Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Secure on-demand grid computing

M. Smith, M. Schmidt, N. Fallenbeck, T. Dörnemann, C. Schridde, B. Freisleben [*]

Department of Mathematics and Computer Science, University of Marburg, Hans-Meerwein-Street 3, D-35032 Marburg, Germany

ARTICLE INFO

Article history:
Received 2 December 2007
Accepted 12 March 2008
Available online 29 March 2008

Keywords:
Grid security
On-demand computing
Service orientation
Sandboxing
Virtualization
Trusted computing
Grid workflows

ABSTRACT

In this paper, a novel approach for enabling Grid users to autonomously install and use custom software on demand using an image creation station is presented, while at the same time offering new security mechanisms to protect both software and data from other Grid users and external attackers. An automated dynamic firewalling mechanism enables both virtual organization and user-based network security setups. Furthermore, the Grid environment is partitioned into several zones to protect local cluster resources from compromised Grid middleware. To enable the secure integration of this Grid environment into existing business processes, an extension of BPEL is presented which allows the execution of GSI secured Grid services in combination with existing business web services. The workflow engine transparently handles proxy certificate creation and monitors proxy certificate lifetime. An implementation based on the Globus Toolkit 4, the Sun Grid Engine and the ActiveBPEL Engine is presented. A performance evaluation of the critical components of the new Grid setup is provided.

1. Introduction

Grid computing [1] has become a well established method for Internet-based high-performance computing. While the first generation of Grid computing middleware solutions implemented their own proprietary interfaces, the introduction of the service-oriented computing paradigm and the corresponding web service standards such as WSDL and SOAP in the field of Grid computing through the Open Grid Services Architecture (OGSA) [2] has increased the interoperability and thus has paved the way for national and international Grid environments in which a large number of academic and a growing number of business applications can be hosted. In a commercial on-demand Grid, users and applications come and go frequently, and the economic value of both software and data is typically much higher than that in a purely academic Grid. To facilitate the on-demand usage of a Grid, it must be possible for users to autonomously install and use their applications in a timely fashion even if the software contains third-party components and requires root privileges for installation. Consequently, there are much higher requirements for administrative procedures and security mechanisms to enable on-demand Grid computing.

Like most complex software systems, Grid middleware solutions exhibit a number of security problems [3–5] which are further compounded by the new on-demand usage scenario. Not only do these security holes expose the heterogeneous Grid resources to a homogeneous attack vector, but they also threaten existing cluster resources and their users who up until now have worked in a local and secure environment. Unlike traditional cluster systems and the small academic Grid initiatives where local administrators usually know their users' software and usage habits, the larger mixed academic and business Grids expose cluster administrators to a large number of unknown users with a great variety of usage patterns. This makes the detection of malicious behavior an extremely complex task. To make matters worse, software and data are quickly becoming far more valuable than physical resources, with organizations like the Deutsche Bank, Dresdner Bank, IBM, T-Systems and BMW joining the German D-Grid initiative [6], in which our work is performed.

As a consequence, Grids are now becoming an attractive target for attackers, since the Grid offers standardized access to a large number of machines storing potentially valuable data, which can be misused in various ways. The considerable computing power of clusters exposed via the Grid can be used to break passwords and the large storage capacity can be used for storing and sharing illegal software and data. The generous bandwidth of the Internet connection is ideal for launching Denial-of-Service (DoS) attacks or for hosting file sharing services, to name just a few attacks. However, far more critical than these resource attacks are the attacks against customer data. Crash test model data of a new prototype car or a custom fluid simulation suite both represent intellectual property worth substantial amounts of money and need to be protected. If a Grid resource provider

[*] Corresponding author.
E-mail addresses: matthew@informatik.uni-marburg.de (M. Smith), schmidtm@informatik.uni-marburg.de (M. Schmidt), fallenbe@informatik.uni-marburg.de (N. Fallenbeck), doernemt@informatik.uni-marburg.de (T. Dörnemann), schriddc@informatik.uni-marburg.de (C. Schridde), freisleb@informatik.uni-marburg.de (B. Freisleben).

0167-739X/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2008.03.002

cannot ensure the end-to-end integrity and safety of customer software and data, an industrial adoption of Grid technology will not be possible. However, at the same time easy to use and unobstructive administration capabilities must exist to enable on-demand installation and usage of custom applications. These are usually opposite requirements, and careful balancing is required to fulfill both of them.

In this paper, a novel Grid environment is presented which enables users to autonomously install and use custom software (both service-oriented and traditional) on demand using an image creation station, while at the same time offering new security mechanisms to protect both software, data and business process information from other Grid users and external attackers. The solution is based on operating system virtualization and offers dynamic image creation and deployment in a secured environment. An automated dynamic firewalling mechanism offers a Virtual Organization (VO) and user-based network security setup and creates secure user network regions on demand. In addition, the Grid environment is separated into several zones to protect local cluster resources from compromised Grid middleware. The Grid headnode and the image creation station are both confined into separate compartments in a Grid demilitarized zone (DMZ). To enable the secure integration of this Grid environment into existing business workflows, an extension to the Business Process Execution Language for Web Services (BPEL) language and workflow execution engine is presented which allows the execution of the Grid Security Infrastructure (GSI) secured Grid services in combination with existing business web services. The workflow engine transparently deals with the issues of proxy certificate creation and certificate renewal (in the case of long running jobs). The presented system allows both fine grained service-oriented applications and legacy Grid applications to be run side by side through a novel integration of the secure system into existing cluster scheduling solutions. An implementation is presented based on the Globus Toolkit 4, the Sun Grid Engine and the ActiveBPEL Engine. A performance evaluation for the critical components of the new Grid setup is provided.

The paper is organized as follows. Section 2 presents the problem statement. Section 3 shows the proposed Grid architecture. Section 4 presents some implementation details and experimental results. Section 5 discusses related work. Section 6 concludes the paper and outlines areas for future research.

2. Problem statement

In this paper, we deal with security issues currently hindering commercial Grid adoption, which we encountered during our work on the German national Grid project D-Grid [6]. The aim of the D-Grid project is twofold. In the first phase, a research Grid is to be created linking the existing high-performance compute resources of German universities and research institutions in a free academic Grid. The second phase is to encourage a pay-per-use of the Grid by industrial users.

The classical computational Grid consists of a number of backend clusters running a standard cluster scheduling solution like Sun Grid Engine [7] or Torque [8] on the individual clusters. To connect the clusters to the Grid, a Grid middleware like Globus, Unicore or gLite is also installed on the cluster headnode, thus enabling direct access to the cluster scheduling system. Grid users either get a personal account on the cluster or share a pool account with a number of other Grid users. Their software must be installed locally on the cluster. This can be done in a number of ways. If the software does not require root access to be installed and the user has a local login, the user can log on to each cluster and manually install the software in his or her user account. If the user does not have login rights (which is quite often the case), the user is forced to copy the source code of the application onto the cluster using GridFTP [9] and then configure and compile the software using batch commands submitted as Grid jobs via WS-GRAM [10]. This is a painful way to install software, since each batch command (i.e. ./configure && make && make install) is submitted as a Grid job and is scheduled by the cluster scheduler. Output from the commands can be returned as the job result or can be fetched with GridFTP. Anyone who has installed moderately complex software on foreign machines can imagine the difficulties involved in installing software in this way, since it can take many iterations until all library dependencies are met. The state-of-the-art Grid fares even worse in the case of software which is not available in source code and/or requires root privileges to install (any software supplied as a Debian or Redhat package requires root privileges to install since the package managers require root privileges to run). In these cases, the users cannot install the software at all and the administrators of the local clusters must be asked to do it for them. This is an administrative hassle, not to mention the security nightmare involved in granting any unknown user software root privileges, and consequently, this will never happen. The installation process is made even more complicated if the application should offer custom service-oriented interfaces, since these custom services need to be hosted by the Grid middleware and as such should require administrative rights to be installed and run with the same rights as the rest of the Grid middleware. It should also be noted that the software of different users is installed natively in the same system. Classical Grid computing relies fully on standard operating system security mechanisms to protect users from each other. These are all factors hindering the adoption of the classical Grid in a business environment where customers want to install and use software on demand. The situation is further complicated by the introduction of service-oriented applications whose workflow-based execution can collide with traditional batch job applications both in their execution and their security requirements.

If the Grid is to fulfill the vision of becoming the next-generation Internet (as described in [11,12]), the complexity of installing and maintaining it must be reduced significantly while also increasing the level of security. One of the main functional goals of an on-demand Grid is the ease of installation of applications and services and their use on a dynamic basis for a large number of users in a secure fashion. This is a major departure from the classical Grid in which only a relatively small number of known users work in a closed system on a small range of custom software. The larger number of unknown users with the capability to autonomously install their own software creates a large number of new security issues which need to be dealt with:

- **Secure application deployment:** In an on-demand Grid, users should be capable of installing their software autonomously without endangering other users, even if root privileges are required.
- **Worker node sandboxing:** Due to the shared use of Grid resources and the sensitive nature of both user software and data, it is necessary to place each user in a separate sandbox in which the user is safe and from which the user cannot attack other users of the Grid. This includes both local issues such as file system access and process monitoring, as well as network issues, such as packet sniffing and denial-of-service attacks. The sandboxing environment must be flexible enough to support both traditional batch job security and service-oriented application security.
- **Middleware separation:** The Grid middleware is a central component in any Grid environment, offers a single homogeneous point of entry to many Grid sites and is per necessity reachable from the Internet. Thus, it is vital that the Grid headnode is as separate as possible from the local Grid resources and restricted