Cloud Campus Services in PLATON e-Science Platform

Roman Wyrzykowski[1], Marek Zawadzki[2], Tomasz Chmiel[1], Piotr Dzierzak[1], Artur Kaszuba[2], Jacek Kochan[2], Jerzy Mikolajczak[2], Tomasz Olas[1], Sebastian Petruczynik[2], Krzysztof Rojek[1], and Lukasz Szustak[1]

[1] Dept. of Comp. & Information Science, Czestochowa Univ. of Technology, Poland
[roman, chmiel, dzierzak, olas, krojek, lszustak]@icis.pcz.pl
[2] Poznan Supercomputing and Networking Center, Poland
[mzawadzk, akaszuba, jacolex, jerzym, seba]@man.poznan.pl

Abstract. In this paper, we present the Campus Services platform, the Polish computing cloud for academic and research community. This platform is built within the PLATON project, which aims at creating a nationwide service platform based on the wide-area network infrastructure provided by the Polish Optical Network PIONIER. The PLATON Campus Services will be carried out by 20 MANs (Metropolitan Area Networks) and HPC centers, equipped with local clusters connected by the PIONIER network. Campus Services are built on the basis of an innovative computing and service infrastructure operating nationwide. It will deliver applications on demand, and will be able to provide users with flexible and scalable access to specific applications, both under MS Windows and Linux, for a wide range of users from academia and research environments, taking into account the needs of specific workgroups in these environments.

1 Introduction

Over the past few years, cloud computing has rapidly emerged [4] as a widely accepted computing paradigm. It refers [1] to both the applications delivered as services over the Internet and the hardware and system software in the data centers that provide those services. Numerous cloud platforms have been developed over such a short period of time, both in industry and academia [4, 3].

In this paper, we present the Campus Services platform, the Polish computing cloud for academic and research community. This platform is built within the PLATON project [2], which aims at creating a nationwide service platform based on the wide-area network infrastructure provided by the Polish Optical Network PIONIER

The PLATON Campus Services will be carried out by 20 MANs (Metropolitan Area Networks) and HPC centers (collectively "MANs"), equipped with local clusters connected by the PIONIER network. Campus Services are built on the basis of an innovative computing and service infrastructure operating nationwide. It will deliver applications on demand, and will be able to provide users with flexible and scalable access to specific applications, both under MS Windows and Linux. Users will access Campus Services in a unified fashion, regardless of his/her actual physical location. This will be enabled by an upper management layer, responsible for directing users' tasks to appropriate local clusters and, subsequently, cluster nodes. The direct support offered by service providers will be also essential. The partners will open technical support desks for users in 20 MANs.

The materials of this paper is organized as follows. Section 2 introduces the PLATON service platform. The main objectives of Campus Services, as well as the basic cluster infrastructure, are presented in Section 3, while the architecture of Campus Services is overviewed in Section 4, including characterization of software layers. Section 5 presents the main components of the front-end (upper) layer and its role in reservation of applications, except for the resource scheduler which is outlined in a separate Section 6.

2 PLATON - Service Platform for e-Science

The PLATON project [2] is realized by the consortium of 22 MANs – members of the PIONIER networking consortium. The project is based on the nation-wide network infrastructure provided by the PIONIER network, and is financed by the European Regional Development Fund (81.9%), Polish Ministry of Science and High Education (14.5%), and from own resources of the project consortium members. The total cost of the project exceeds 20 million euros. The whole project is entirely non-commercial, and is addressed to Polish scientific and academic communities. Its duration is from July 1st, 2009 until July 25th, 2012.

The main objective of the PLATON project is development of the national ICT infrastructure for science, providing applications and services to support scientific research and development of Polish research teams for the innovative economy. The direct goal of the project is to implement modern ICT services available to the scientific community in Poland, namely:

1. Videoconference Services: realized by building a high-quality, secure videoconference system in the PIONIER network;
2. Eduroam Services: simple and secure roaming, providing ubiquitous access to the wireless network in every MAN and HPC center;
3. Campus Services;
4. Archiving Services: available at the national level, offer remote archiving and backup as a value added to the PIONIER network;
5. Science TV Services: the national platform offering interactive HD television for both education, and popularization of science and telemedicine.

3 PLATON Campus Services: Objectives and Basic Infrastructure

The PLATON Campus Services are built on the basis of a computing-service infrastructure operating nationwide (Fig. 1). It will deliver applications on-demand, and will be able to provide flexible and scalable access to specific applications, both in MS Windows and Linux systems, for a wide range of users from academia and research environments, taking into account the needs of specific workgroups in these environments. The integration of services already existing in the PIONIER network, MANs and archiving services will take place. The implementation of the so-called. "last mile" on the basis of metropolitan and regional networks will ensure the appropriate quality of access.

Fig. 1. Distributed infrastructure of PLATON Campus Services based on PIONIER network

In particular, Campus Services will offer the following functionality:

- remote work with interactive (graphic) applications in MS Windows environment, e.g. Matlab/Simulink, AutoCad graphic tools, Corel, Adobe, Ansys, Mathematica, Maple, Comsol, Qsite/Jaguar, etc.;
- on-demand execution of virtual machines (with MS Windows or Linux system) which would create a dedicated working environment for applications of a given user, e.g. researcher, programmer or graphic designer;
- possibility of creating a virtual mini-cluster for the needs of a particular group of users, e.g. a laboratory for a group of students or researchers who use specific applications.

The infrastructure of Campus Services consists of 20 local clusters located in MANs involved in the project. There are eight larger clusters, called "XL", and twelve smaller "L" installations. Totally they consist of 744 two-processor nodes. Among them, there are computing nodes (without graphics accelerators) and graphics ones (with accelerators). Each "XL" cluster contains 32 computing nodes (IBM HS22 blade servers), and 16 graphics nodes (IBM x3550M3 rack servers) equipped with the NVIDIA Quadro FX 580 graphics accelerator, while each "L" installation consists of 20 computing nodes and 10 graphics ones. Every node contains two Intel Nehalem-based Xeon processors, which are either E5530 2.40 GHz or E5520 2.26GHz, as well as 24 – 48 GB of RAM. Besides two 146GB SAS disks in each node, IBM DS3200 disk arrays provide a shared disk storage in each cluster of either 48TB or 24TB in case of "XL" or "L" cluster, respectively. Nodes of a cluster and its shared disk storage are interconnected using 10-Gigabit Ethernet.

4 Architecture of Campus Services: Software Layers

The architecture of software in Campus Services are divided into following layers: (i) front-end (upper) layer), (ii) back-end middleware, (iii) core system services (Fig.2). These layers are developed based on Microsoft Corp. and open-source solutions.

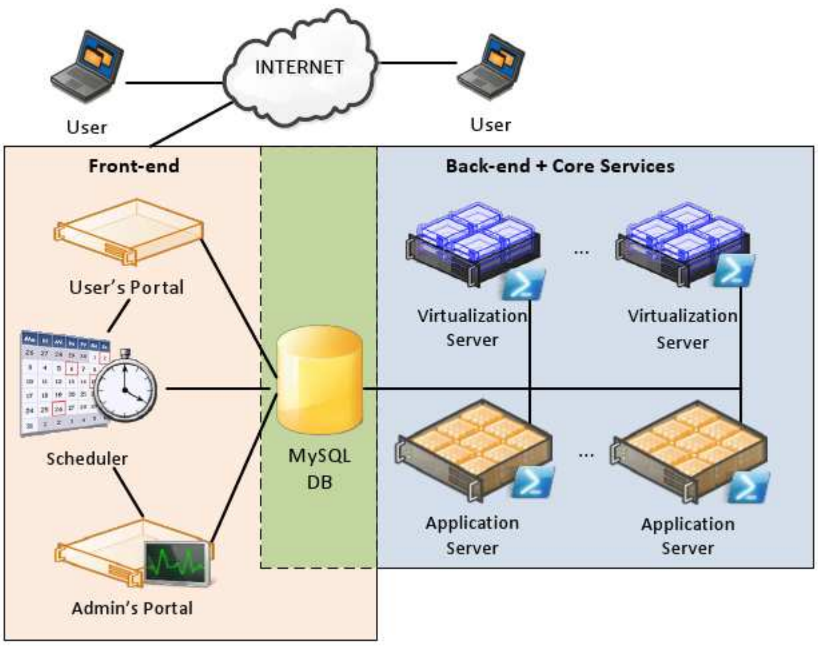

Fig. 2. Logical architecture of local cluster

Users access Campus Services through the back-end layer responsible for directing user's tasks to an appropriate local cluster and, subsequently, cluster nodes. This layer includes the portal (user's and administrator's parts), resource scheduler, and database. The back-end is middleware (written mostly in MS PowerShell) that communicates both with the upper layer and lower layers: core system services (Active Directory services, Remote Desktop Services, Hyper-V virtualization) and hardware (cluster nodes, storage). The functions of the middleware are as follows:

- initializing and managing virtual machines required by reservations stored in the database;
- reconfiguring on-the-fly Remote Desktop Services (RDS) to enable users to access specific nodes and applications accordingly to reservations;
- managing power, networking, and multibooting cluster nodes;
- synchronizing Active Directory (AD) services with portal databases.

Core system services of the cluster are based on Microsoft Windows Server 2008 R2, and Debian Linux. The latter is used as an OS for the cluster router/gateway, running the portal, and boot/power manager (Fig. 3).

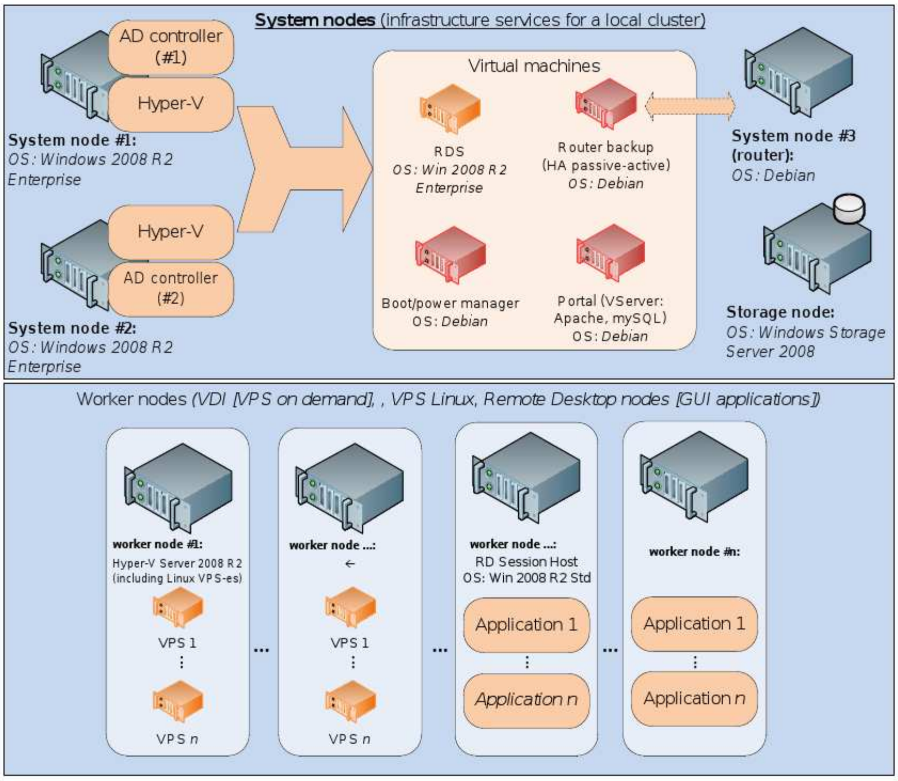

Fig. 3. Core system services of local cluster

5 Components of Front-End Layer and Reservation of Applications

5.1 User's and Administrator's Portals

In Campus Services, the management of resources and users is realized through the Internet portal implemented in the Php/MySQL technologies. Each involved center runs a local copy of the portal.

Basically the portal is divided into two parts: for users, and for administrators. The first portal allows users to register and create their accounts, as well as to reserve resources for running applications and virtual machines. Also, it gives information about available resources (hardware and software). The important component of this portal is a graphical calendar (Fig. 4), which is applied by users to reserve applications. The main functionality of the administrator's portal is management of resources and user's accounts, as well as as possibility to configure parameters which control operation of the resource scheduler.

5.2 User's Account System

In Campus Services, computing resources and applications, as well as their users, are distributed across multiple geographically distant centers. Moreover, all services are restricted to employers of research and academic institutions, as well as students. This requires to guarantee a strict identity verification when creating user's accounts, as well as storage of user's personal data. Another requirement is to give possibility of access to resources and applications located in all centers, using only a single account.

The proposed solution for the user's account system (component of the portal) is combination of centralized and distributed approaches, and is based on a global AD database for user's accounts. This base is synchronized with local databases of users, located in particular centers. The global AD database is applied only for the authentication of users, so it does not contain their personal data. These data are stored in that local database which is located in a center nearest to an user; so the user is entitled to register and create the account just in this center.

All users are assigned to different groups, which among others are used for determination of rights to executing different applications. Relations between groups are expressed as a tree, where users assigned to a child group have access to applications associated both with this group and a parent group.

5.3 Reservation of Applications

In order to use an application or virtual machine (VM), the user has to possess an active account in the system. After login and selecting an application or VM, the users is redirected to a graphical calendar (Fig. 4) available in the portal. In general, the process of reservation is performed in two stages, in collaboration with the scheduler.

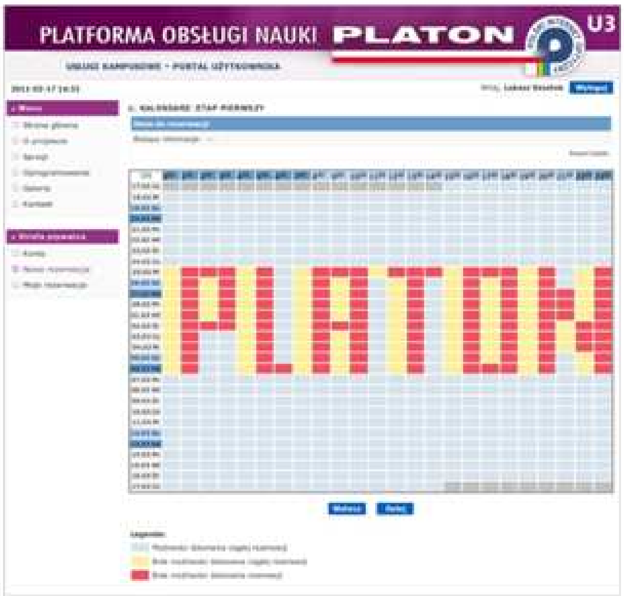

Fig. 4. Using user's portal for reservation of an application executed on local cluster

In the first stage, the user can select one (or multiple) time interval in which he would like to make a reservation, taking into account limitations placed by the administrator. The mechanism of the calendar prevents selecting intervals inconsistent with these limitations. When selecting intervals, there are displayed warnings concerning possible inconsistencies. Also, the user can look trough the limitations placed on his reservations.

Based on the results of the first stage, the scheduler searches for free resources in the second stage. The result is offering of at least one time interval for reservation of resources. The scheduler generates a single offering only in that case when it is possible to make a continuous reservation in the whole time interval selected in the first stage. In the opposite case, there are generated multiple offerings (currently up to 5) for new time intervals whose duration best fits the time interval selected by the user in the first stage. These solutions are extended by time buffers (approximately 30 minutes long) from both sides of an interval. Finally, the user chooses only one solution among the proposed offerings, and this solution is used for making reservation.

6 Resource Scheduler

The resource scheduler is one of key elements for operation of the reservation system. The scheduler mechanism was implemented in PHP using object-oriented programming. This mechanism is embedded into the user's and administrator's portals. Data required for the scheduler operation are stored in the MySQL relational database.

The main tasks of the resource scheduler are as follows:

- generation of a schedule describing availability of resources of cluster nodes in a given time interval, taking into account multiple attributes;
- performing resource allocation with consideration of adopted guidelines;
- management of reservations aimed at a better utilization of resources, and power saving;
- management of access to applications in accordance with licencing constraints.

There are two types of resources managed by the scheduler: hardware and software. Hardware resources include blade servers, and rack servers with graphics accelerators, while software resources embrace applications and virtual machines. The basic entity for scheduling is a cluster node. The whole process of the resource analysis starts with analysis of nodes, since they may feature different environments and applications.

The scheduler is responsible for the efficient utilization of resources depending on users' needs. So the analysis and allocation of resources are realized on demand, taking into account the adopted quantum of time (10 minutes). Resources are characterized by such parameters as: processor and RAM utilization requirements, necessity to use graphics accelerator, licencing requirements, type of system platform, etc.

The allocation of resources in a cluster may be performed either in a static or dynamic mode (the latter is preferred). In the static mode, the administrator is responsible for setting an environment on particular nodes. This solution limits a pool of nodes where the scheduler may allocate particular tasks. In consequence, flexibility of utilizing nodes is restricted. The dynamic (multiboot) mode permits the scheduler to decide which environment should be activated on particular nodes. This solution allows for a better flexibility and utilization of resources. For example, it becomes possible to adjust amounts of nodes in two cluster partitions, utilizing either the physical platform of MS Windows Server 2008 R2 or Hyper-V virtualization, in accordance with users' demands.

References

1. Ambrust, M. et al.: A View of Cloud Computing. Communications of the ACM **53** (4), 50 - 58 (2010)
2. PLATON - Service Platform for e-Science, http://www.platon.pionier.net.pl/online/?lang=en
3. Schaffer, H.E. et al.: NCSU's Virtual Computing Lab: A Cloud Computing Solution. IEEE **42** (7), 94 –97 (2009)
4. Srinivasan, S., Getov, V.: Navigating the Cloud Computing Landscape - Technologies, Services, and Adopters. Computer **44** (3), 22 –23 (2011)