20

Loren ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet sem. Maecenas eleifend facilisis leo. Vestibulum et mi. Aliquam posuere, ante non tristique consectetur, dui elit scelerisque augue, eu vehicula nibh nisi ac est. Suspendisse elementum sodales felis. Nullam laoreet fermentum urna.

22

24

Duis eget diam. In est justo, tristique in, lacinia vel, feugiat eget, quam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Fusce feugiat, elit ac placerat fermentum, augue nisl ultricies eros, id fringilla enim sapien eu felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed dolor mi, porttitor quis, condimentum sed luctus.

26

28

Table 3.2: Uma Tabela Mais Complicada

k	Iteração k de $f(x_n)$			comentários
	x_1^k	x_2^k	x_3^k	
0	-0.3	0.6	0.7	-
1	0.47102965	0.04883157	-0.53345964	$\delta < \varepsilon$
2	0.49988691	0.00228830	-0.52246185	$\delta < \varepsilon$
3	0.49999976	0.00005380	-0.523656	N
4	0.5	0.00000307	-0.52359743	
\vdots	\vdots	\ddots	\vdots	
7	0.5	0.0	-0.52359878	$\delta < 10^{-8}$

3.4 Secção Exemplo

- 2 Loren ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet sem. Maecenas
 eleifend facilisis leo. Vestibulum et mi. Aliquam posuere, ante non tristique consectetur, dui
 4 elit scelerisque augue, eu vehicula nibh nisi ac est. Suspendisse elementum sodales felis. Nullam
 laoreet fermentum urna.
- 6 Duis eget diam. In est justo, tristique in, lacinia vel, feugiat eget, quam. Pellentesque habitant
 morbi tristique senectus et netus et malesuada fames ac turpis egestas. Fusce feugiat, elit ac
 8 placerat fermentum, augue nisl ultricies eros, id fringilla enim sapien eu felis. Vestibulum ante
 ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed dolor mi, porttitor quis,
 10 condimentum sed luctus.

3.5 Resumo e Conclusões

- 12 Resumir e apresentar as conclusões que se podem tirar no fim deste capítulo.

The problem and the approaches

Chapter 4

2 Implementação

4 Este capítulo pode ser dedicado à apresentação de detalhes de nível mais baixo relacionados com
o enquadramento e implementação das soluções preconizadas no capítulo anterior. Note-se no
6 entanto que detalhes desnecessários à compreensão do trabalho devem ser remetidos para anexos.

Dependendo do volume, a avaliação do trabalho pode ser incluída neste capítulo ou pode
8 constituir um capítulo separado.

4.1 Secção Exemplo

10 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer hendrerit commodo ante. Pel-
lentesque nibh libero, aliquam at, faucibus id, commodo a, velit. Duis eleifend sem eget leo.
12 Morbi in est. Suspendisse magna sem, varius nec, hendrerit non, tincidunt quis, quam. Aenean
congue. Vivamus vel est sit amet sem iaculis posuere. Cras mollis, enim vel gravida aliquam,
14 libero nunc ullamcorper dui, ullamcorper sodales lectus nulla sed urna. Morbi aliquet porta risus.
Proin vestibulum ligula a purus. Maecenas a nulla. Maecenas mattis est vitae neque auctor tempus.
16 Etiam nulla dui, mattis vitae, porttitor sed, aliquet ut, enim. Cras nisl magna, aliquet et, laoreet at,
gravida ac, neque. Sed id est. Nulla dapibus dolor quis ipsum rhoncus cursus.

18 4.2 Mais uma Secção

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque purus sapien, interdum ut,
20 vestibulum a, accumsan ullamcorper, erat. Mauris a magna ut leo porta imperdiet. Donec dui odio,
porta in, pretium non, semper quis, orci. Quisque erat diam, pharetra vel, laoreet ac, hendrerit vel,
22 enim. Donec tristique luctus risus. Fusce dolor est, eleifend id, elementum sit amet, varius vitae,
neque. Morbi at augue. Ut sem ligula, auctor vitae, facilisis id, pharetra non, lectus. Nulla lacus
24 augue, aliquam eget, sollicitudin sed, hendrerit eu, leo. Suspendisse ac tortor. Mauris at odio.
Etiam vehicula. Nam lacinia purus at nibh. Aliquam fringilla lorem ac justo. Ut nec enim.

Implementação

```
1 map(String key, String value):  
2 // key: document name  
3 // value: document contents  
4 for each word w in value:  
5 EmitIntermediate(w, "1");  
6  
7 reduce(String key, Iterator values):  
8 // key: a word  
9 // values: a list of counts  
10 int result = 0;  
11 for each v in values:  
12 result += ParseInt(v);  
13  
14 Emit(AsString(result))
```

Listing 4.1: Example map and reduce functions for word counting

Quisque ullamcorper. Aliquam vel magna. Sed pulvinar dictum ligula. Sed ultrices dolor
ut turpis. Vivamus sagittis orci malesuada arcu venenatis auctor. Proin vehicula pharetra urna. 2
Aliquam egestas nunc quis nisl. Donec ullamcorper. Nulla purus. Ut suscipit lacus vitae dui.
Mauris semper. Ut eget sem. Integer orci. Nam vitae dui eget nisi placerat convallis. 4

Sed id lorem. Proin gravida bibendum lacus. Sed molestie, urna quis euismod laoreet, diam
dolor dictum diam, vitae consectetur leo ipsum id ante. Integer eu lectus non mauris pharetra 6
viverra. In feugiat libero ut massa. Morbi cursus, lorem sollicitudin blandit semper, felis magna
pellentesque lacus, ut rhoncus leo neque at tellus. Sed mattis, diam eget eleifend tincidunt, ligula 8
eros tincidunt diam, vitae auctor turpis est vel nunc. In eu magna. Donec dolor metus, egestas sit
amet, ultrices in, faucibus sed, lectus. Etiam est enim, vehicula pharetra, porta non, viverra vel, 10
nunc. Ut non sem. Etiam nec neque.

4.3 Resumo ou Conclusões

Proin vehicula pharetra urna. Aliquam egestas nunc quis nisl. Donec ullamcorper. Nulla purus.
Ut suscipit lacus vitae dui. Mauris semper. Ut eget sem. Integer orci. Nam vitae dui eget nisi 14
placerat convallis.

Chapter 5

2 Conclusões e Trabalho Futuro

4 Deve ser apresentado um resumo do trabalho realizado e apreciada a satisfação dos objectivos do trabalho, uma lista de contribuições principais do trabalho e as direcções para trabalho futuro.

6 A escrita deste capítulo deve ser orientada para a total compreensão do trabalho, tendo em atenção que, depois de ler o Resumo e a Introdução, a maioria dos leitores passará à leitura deste
8 capítulo de conclusões e recomendações para trabalho futuro.

5.1 Satisfação dos Objectivos

10 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam non felis sed odio rutrum ultrices. Donec tempor dolor. Vivamus justo neque, tempus id, ullamcorper in, pharetra non, tellus.
12 Praesent eu orci eu dolor congue gravida. Sed eu est. Donec pulvinar, lectus et eleifend volutpat, diam sapien sollicitudin arcu, a sagittis libero neque et dolor. Nam ligula. Cras tincidunt lectus
14 quis nunc. Cras tincidunt congue turpis. Nulla pede velit, sagittis a, faucibus vitae, porttitor nec, ante. Nulla ut arcu. Cras eu augue at ipsum feugiat hendrerit. Proin sed justo eu sapien eleifend
16 elementum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vivamus quam lacus, pharetra vel, aliquam vel, volutpat sed, nisl.

18 Nullam erat est, vehicula id, tempor non, scelerisque at, tellus. Pellentesque tincidunt, ante vehicula bibendum adipiscing, lorem augue tempor felis, in dictum massa justo sed metus. Suspendisse
20 placerat, mi eget molestie sodales, tortor ante interdum dui, ac sagittis est pede et lacus. Duis sapien. Nam ornare turpis et magna. Etiam adipiscing adipiscing ipsum. Fusce sodales nisl
22 a arcu. Cras massa leo, vehicula facilisis, commodo a, molestie faucibus, metus. Suspendisse potenti. Duis sagittis. Donec porta. Sed urna. Maecenas eros. Vivamus erat ligula, pharetra sit
24 amet, bibendum et, fermentum sed, dolor. Nullam eleifend condimentum nibh. Integer leo nibh, consequat eget, mollis et, sagittis ac, felis. Duis viverra pede in pede. Phasellus molestie placerat
26 leo. Praesent at tellus a augue congue molestie. Proin sed justo eu sapien eleifend elementum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

5.2 Trabalho Futuro

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam tempor tristique risus. Suspendisse potenti. Fusce id eros. In eu enim. Praesent commodo leo. Nullam augue. Pellentesque tellus. Integer pulvinar purus a dui convallis consectetur. In adipiscing, orci vitae lacinia semper, sapien elit posuere sem, ac euismod ipsum elit tempus urna. Aliquam erat volutpat. Nullam suscipit augue sed felis. Phasellus faucibus accumsan est.

Aliquam felis justo, facilisis sit amet, bibendum ut, tempus ac, dolor. Sed malesuada. Nunc non massa. In erat. Nulla facilisi. Phasellus blandit, est in accumsan cursus, libero augue elementum leo, vitae auctor mauris nisl ac tortor. Cras porttitor ornare elit. Fusce at lorem. Sed lectus tortor, vestibulum id, varius a, condimentum nec, lectus. Maecenas in nisi et magna pretium aliquam. Pellentesque justo elit, feugiat nec, tincidunt a, dignissim vel, ipsum. Sed nunc. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam tempus rhoncus leo. Donec neque quam, cursus sit amet, ultricies varius, semper non, pede. Donec porttitor. Sed aliquet feugiat elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus tellus pede, auctor ut, tincidunt a, consectetur in, felis. Mauris quis dolor et neque accumsan pellentesque. Donec dui magna, scelerisque mattis, sagittis nec, porta quis, nulla. Vivamus quis nisl. Etiam vitae nisl in diam vehicula viverra. Sed sollicitudin scelerisque est. Nunc dapibus. Sed urna. Nulla gravida. Praesent faucibus, risus ac lobortis dignissim, est tortor laoreet mauris, dictum pellentesque nunc orci tincidunt tellus. Nullam pulvinar, leo sed vestibulum euismod, ante ligula elementum pede, sit amet dapibus lacus tortor ac nisl. Morbi libero. Integer sed dolor ac lectus commodo iaculis. Donec ut odio.

Appendix A

2 Loren Ipsum

Depois das conclusões e antes das referências bibliográficas, apresenta-se neste anexo numerado
4 o texto usado para preencher a dissertação.

A.1 O que é o *Loren Ipsum*?

6 *Lorem Ipsum* is simply dummy text of the printing and typesetting industry. Lorem Ipsum has
been the industry’s standard dummy text ever since the 1500s, when an unknown printer took
8 a galley of type and scrambled it to make a type specimen book. It has survived not only five
centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was
10 popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages,
and more recently with desktop publishing software like Aldus PageMaker including versions of
12 Lorem Ipsum [?].

A.2 De onde Vem o Loren?

14 Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of
classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin
16 professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words,
consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical
18 literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and
1.10.33 of “de Finibus Bonorum et Malorum” (The Extremes of Good and Evil) by Cicero, written
20 in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The
first line of Lorem Ipsum, “Lorem ipsum dolor sit amet. . .”, comes from a line in section 1.10.32.

22 The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those inter-
ested. Sections 1.10.32 and 1.10.33 from “de Finibus Bonorum et Malorum” by Cicero are also
24 reproduced in their exact original form, accompanied by English versions from the 1914 transla-
tion by H. Rackham.

A.3 Porque se usa o Loren?

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using “Content here, content here”, making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for “lorem ipsum” will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

A.4 Onde se Podem Encontrar Exemplos?

There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don’t look even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure there isn’t anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic words etc.

References

- 2 [Ama] Amazon. Amazon EC2 FAQ - what is a “EC2 Compute Unit” and why did you
introduce it? [http://aws.amazon.com/ec2/faqs/#What_is_an_EC2_](http://aws.amazon.com/ec2/faqs/#What_is_an_EC2_Compute_Unit_and_why_did_you_introduce_it)
4 [Compute_Unit_and_why_did_you_introduce_it](http://aws.amazon.com/ec2/faqs/#What_is_an_EC2_Compute_Unit_and_why_did_you_introduce_it). Last accessed 16 July
2011.
- 6 [BÓ8] Marc-Elian Bégin. An EGEE comparative study: Grids and clouds - evolution or
revolution. Technical report, CERN - Engineering and Equipment Data Manage-
8 ment Service, June 2008.
- [BAFB05] M. Baker, A. Apon, C. Ferner, and J. Brown. Emerging grid standards. *Computer*,
10 38(4):43 – 50, april 2005.
- [BLDGS04] Miguel Bote-Lorenzo, Yannis Dimitriadis, and Eduardo Gómez-Sánchez. Grid
12 characteristics and uses: A grid definition. In Francisco Fernández Rivera, Mar-
ian Bubak, Andrés Gómez Tato, and Ramón Doallo, editors, *Grid Computing*, vol-
14 ume 2970 of *Lecture Notes in Computer Science*, pages 291–298. Springer Berlin /
Heidelberg, 2004.
- 16 [BV08] Jaijit Bhattacharya and Sushant Vashistha. Utility computing-based framework for
e-governance. In *Proceedings of the 2nd international conference on Theory and*
18 *practice of electronic governance*, ICEGOV ’08, pages 303–309, New York, NY,
USA, 2008. ACM.
- 20 [BYV⁺09] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona
Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality
22 for delivering computing as the 5th utility. *Future Generation Computer Systems*,
25(6):599 – 616, 2009.
- 24 [Car09] Nicholas Carr. *The Big Switch: Rewiring the World, from Edison to Google*. W.W.
Norton & Company, 2009.
- 26 [Car11] Nuno Cardoso. Virtual clusters sustained by cloud computing infrastructures. Mas-
ter’s thesis, FEUP - Faculdade de Engenharia da Universidade do Porto, 2011.
- 28 [CC] Rackspace Cloud Computing. Open Stack - Open Source Cloud Com-
puting Software. [http://stackoverflow.com/questions/209470/](http://stackoverflow.com/questions/209470/can-i-use-python-as-a-bash-replacement)
30 [can-i-use-python-as-a-bash-replacement](http://stackoverflow.com/questions/209470/can-i-use-python-as-a-bash-replacement). Last accessed 16 July
2011.
- 32 [CER] CERN. gLite - Lightweight Middleware for Grid Computing. [http://glite.](http://glite.cern.ch/)
[cern.ch/](http://glite.cern.ch/). Last accessed 16 July 2011.

REFERENCES

- [CGH09] Damien Cerbelaud, Shishir Garg, and Jeremy Huylebroeck. Opening the clouds: qualitative overview of the state-of-the-art open source vm-based cloud management platforms. In *Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware*, Middleware '09, pages 22:1–22:8, New York, NY, USA, 2009. Springer-Verlag New York, Inc. 2 4
- [FFK⁺06] I. Foster, T. Freeman, K. Keahy, D. Scheftner, B. Sotomayer, and X. Zhang. Virtual clusters for grid communities. *Cluster Computing and the Grid, IEEE International Symposium on*, pages 513–520, 2006. 6 8
- [Fos02] Ian Foster. What is the grid? a three point checklist. *Grid Today*, 1(6):22–25, 2002.
- [GKC⁺] Jeremy Geelan, Markus Klems, Reuven Cohen, Jeff Kaplan, Douglas Gourlay, Praising Gaw, Damon Edwards, Brian de Haaff, Ben Kepes, Kirill Sheynkman, Omar Sultan, Kevin Hartig, Jan Pritzker, Trevor Doerksen, Thorsten von Eicken, Paul Wallis, Michael Sheehan, Don Dodge, Aaron Ricadela, Bill Martin, Ben Kepes, and Irving W. Berger. Twenty-One Experts Define Cloud Computing. <http://virtualization.sys-con.com/node/612375>. Last accessed in 16 July 2011. 10 12 14 16
- [GWQF10] Thilina Gunarathne, Tak-Lon Wu, Judy Qiu, and Geoffrey Fox. Cloud computing paradigms for pleasingly parallel biomedical applications. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 460–469, New York, NY, USA, 2010. ACM. 18 20
- [Hay08] Brian Hayes. Cloud computing. *Commun. ACM*, 51:9–11, July 2008.
- [Haz08] Scott Hazelhurst. Scientific computing using virtual high-performance computing: a case study using the amazon elastic computing cloud. In *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, SAICSIT '08, pages 94–103, New York, NY, USA, 2008. ACM. 22 24 26
- [HP10] Joel Hollingsworth and David J. Powell. Teaching web programming using the google cloud. In *Proceedings of the 48th Annual Southeast Regional Conference*, ACM SE '10, pages 76:1–76:5, New York, NY, USA, 2010. ACM. 28
- [IDE⁺06] Alexandru Iosup, Catalin Dumitrescu, Dick Epema, Hui Li, and Lex Wolters. How are real grids used? the analysis of four grid traces and its implications. In *Proceedings of the 7th IEEE/ACM International Conference on Grid Computing*, GRID '06, pages 262–269, Washington, DC, USA, 2006. IEEE Computer Society. 30 32
- [KF98] Carl Kesselman and Ian Foster. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, November 1998. 34
- [KF08] K. Keahey and T. Freeman. Contextualization: Providing one-click virtual clusters. In *eScience, 2008. eScience '08. IEEE Fourth International Conference on*, pages 301–308, dec. 2008. 36 38
- [KG09] Eric Knorr and Galen Gruman. What cloud computing really means. <http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031>, 2009. Last accessed 16 July 2011. 40 42

REFERENCES

- [McF] Paul McFedries. Ieee spectrum - Inside Technology. <http://spectrum.ieee.org/computing/hardware/the-cloud-is-the-computer>. Last accessed 16 July 2011.
- [Mic] Microsoft. Windows Azure | Microsoft Paas | Cloud Services | Application Hosting. <http://www.microsoft.com/windowsazure/>. Last accessed 16 July 2011.
- [NMM07] Hideo Nishimura, Naoya Maruyama, and Satoshi Matsuoka. Virtual clusters on the fly - fast, scalable, and flexible installation. *Cluster Computing and the Grid, IEEE International Symposium on*, 0:549–556, 2007.
- [NS07] Ripal Nathuji and Karsten Schwan. Virtualpower: coordinated power management in virtualized enterprise systems. In *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, SOSP '07, pages 265–278, New York, NY, USA, 2007. ACM.
- [Pin10] Jorge Fernando Maciel Rodrigues Ruão Pinheiro. Interligação de infra-estruturas de computação de elevado desempenho heterogêneas recorrendo a um super escalonador. Master's thesis, FEUP - Faculdade de Engenharia da Universidade do Porto, 2010.
- [Rap04] M. A. Rappa. The utility business model and the future of computing services. *IBM Syst. J.*, 43:32–42, January 2004.
- [Res] Cluster Resources. Cluster Resources: Moab Cluster Software Suite. <http://www.clusterresources.com/products/moab-cluster-suite.php>. Last accessed 16 July 2011.
- [RW04] J. W. Ross and G. Westerman. Preparing for utility computing: The role of it architecture and relationship management. *IBM Syst. J.*, 43:5–19, January 2004.
- [Sto07] Heinz Stockinger. Defining the grid: a snapshot on the current view. *The Journal of Supercomputing*, 42:3–17, 2007.
- [Sun] Karishma Sundaram. Bright Hub - The Hub for Bright Minds. <http://www.brighthouse.com/environment/green-computing/articles/68785.aspx>. Last accessed 16 July 2011.
- [Tea] Condor Team. Condor Project Homepage. <http://www.cs.wisc.edu/condor/>. Last accessed 16 July 2011.
- [VRMCL08] Luis M. Vaquero, Luis Roderio-Merino, Juan Caceres, and Maik Lindner. A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, 39:50–55, December 2008.
- [Wei07] Aaron Weiss. Computing in the clouds. *netWorker*, 11:16–25, December 2007.
- [ZKFF05] Xuehai Zhang, Katarzyna Keahey, Ian Foster, and Timothy Freeman. Virtual cluster workspaces for grid applications. Technical report, 2005.