HAL
open science

Clouds: a New Playground for the XtreemOS Grid Operating System

Christine Morin, Jérôme Gallard, Yvon Jégou, Pierre Riteau

To cite this version:

Christine Morin, Jérôme Gallard, Yvon Jégou, Pierre Riteau. Clouds: a New Playground for the XtreemOS Grid Operating System. [Research Report] RR-6824, INRIA. 2009. inria-00358594

HAL Id: inria-00358594

https://inria.hal.science/inria-00358594v1

Submitted on 3 Feb 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Clouds: a New Playground for the XtreemOS Grid Operating System

Christine Morin — Jérôme Gallard — Yvon Jégou — Pierre Riteau

N° 6824

Février 2009

Thème NUM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6824--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche
RENNES - BRETAGNE ATLANTIQUE

Clouds: a New Playground for the XtreemOS* Grid Operating System

Christine Morin† ‡ , Jérôme Gallard† , Yvon Jégou† , Pierre Riteau†

Thème NUM — Systèmes numériques
Équipe-Projet PARIS

Rapport de recherche n° 6824 — Février 2009 — 16 pages

Abstract: The emerging cloud computing model has recently gained a lot of interest both from commercial companies and from the research community. XtreemOS is a distributed operating system for large-scale wide-area dynamic infrastructures spanning multiple administrative domains. XtreemOS, which is based on the Linux operating system, has been designed as a Grid operating system providing native support for virtual organizations. In this paper, we discuss the positioning of XtreemOS technologies with regard to cloud computing. More specifically, we investigate a scenario where XtreemOS could help users take full advantage of clouds in a global environment including their own resources and cloud resources. We also discuss how the XtreemOS system could be used by cloud service providers to manage their underlying infrastructure. This study shows that the XtreemOS distributed operating system is a highly relevant technology in the new era of cloud computing where future clouds seamlessly span multiple bare hardware providers and where customers extend their IT infrastructure by provisioning resources from different cloud service providers.

Key-words: Cloud Computing, Grid Computing, Virtualization, Distributed System

* The XtreemOS project is partially funded by the European Commission under contract #FP6-033576.

† INRIA, Centre Rennes - Bretagne Atlantique, Campus universitaire de Beaulieu, 35042 Rennes Cedex, France

‡ Corresponding author: Christine.Morin@inria.fr

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex
Téléphone : +33 2 99 84 71 00 — Télécopie : +33 2 99 84 71 71

Les clouds: un nouveau terrain de jeu pour le système d'exploitation de grille XtreemOS

Résumé : L'émergence récente du modèle du *cloud computing* a suscité beaucoup d'intérêt, aussi bien dans les entreprises commerciales que dans le monde académique. XtreemOS est un système d'exploitation distribué pour les infrastructures à grande échelle réparties géographiquement et recouvrant de multiples domaines d'administration. XtreemOS, fondé sur le système d'exploitation Linux, a été conçu comme un système d'exploitation de grille supportant nativement les organisations virtuelles. Dans cet article, nous étudions le positionnement des technologies propres à XtreemOS par rapport au *cloud computing*. Plus spécifiquement, nous étudions un scénario où XtreemOS aiderait les utilisateurs à tirer profit des *clouds* dans un environnement global réunissant leur propres ressources et des ressources émanant de *clouds*. Nous discutons également de la façon dont XtreemOS peut être utilisé par les fournisseurs de service de type *cloud* pour gérer leur propre infrastructure. Cette étude montre que le système d'exploitation XtreemOS est particulièrement pertinent dans la nouvelle ère du *cloud computing* pour gérer des scénarios émergents dans lesquels les *clouds* s'étendent sur de multiples fournisseurs de matériel et les clients étendent leur infrastructure informatique en louant des ressources chez les différents fournisseurs de services de type *cloud*.

Mots-clés : *Cloud Computing*, Grille, Virtualisation, Système Distribué

1 Introduction

The emerging cloud computing model [26, 19, 20] has recently gained a lot of interest both from commercial companies and from the research community, especially from the Grid and distributed systems communities.

Cloud computing considers the outsourcing of hardware and software to Internet providers. In this model, there is no need to bother with hardware acquisition and management: users just rent the services (or virtual machines) they need, and only pay for effective resource usage. Cloud computing is mainly based on virtualization technologies and Grid computing software.

XtreemOS is a four-year integrated project started in June 2006 and partially funded by the European Commission. It aims at developing a Grid operating system (called XtreemOS) based on Linux and providing native support for Virtual Organizations (VOs). Grids are distributed systems enabling the creation of Virtual Organizations (VOs) [12, 14]. By working within VOs, users can share, select, and aggregate a wide variety of geographically distributed resources, owned by different organizations, to solve large-scale computation and data intensive problems in science, engineering, and commerce. Those resources may include any kind of computational resources like supercomputers, storage systems, data sources, sensors, and specialized devices. XtreemOS targets large scale dynamic Grid infrastructures and its goal is to facilitate the use and management of Grid resources.

The contribution of this paper is to position XtreemOS technologies with regard to cloud computing. We identify a number of scenarios that are not yet covered in the existing cloud implementations and some issues in current cloud computing technologies that may be solved using XtreemOS functionalities.

In Section 2, we present a background on cloud computing. Then, in Section 3, we highlight the main features of XtreemOS. The remainder of the paper is devoted to a discussion on how to position XtreemOS in the context of cloud computing. In Section 4, we investigate how XtreemOS could help users take full advantage of clouds in a more global environment including their own resources and cloud resources. In Section 5, we discuss how XtreemOS technologies could be used to manage a cloud computing infrastructure. We conclude and give some work perspectives in the context of the XtreemOS European project in Section 6.

2 Background

Cloud computing offers a whole new paradigm for managing computing resources. Instead of buying and managing real hardware, users rent virtual machines or services from cloud providers. Various cloud software stacks have been developed so far from fully configurable virtual machines to predefined appliances and services. Cloud computing can be layered from the physical architecture (layer 0) to the services provided to the client (layer n):

- *IaaS – Infrastructure as a Service*: companies providing hardware resources (physical or virtual),
- *PaaS – Platform as a Service*: companies providing platforms to deploy and manage applications,

- *SaaS – Service as a Service*: companies providing services (utility computing) above these clouds.

Figure 1 presents a possible cloud computing architecture.

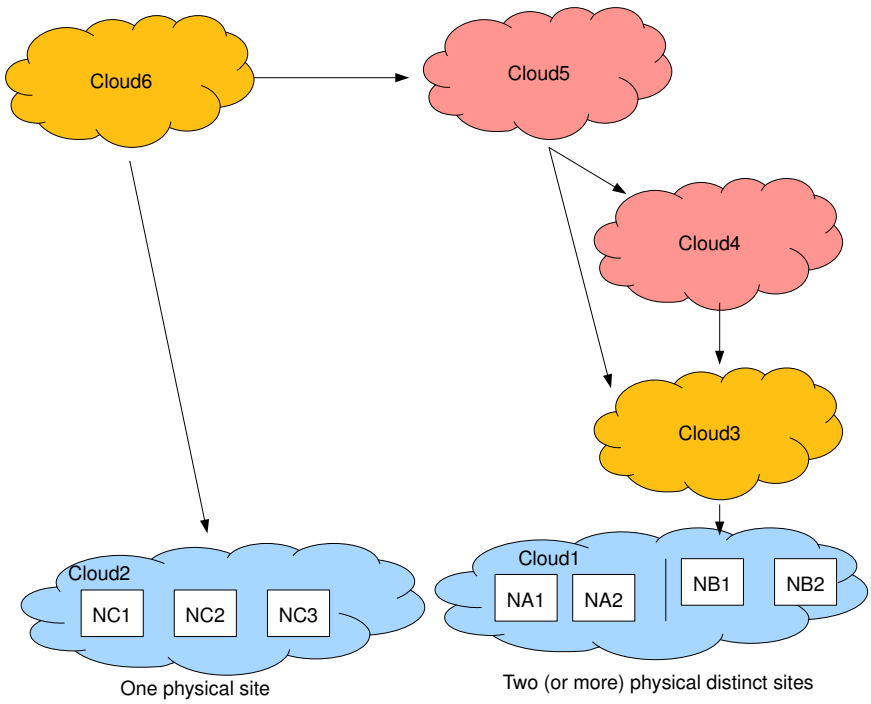

Figure 1: An example of a cloud computing architecture.

This figure describes six clouds. *Cloud1* and *Cloud2* are IaaS (e.g. Amazon EC2 is an IaaS). They provide computing resources to their clients. *Cloud3* is a PaaS: it provides an abstraction of the underlying IaaS to easily deploy and run applications (for example, Google App Engine is a PaaS). *Cloud4* is a SaaS: it provides already set up applications to the user (for instance, Gmail and Animoto are SaaS clouds). *Cloud5* is also a SaaS, providing services built from the platform of *Cloud3* and from services offered by *Cloud4*. *Cloud6* is a PaaS offering a platform built from services of *Cloud5* and from computing resources provided by *Cloud2*.

Amazon Elastic Compute Cloud (EC2) [1] as well as AppNexus [7] and FlexiScale [2] propose fully configurable Linux machines with root access on which clients can install their applications. Data can be stored in Amazon Simple Storage Service (S3), or on a virtual disk using FlexiScale. Virtual machines can communicate and synchronize through Amazon Simple Queue Service (SQS), a message-passing API. GoGrid [3] provides server images with preinstalled software, such as Apache, IIS, MySQL, a load balancer, and a Web-based control panel which allows fast configuration of a distributed server.

Google App Engine [8] lets users run their web applications on Google's infrastructure. These applications run in a sandbox, which strongly restricts possible modifications to the environment: no possibility to write to local files — the App Engine datastore [9] must be used for all data that persists between

requests —, limited access to the Internet, activity allowed only in response to Internet requests within a limited period, etc.

AppLogic [11] appliances run inside virtual machines which can be configured by users. Users build their distributed applications using a graphic interface.

The Hadoop [4] open source software platform lets one easily write and run applications that process vast amounts of data. These applications can be developed using specialized environments such as Pig [10], supported by Yahoo, which generate sequences of Map-Reduce programs.

Google's implementation of MapReduce [16, 5] runs on a large cluster of commodity machines and is highly scalable: a typical MapReduce computation processes many terabytes of data on thousands of machines. Users just need to provide the *map* and *reduce* functions to the system.

These few examples of cloud platforms show a large variety of architectures and a major lack of interoperability between different providers. Moreover, companies already operating their own datacenters cannot easily extend their computation capacity using the cloud.

Using clouds, clients pay only for the resources they effectively consume. Cloud providers allocate physical resources to users on demand. This allows to optimize resource utilization and to propose low pricing. Cloud computing provides companies with a way to increase capacity or add capabilities on the fly without investing in a new infrastructure, hiring skilled personnel, or licensing new services. Ideally, users should not even see any disruption in their applications when the hardware providers upgrade or change their platforms.

Resource management software used in clouds to allocate and deploy virtual machines should have the following properties:

- efficient scheduling of VMs on physical nodes,
- resource usage accounting,
- fault tolerance (e.g. using VM checkpoint/restart),
- efficient storage of VM images.

In order to provide powerful services to users, cloud systems face many challenges:
Scalability: supporting large numbers of nodes in order to support large numbers of concurrent applications,
Agility: providing fast response to user requests,
Transparency: hiding the complexity and the heterogeneity of cloud services,
Interoperability: allowing multi-cloud applications to be deployed and providing means to dynamically extend private clouds,
Dependability: providing reliability & high availability,
Security: ensuring trust and integrity according to service level agreements.

3 Overview of XtreemOS Features

In this section, we provide an overview of XtreemOS services [13], focusing on those which are of interest for using the XtreemOS system in the framework of cloud computing scenarios.

3.1 XtreemOS Fundamental Properties

While much work has been done to build Grid middlewares on top of existing operating systems, the XtreemOS Grid operating system results from a novel approach where the underlying operating system is extended for enabling and facilitating Grid computing [22, 15]. The XtreemOS operating system is based on the Linux traditional general-purpose OS, extended as needed to support VOs, and to provide appropriate interfaces to Grid OS services. In contrast to middleware approaches, XtreemOS is an operating system able to execute any kind of application, including unmodified existing applications. XtreemOS aims at providing transparency for users and application developers, scalability, manageability, security, trust. XtreemOS targets dynamic Grids with high node churn. Grid nodes may join or leave the system at anytime based on decisions of their administrator or user. This may happen with a prior announcement, via a sign on/off, or without, for example in the event of a crash. A given Grid node may be temporarily disconnected as network failures may occur at any time.

3.2 XtreemOS Services

Figure 2 depicts the XtreemOS software architecture at a high level of abstraction. Internally, XtreemOS is composed of two parts: the XtreemOS foundation called XtreemOS-F and high-level services called XtreemOS-G. XtreemOS-F is a modified Linux kernel embedding VO support mechanisms and providing kernel level process checkpoint/restart functionalities. XtreemOS-G is implemented on top of XtreemOS-F at user level. XtreemOS-G is comprised of several Grid OS distributed services to deal with resource and application management in VOs.

XtreemOS targets scalable and flexible management of dynamic VOs [14]. In XtreemOS, a VO-frame spans multiple administrative domains on different sites and is comprised of the set of resources (computers, VO-frame members (human beings)) shared by the participating organizations. A VO-frame member can create a VO, for which he becomes the owner. Any VO-frame member may request his registration in a given VO, subject to the VO owner approval. Resources can be registered as well in VOs. The VO owner defines policies, stating permission and usage rules for VO resources. The VO-frame administrator also defines policies regulating what a VO-frame member can do (for example, permission to create a VO). Resource owners in the different administrative domains may also define local policies for resource usage. VO-frame, VO and local policies are enforced by the XtreemOS system. A VO-frame member or a resource may belong to multiple VOs and to multiple VO-frames. XtreemOS implements a set of security and VO management services to manage authentication, authorization, single-sign-on and accounting. XtreemOS relies on node level system mechanisms in order to dynamically map global identities of VO members into local Linux accounts.

XtreemFS is the XtreemOS Grid file system [21]. It allows the federation of multiple data stores located in different administrative domains and provides secure access to stored files to VO members, whatever their location. It provides a POSIX interface and consistent data sharing. Moreover, XtreemFS allows automatic data replication management for data access efficiency and availability.

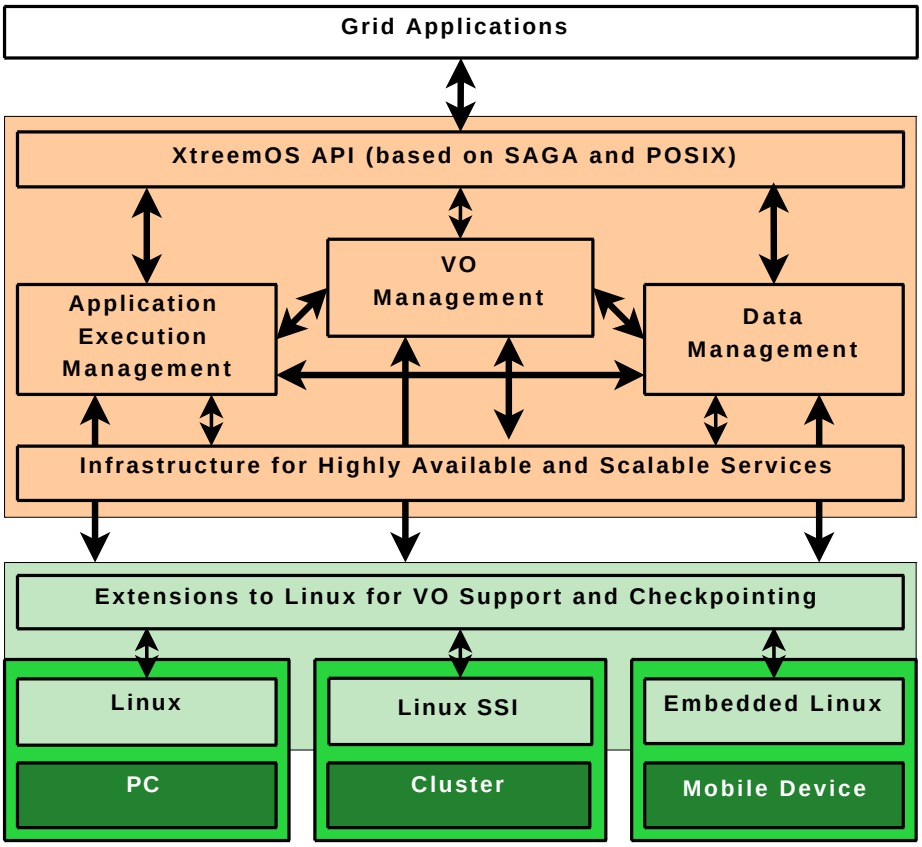

Figure 2: XtreemOS software architecture
The lower layers of the system build the XtreemOS foundation (XtreemOS-F). Grid OS distributed services constitute the upper layers (XtreemOS-G).

Files are stored in XtreemOS volumes. Each VO user gets an XtreemFS home volume, which is automatically mounted by the XtreemOS system as needed.

The application execution management service (AEM) of XtreemOS is in charge on the one hand of discovering, selecting and allocating resources for job execution and on the other hand of starting, controlling and monitoring jobs. In XtreemOS, jobs are self-scheduled, meaning that there is no assumption that a batch scheduler exists on Grid nodes [18]. When a user launches a job, the AEM service dynamically discovers resources based on an overlay linking all VO resources. Resources dynamically join and leave the overlay.

The infrastructure for highly available services provides a set of building blocks (such as overlays, publish/subscribe, replication and distributed directory services) that allow the upper services (AEM, XtreemFS, VO and security management services) to cope with the large scale and dynamicity of grids.

There are three flavors of the XtreemOS-F system, one for each kind of Grid node supported by XtreemOS: PC, cluster and mobile device. LinuxSSI, the cluster flavor of XtreemOS-F, leverages the Kerrighed [23, 24] Linux-based

single system image (SSI). A cluster appears as a single powerful node in the Grid, its individual nodes being invisible at Grid level.

To execute an application in XtreemOS, a VO user logs in a VO he belongs to and gets XtreemOS credentials from the Credential Distribution Authority service. These credentials contain his public key and VO attributes and are used by other XtreemOS services (AEM, XtreemFS, etc.) to authenticate the user and check his authorizations. The AEM service launches a resource discovery over the VO-frame overlay. Suitable resources registered in the user VO reply to this discovery. Then AEM selects and allocates resources for the application execution, that it monitors until termination. Access to files stored in XtreemOS volumes is granted based on VO user credentials.

4 XtreemOS in Virtual Machines for Supporting Cooperation over Clouds

4.1 Scenario Description

Current operating systems used in clouds (e.g. standard Linux distributions) offer little out of the box support for collaboration between different users or organizations. Contrarily, XtreemOS has been designed with such capabilities.

In this section, we study a scenario where XtreemOS is used as a foundation for supporting cooperation between users or organizations over clouds. We also study an extension of this scenario where XtreemOS is used in a heterogeneous environment: some resources are virtual machines from clouds while other resources are physical machines owned by organizations participating in the XtreemOS Grid (private clouds).

4.2 Advantages of Using XtreemOS in this Context

Being designed for the Grid, the services offered by XtreemOS are a sound basis to support cooperation between users or organizations over clouds.

First, organizations using cloud resources should be allowed to easily share data, even when simultaneously using several separate clouds. By deploying XtreemOS in virtual machines from clouds, organizations are able to take advantage of the services offered by XtreemOS. For instance, XtreemFS, the Grid file system part of XtreemOS, allows to share data inside and across clouds boundaries. The virtual organizations (VO) management of XtreemOS allows to describe the level of sharing between the different organizations. Since XtreemFS can be accessed using the POSIX API, it integrates well with legacy applications, in contrast to data storage services using specific interfaces like Amazon S3. XtreemFS also offers fault tolerance, using data replication.

Second, the AEM (Application Execution Manager) service not only schedules jobs but also monitors them to make sure they are executed seamlessly. Jobs can be periodically checkpointed to be able to restart them in case of failure. This can be valuable when using clouds where virtual machines availability is not guaranteed. It can also offer an alternative to virtual machines snapshots with a lower overhead (thus at a lower cost).

Finally, the cluster flavor of XtreemOS based on SSI technology (LinuxSSI) allows users to build virtual machines with a large number of CPUs to run SMP

applications. For instance, Amazon EC2 offers at most 8-cores virtual machines. LinuxSSI allows running SMP applications in virtual machines having more than 8 CPUs.

The preceding services can also be used in a scenario where organizations link their own computing resources with resources from cloud providers (see Figure 3). This allows organizations to increase their computing capabilities when the workload gets higher while retaining seamless integration of the whole infrastructure.

4.3 Limitation of the Current Version of XtreemOS, Research Avenues

The current version of XtreemOS shows a few limitations in this context. First, while XtreemOS was designed for dynamic addition/removal of computing resources, it is not capable of dimensioning itself a pool of resources obtained from the cloud in response to the current job workload. A new service in XtreemOS would add this functionality. It would be written in a generic way, in order to be interconnected with existing cloud providers (each cloud infrastructure having its own different interface).

In the context of a Grid composed of cloud and non-cloud resources, XtreemOS should be able to balance usage between real hardware and virtual machines, depending on the priorities of jobs. For example, a high priority job which can be easily parallelized can take advantage of the elasticity of cloud architectures: the cost of computing a parallel job in the cloud is mostly the same whether using only one machine or using hundreds of machines. Less important jobs or jobs that cannot be parallelized can run on real hardware where the flexibility offered by clouds is not interesting and results in higher computing costs.

XtreemOS would need to consider the economic constraints of cloud architectures. For instance, when renting virtual machines on Amazon EC2, users are charged not only for hours of computation but also for traffic entering or leaving the cloud. In order to minimize the cost of running parallel applications on clouds, XtreemOS should be aware of clouds topologies. For example, the AEM service should not deploy heavily communicating processes on different clouds. Similarly, XtreemFS should place data replicas in order to minimize the amount of traffic between clouds. In conclusion, XtreemOS should have mechanisms taking into account the cost of inter-cloud and intra-cloud communications.

From a technical standpoint, XtreemOS should run on most virtualization technologies. Type-I and type-II virtualization technologies should be handled without modification because they provide an interface similar to real hardware.

5 XtreemOS as an Infrastructure System for Cloud Providers

5.1 Scenario Description

We define a scenario with three levels of entities. The first one is the provider of physical hardware (Bare Hardware Provider – BHP). The second one is the provider of cloud services (Cloud Service Provider – CSP). The third one is the

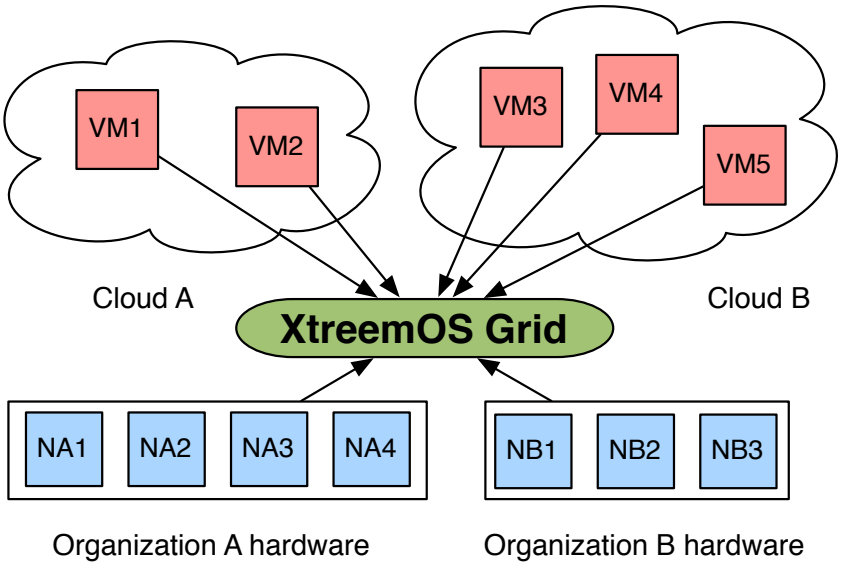

Figure 3: Organizations A and B participate in an XtreemOS Grid and extend their resources with cloud virtual machines.

end user, or an intermediate customer able to offer services to others (like a PaaS or a SaaS). Figure 4 describes a simple case of cloud study.

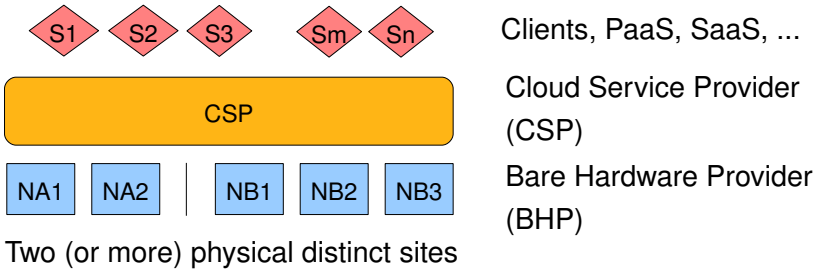

Figure 4: Cloud, simple example: From bare hardware to customers.

In this scenario, XtreemOS is set up at the CSP layer: it is used as an infrastructure for cloud providers. In this case, administrators of the CSP can take advantage of all services offered by XtreemOS. From the administrators point of view, XtreemOS can logically aggregate resources provided by different BHPs. In addition, with the VO layer of XtreemOS, the CSP could easily manage its clients by putting them in specific VOs. Indeed, VOs allow the definition of policies and rules controlling access rights to resources by users. Moreover,

using VOs naturally implies isolation between resources and/or applications running on top of these resources. Note that, in the following paragraphs, we consider that a VM is managed by XtreemOS as an application: then VM can run any operating system (an architecture based on XtreemOS installed in VMs is addressed in Section 4).

Figure 5 presents this case of study. A CSP decides to setup XtreemOS on all the physical resources provided by several BHPs. Then the CSP creates several VOs with different policies and associates its clients to each VO.

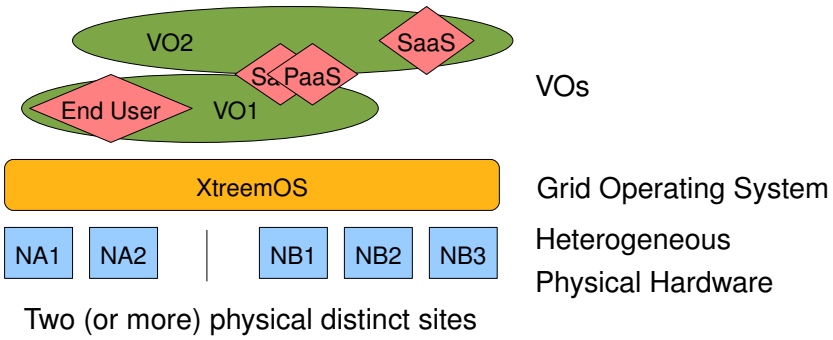

Figure 5: XtreemOS on top of bare hardware.

Additionally, XtreemOS services provide application migration and real time addition/removal of physical resources. Leveraging both kinds of services, XtreemOS is able to perform server consolidation by dynamically allocating more or less resources according to the current workload. Naturally, using VMs instead of standard applications extends server consolidation to a higher level.

In addition, in the case of application workflows (for instance, an application geographically distributed on several sites), XtreemOS proposes services allowing to deploy and manage these workflows. For instance, the XtreemOS publish/subscribe service can be used for service coordination.

Nowadays, solutions allowing to create a cloud exclusively use virtualization technologies due to their flexibility. However, in some cases, introducing a virtualization layer is not necessary. In these cases the overhead added by virtualization can be avoided. This is especially true in the area of green computing. In essence, XtreemOS allows users to define the granularity they need. According to workload and resources availability, XtreemOS will provide users with a complete VM (using full virtualization or not) or just a container associated with VO policies.

Currently, clouds provide a directory of resources available for renting. Generally these resources are elementary virtual machines with x GHz CPUs and y GB of memory. Contrarily, XtreemOS is able to propose to users a large collection of resources including physical nodes, virtual nodes, and large SMP machines with its LinuxSSI flavor.

5.2 Advantage of Using XtreemOS in this Context

Assuming XtreemOS would be able to manage VMs as applications, a physical node could be used by XtreemOS as a simple XtreemOS node (basic case), as a node able to run VMs, or both. Thanks to this property, a cloud could be managed by XtreemOS in a very easy and flexible way.

As said earlier, XtreemOS provides VO support. This support allows a CSP to manage its customers easily. For instance, it is possible for a CSP to create one VO per customer (the customer would be responsible for his VO).

In addition, a CSP can create several kinds of VOs (a specific VO for clients of type A, and another for clients of type B). For instance, it is possible for a CSP to define a VO providing only VMs, as in current clouds. In this case, clients of the CSP deal only with VMs and never deal with the physical hardware. A VM can be migrated from one node to another node of the same VO. This way, all services provided by XtreemOS are available for the CSP. A CSP would be able to make server consolidation by renting more or less physical machines in real time (for instance, to migrate all applications and VMs running on one BHP to another BHP).

Moreover, XtreemOS provides job management services (AEM). This job management can be used in two modes: interactive or not. In the case of interactive jobs, all submitted jobs are executed immediately. But, in the case of non-interactive mode, XtreemOS is able to schedule jobs over time on the pool of physical resources corresponding to a certain VO. In addition, the job management services of XtreemOS are able to guide the deployment of applications in order to optimize their execution (for instance, to minimize the network traffic and maximize the network bandwidth).

Using VMs, users are able to run legacy applications. For example, a Windows application could be executed on a VM managed by XtreemOS. Similarly, it is possible for a user to run applications natively on bare hardware (without virtualization) and take advantage of all services provided by XtreemOS. Both cases could be of high interest for a CSP.

Concerning SMP scalability, lots of current clouds provide SMP machines with a maximum of 8 CPUs. Thanks to the LinuxSSI flavor of XtreemOS, it is possible to instantiate large SMP machines (up to 256 CPUs), allowing the creation of very large clusters with the SSI flavor.

5.3 Limitation of the Current Version of XtreemOS, Research Avenues

In the current version of XtreemOS, support for managing VMs is not yet possible. We have started to study the behavior of XtreemOS with type-I and type-II VMs.

In addition, we would like to see how XtreemFS could be used as a storage backend by operating systems running inside VMs. Using a network file system such as Samba could help seamlessly exporting XtreemFS to guest operating systems.

The job deployment capabilities of XtreemOS need to be modified to deal with VMs. XtreemOS has to manage VMs instead of simple processes and has to optimize resource usage.

The current version of XtreemOS can add/remove physical resources. However, it is not able to add computing resources obtained from another cloud. Managing the addition/removal of virtual resources has to be investigated.

6 Conclusion

This study shows that the XtreemOS distributed operating system is highly relevant in the new cloud computing era. In the future of the Internet, customers will provision their resources and get services from multiple clouds exhibiting heterogeneous interfaces. We claim that it is more realistic to study the interoperability between clouds at the resource management level than at the customer API level. XtreemOS technologies can be used for federating clouds. They also enable single clouds spanning multiple administrative domains and built on top of resources provisioned from different bare hardware providers.

XtreemFS, XtreemOS security and VO management services and the single system image technology used in the XtreemOS cluster flavor are of particular interest in cloud computing scenarios. XtreemFS, exhibiting a POSIX interface and federating storage from different administrative domains in a wide-area system, provides an interesting alternative to existing cloud storage services. XtreemOS security and VO management services offer a very convenient tool to manage customers classes and resource usage policies. Last but not least, the single system image technology allows to extend the range of virtual architectures offered to customers to virtual SMP machines providing a larger amount of cores than the traditional virtual multi-core PC, typical in current clouds such as Amazon EC2.

The first basic version of the XtreemOS Grid operating system needs to be improved to fit with all the above mentioned benefits it can provide in the framework of cloud computing. We plan to experiment the various cloud computing scenarios depicted in Figure 6 first with the XtreemOS Grid system and the Nimbus cloud middleware [6] open source software, and then with additional cloud middleware, for example Eucalyptus [25] and Amazon EC2. The Science clouds [6] and XtreemOS permanent testbed provide us with a sound foundation for these experiments.

We also plan to study scenarios combining virtual and physical resources. The ALADDIN-G5K French national platform for experimenting software for large-scale distributed systems provides us with an interesting use case in this area. Researchers in computer science reserve a subset of ALADDIN-G5K physical resources to deploy their software directly on bare hardware. In order to avoid wasting resources, best effort jobs are launched on idle resources but are stopped without notice when a reservation needs the resources they take advantage of. ALADDIN-G5K can be seen as a cloud service where the customers do not get virtual machines but physical machines. It is desirable in the context of ALADDIN-G5K to automatically manage a combination of physical and virtual machines in order to execute best efforts jobs in virtual machines on idle resources while continuing to give the highest priority to experiments requiring bare hardware [17].

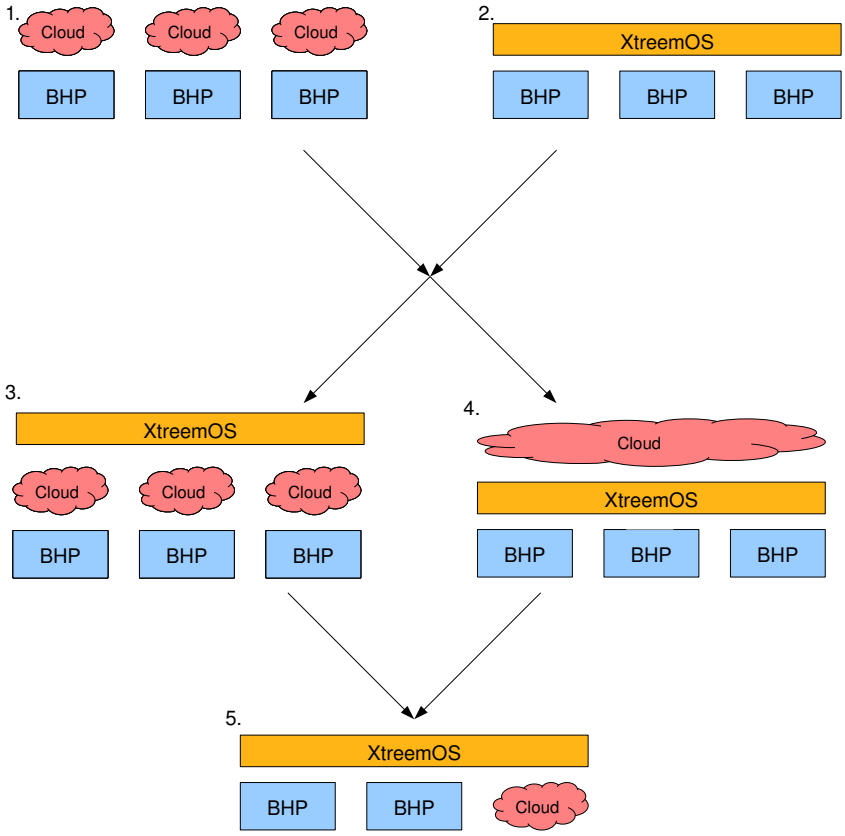

Figure 6: Principal possibilities of combinations of clouds and XtreemOS

Case 1 and *Case 2* are basic scenarios. *Case 1* presents a cloud (for instance, Nimbus) installed on BHP resources and *case 2* presents XtreemOS installed on BHP resources. The combination of *Case 1* and *Case 2* provides two other cases. The first one, *Case 3*, presents XtreemOS on top of several clouds. In this case, XtreemOS allows federating several different clouds. The second one, *Case 4*, presents a cloud setup on XtreemOS. This case provides the advantage of allowing a cloud distributed on several administration domains. The last case, *Case 5*, is produced by the combination of *Case 3* and *Case 4*. In this case, XtreemOS is able to federate physical resources provided by BHPs and virtual resources provided by clouds. An obvious extension, not described here, is the case where XtreemOS is setup on several clouds to provide one unified cloud.

References

[1] Amazon Elastic Compute Cloud (Amazon EC2). `http://aws.amazon.com/ec2/`.

[2] FlexiScale - Utility Computing On-Demand. `http://www.flexiscale.com/`.

[3] GoGrid Cloud Hosting: Instant Windows and Linux Cloud Servers. `http://www.gogrid.com/`.

[4] Hadoop home page. `http://hadoop.apache.org/core/`.

[5] MapReduce: Simplified Data Processing on Large Clusters. `http://labs.google.com/papers/mapreduce.html`.

[6] Nimbus and Science clouds. `http://workspace.globus.org/`.

[7] Scalable, Enterprise, Self-Service Hosting. `http://www.appnexus.com/`.

[8] Software-as-a-service for business email, information sharing and security. `http://www.google.com/apps/intl/en/business/index.html`.

[9] The Python Datastore API. `http://code.google.com/appengine/docs/python/datastore/`.

[10] Yahoo Pig. `http://research.yahoo.com/node/90`.

[11] 3tera. AppLogic - Grid Operating System for Web Applications. `http://www.3tera.com/AppLogic/`.

[12] Marcim Adamski, Gracjan Jankowski, Norbert Meyer, Alvaro Arenas, Brian Matthews, Michael Wilson, Angelos Bilas, Paraskevi Fragopoulou, Vasil Georgiev, Alejandro Hevia, and Jorg Platte. Trust and Security in Grids: A State of the Art, May 2008.

[13] XtreemOS consortium. Revised system architecture, November 2008.

[14] Massimo Coppola, Yvon Jégou, Brian Matthews, Christine Morin, Luis Pablo Prieto, Oscar David Sánchez, Erica Yang, and Haiyan Yu. Virtual Organization Support within a Grid-wide Operating System. *IEEE Internet Computing*, 12(2):20–28, March 2008.

[15] Toni Cortes, Carsten Franke, Yvon Jégou, Thilo Kielmann, Domenico Laforenza, Brian Matthews, Christine Morin, Luis Pablo Prieto, and Alexander Reinefeld. XtreemOS: a Vision for a Grid Operating System. Technical report 4, XtreemOS European Integrated Projet, May 2008.

[16] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI'04: Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, pages 137–150, 2004.

[17] Jérôme Gallard, Oana Goga, Adrien Lèbre, and Christine Morin. VMdeploy, Improving Best-Effort Job Management in Grid'5000. Technical Report RR-6764, INRIA, December 2008.

[18] F. Guim, I. Rodero, M. Garcia, and J. Corbalan. The XtreemOS JScheduler: Using Self-Scheduling Techniques in Large Computing Architectures. In *LASCO'08: First USENIX Workshop on Large-Scale Computing*, pages 1–10, 2008.

[19] Eric Hand. Head in the clouds. *Nature*, 449:963, October 2007.

[20] Brian Hayes. Cloud Computing. *Communications of the ACM*, 51(7):9–11, 2008.

[21] Felix Hupfeld, Toni Cortes, Björn Kolbeck, Jan Stender, Erich Focht, Matthias Hess, Jesus Malo, Jonathan Marti, and Eugenio Cesario. The XtreemFS architecture—a case for object-based file systems in Grids. *Concurrency and Computation: Practice & Experience*, 20(17):2049–2060, 2008.

[22] Christine Morin. XtreemOS: a Grid Operating System Making your Computer Ready for Participating in Virtual Organizations. In *ISORC '07: Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, pages 393–402, May 2007.

[23] Christine Morin, Pascal Gallard, Renaud Lottiaux, and Geoffroy Vallée. Towards an Efficient Single System Image Cluster Operating System. *Future Generation Computer Systems*, 20(4):505–521, May 2004.

[24] Christine Morin, Renaud Lottiaux, Geoffroy Vallée, Pascal Gallard, David Margery, Jean-Yves Berthou, and Isaac Scherson. Kerrighed and Data Parallelism: Cluster Computing on Single System Image Operating Systems. In *Proceedings of the 2004 IEEE International Conference on Cluster Computing*, pages 277–286, September 2004.

[25] Daniel Nurmi, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov. The Eucalyptus Open-source Cloud-computing System. In *Proceedings of Cloud Computing and Its Applications*, October 2008.

[26] Aaron Weiss. Computing in the Clouds. *netWorker*, 11(4):16–25, 2007.

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399