Proceedings of the Joint GISELA-CHAIN Conference
R. Barbera et al. (Eds.)
COMETA 2012

CLEVER: a light cloud environment for clusters and grids

Giuseppe Andronico[1], Antonio Puliafito[2], Salvatore Monforte[1], Massimo Villari[2], Davide Saitta[1], Andrea Fornaia[3]

[1]INFN Sez. Catania
{giuseppe.andronico, salvatore.monforte, davide.saitta}@ct.infn.it
[2] University of Messina
{apuliafito, mvillari}@unime.it
[3] GARR
andrea.fornaia@garr.it

Abstract

The Cloud Computing paradigm is nowadays already affirmed in the scenario of computing at least on the same foot of his ancestor, Grid Computing. Both Grids and Clouds provide access to large, distributed, computing and storage resources, but Clouds massively exploit virtualization to provide uniform interface to the underlying resource, thus hiding physical heterogeneity, geographical distribution and faults.
Similarities exist between the two computing models, but the differences are those that matter most. Grid computing is better suited for organizations with heavy applications and/or large amounts of data being analyzed by a small number of users, whereas Cloud computing is better suited to environments where there is a large number of users requesting small amounts of data, or many but small allocation requests.
Although Grid technology primarily dominates the public sector and the scientific computing environments, new interests have raised in deploying cloud technology to improve the management and reliability of those resources via the virtualization layer.
The aim of this contribution is to present an integration example of both technologies where cluster and/or Grid resources are used to easily deploy on-demand Clouds using the CLEVER innovative Cloud middleware. CLEVER 1) simplifies the access management of private/hybrid Clouds, 2) provides simple and easily accessible interfaces to interact with different interconnected Clouds, 3) deploys Virtual Machines and 4) performs load balancing through migration.
A route to CLEVER integration with a grid environment will be shown and the preliminary work reported.

Designing a solution for on-demand private Clouds rather than deploying individual products to address users' challenges leverages cluster and Grid computing in ways that extend control and flexibility, enabling the infrastructure required to fully take advantage of a dynamic computing resource model. Moreover, integrating a Cloud in a Grid has the benefit to strengthen the Cloud security with the robust federated identity and access management architecture of Grids.
The shown integration example is intended to introduce the basic concept that Grid or Cloud computing can be seen as usage interface to the same resources and that it is possible to allow users to dynamically exploit high throughput and elasticity on the basis of the contingent needs.

1. Introduction

Cloud computing is considered a successful technology leveraging the concept of Virtualization of Physical Resources for actuating useful economic of scale bringing a new level of efficiency in delivering services leading to a tempting business opportunity for IT operators of increasing their revenues. The Future Internet of Services aims to enable the deployment and delivery of complex and large-scale services to consumers with qualities of service (QoS) agreements[1].

Looking at the above requisite, in this paper we consider the opportunity of managing Virtualization Infrastructures exploiting the possibility of deploying Virtual Machines (VMs) also on Grid resources.

Grid is identified as the progenitor of Cloud Computing technology and it had a wide consensus in using it for non-commercial applications. Due to the intrinsic nature of Grid during the time it has been confined in scenarios related more to scientific research.

The idea we are describing in this paper comes from the assumption that existing hardware infrastructures supporting Grid, hold the overall characteristics for moving toward the more profit Cloud Computing IT technology. In order to reshape Grids, we think it is necessary to partially maintain the current Grid middleware, enriching it with a Virtual Machine Manager (i.e. XEN[2], VMware[3], KVM**Error! Reference source not found.**, etc.) at the lowest layer, and a new middleware able to manage VMs in Grid sites placed on top of the VMM layer.

CLEVER[5] is an innovative Cloud middleware designed and developed at the University of Messina for managing virtual appliances supplying an abstraction in the management of virtual resources providing a useful and easy management of private/hybrid clouds: by handling simple and easily accessible interfaces, it allows the possibility to interact with different "interconnected" computing infrastructures, also based on the Grid paradigm.

In the current status of the work, an early prototype, integrating some features of the original architecture design, has been accomplished and deployed in a Grid infrastructure based on the gLite Grid middleware[6]. In other words we aim to use an innovative Cloud middleware for easily enlarging the concept of resource federation to Clouds even gathering resources dedicated to Grids, which represent considerable pool of resources to be accessed and used.

We might summarize the content of this paper stating that we aim to propose a way for melting together the new emerging Cloud Computing paradigm with the well-known Grid one to match as far as possible the Future Internet of Services requirements. This approach will allow the possibility of easily providing massive physical resources on which deploying VMs also across different administrative domains. The key point in the implementation of such an infrastructure is represented by the CLEVER Cloud middleware implemented as prototype and still under development and enhancement.

The paper is organized as follows. In Section 2 we will analyze the current fully operating Grid infrastructures and the motivation that could lead Cloud operators to exploit such environments for resources federation pointing out the main features that a Cloud middleware should supply with for an easy deployment over a Grid infrastructure. The section aims to motivate the need of designing and implementing a new middleware solution that put together all the analyzed features in order to be used over Grid resources. In Section 3 we will describe the CLEVER middleware architecture, highlighting the technological solution on which it relies, its main components and how they interact each other. After the description of the CLEVER architecture, Section 4 describes how it is possible to create an hybrid computing infrastructure where CLEVER components are deployed on the Grid. The most important thing to highlight is that this task can be performed without the necessity of preemptive configurations and installations on the involved Grid components. We will also see how easy was extending the CLEVER middleware in order to interact with the Grid infrastructure services, thanks to his fully pluggable and modular design. Finally, Section 5 gives some conclusions and considerations about both the performed and the future work.

2. Motivations for exploiting Grid resources with Cloud paradigm

Giving a look at the Grid infrastructure that is currently fully operating in the worldwide picture, we see a huge number of data centers exploiting this technology. In Europe the main Grid initiative is EGI (European Grid Infrastructure[7]) which is co-funded by the European Commission. It's also present in 50 countries around the world, embracing Grid resources that are used by a wide range of research communities and industrial companies, all sharing a common federated infrastructure. More of 300 sites contribute to its infrastructure, comprising roughly 150,000 CPU cores, available to users 24 hours a day, 7 days a week, with more than 15,000 users. It also comprises more than 20 Petabytes (20 million Gigabytes) of storage, sustaining regular workloads of 150K jobs/day that reaches up to 188K jobs/day.

These statements remark how Grid currently represents a consistent part of distributed computations executing in the World. Migrate Grid to Cloud may have many added values in terms of new business models, better physical resource management, and for adopting new policy for cost and CO2 reduction and for saving energy (i.e. shutting down physical machine); Cloud middleware are aimed to accomplish these capabilities.

It is rather evident that one of main problem of Grid is the maintenance of infrastructures. Indeed, to survive and overcome these problems, Grid needs on one hand Cloud to attract funds and on the other one to meet new energetic constrains. We believe Grids can open their computation for clouds without loosing its original identity. We have also to admit that sometime Grid hardware is under of a full utilization, while other times economical constrains (energy payment, administration costs, and so on) determine the necessity for Grid promoters to provide new services for covering its costs. In that direction, the EGI project is transforming its aims to embracing also cloud functionalities, with the STRATUS LAB EU project[8]. They used OpenNebula[9] as Virtual Infrastructure Manager (VIM) and the Claudia[10] platform service management toolkit for dynamic service provisioning and scalability. Another European initiative for deploying VMs over Grid resources is given in [11], minded for Grid 5000, the main Grid French infrastructure. In both cases we mentioned, the main difficulties they addressed for the porting of cloud paradigm on Grid was about the management of the network, and the huge amounts of work needed in order to adapt canonical IaaS Cloud middleware to accomplish this scope, as we noticed in an our previous work[12].

We think that in order to actuate this migration a Cloud middleware easily deployable should be used. In our work we figure out a soft and temporary migration of Gird resources, that is some kind of “on-demand clouds”, where Grid users can leverage the usage of available resources creating their own private Clouds by submitting Grid jobs without both any intervention of site administrators and any invasive overlaying of yet another software stack installed on the existing Grid infrastructure.

Looking at the current existing middleware, which deal with the Virtual Infrastructure Management, we retain that some new features could be added within their implementation in order to achieve a system able to grant high modularity, scalability and fault tolerance: looking at the scope of this work, it could be a difficult task to change these cloud middlewares for integrating new features or modifying the existing ones in order to be used over heterogeneous resources such Grid ones.

CLEVER is able to accomplish this scope granting a higher scalability, modularity and flexibility exploiting the plug-ins concept. This means that other features can be easily added to the middleware just introducing new plug-ins or modules within its architecture without upsetting the organization.

3. The Cloud-Enabled Virtual EnviRonment (CLEVER)

CLEVER aims to provide Virtual Infrastructure Management services and suitable interfaces at the High-level Management layer to enable the integration of high-level features such as Public Cloud Interfaces, Contextualization, Security and Dynamic Resources provisioning.

The middleware is based on a distributed clustered architecture, where each cluster is organized in two hierarchical layers, as depicted in Figure 1. CLEVER nodes contain a host level management module, called Host Manager (HM). A single node may also include a cluster level management module, called Cluster Manager (CM). The CM contains the intelligence for treating and analyzing all incoming data whereas the HM has simple characteristics at lower level. Indeed it represents the remote agent of the CM. Thus, we have in the cluster at least one active CM at higher layer and, at lower layer, many HMs depending on it. A CM acts as an interface between the clients (software entities, which can exploit the Cloud) and the software running on the HMs. The CM receives commands from the clients, gives instructions to the HMs, elaborates information and finally sends results to the clients. It also performs the management of resources (uploading, discovering, etc.) and the monitoring of the overall state of the cluster (workload, availability, etc.).

Both CMs and HMs are composed by several sub-components, called agents, which are designed to perform specific tasks. Since agents are separated processes running on the same host, (due to fault tolerance purposes) internal communication is based on Inter Process Communications (IPCs), such as exploiting D-Bus interfaces. We call this Internal Remote Method Invoker (IRMI) communication, which refers to the message exchanging protocol among agents within the same manager (both in CMs and HMs).

On the other hand, External Remote Method Invoker (ERMI) communications allow to CM agents to exchange messages with the HMs ones. It is based on the XMPP protocol[13], which was born to drive the communications in the heterogeneous instant messaging systems, where it is possible to convey any type of data. In particular, the protocol is able to guarantee the connectivity among different users even with restrictive network security policies (NAT transversal, firewalling policies, etc.). The XMPP protocol is also able to offer a decentralized service, scalability in terms of number of hosts, flexibility in the system interoperability and native security features based on the use of channel encryption and/or XML encryption. The current implementation of CLEVER is based on the employment of an Ejabberd XMPP server[14].

About storing persistent information of the infrastructure, the current implementation of CLEVER is based on a specific plugin able to interact with the Sedna native XML database (see [15]). Sedna allows the possibility of creating incremental hot backup copies of the databases and supports ACID transactions. It has been also chosen because it natively supports the XML data containers.

Another important thing to report, is that CLEVER has been designed with an eye toward horizontal federation, thanks to the choice of using XMPP for module communication, made thinking about the possibility to support in the future also inter-domain communication between different CLEVER administrative domains.

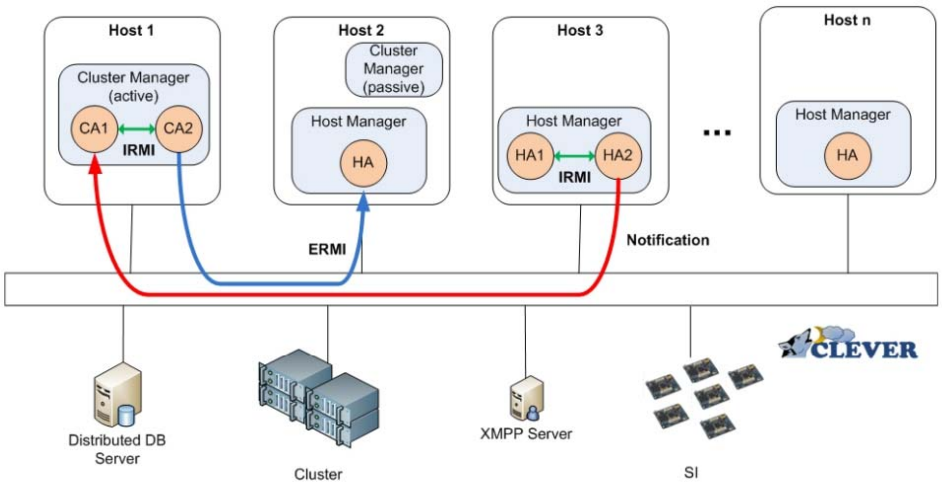

Figure 1. CLEVER middleware

4. The easy deployment of CLEVER over Grid resources

Normally the deployment of a cloud over the Grid could appear a very difficult task to accomplish, due to Grid infrastructure constraints, that do not easily allow users to manage the computing nodes for customizing its configurations: the different components of the cloud middleware have to be deployed on Grid resources, to be properly configured and made accessible over the network. Looking at the existing cloud middleware, which have as strong requirement a preliminary installation and configuration phase, the integration operations will not be so trivial. CLEVER tries to minimize the mentioned issues: it's easy to deploy because it is composed of two self-contained components, that are Host Manager and Cluster Manager, and doesn't need any particular configuration of the system on which it will be deployed: it only requires an Hypervisor and the corresponding plugin. Moreover, thanks to its "autoconf" features, once it has been deployed it is ready to run: only some parameters have to be specified such as the XMPP server.

In this section will be described the mechanisms to deploy a CLEVER-based cloud infrastructure on top of gLite[6] Grid infrastructure. It is possible to consider two different scenarios: an enterprise offering services supplying their users with private resources reserved to the Cloud within a private CLEVER infrastructure, using public Grid resources on-demand in order to handle under provisioning situations[16]; and a generic Grid user who wants to deploy his own private Cloud leveraging only Grid resources.

In this work we will consider the latter one. Summarizing, this scenario, a Grid user wants to create a private Cloud on-demand using a subset of computational resources, i.e. Worker

Nodes with virtualization capabilities, from the ones available on Grid, according to the VO he belongs to.

Therefore, deploying a CLEVER Cloud on top of Grid resources is as simply as submitting a Grid job whose executable is just the Clever Java Archive. This file contains all the packages of the cloud middleware (both the HM and the CM) and the configuration files with the minimal required options: i.e. the XMPP server hostname used to handle the communication between CLEVER components. The following figure depicts the overall architecture of a CLEVER cloud deployed on Grid resources.

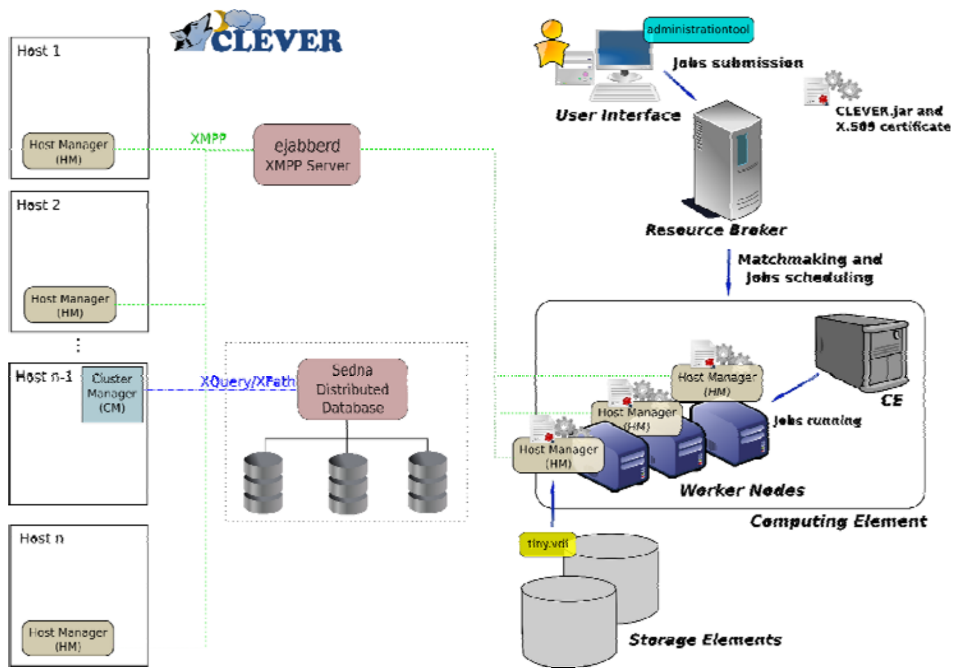

Figure 2. Deployment of CLEVER nodes on Grids resources

Once the CLEVER cloud has been deployed, users may use the CleverAdministration Client, from the node running the gLite UserInterface, for orchestrating the CMs/HMs and any created VMs as well. In other words, in order to reserve the computing resources for the user's private Cloud, a Grid user has to submit a given number of jobs, one for each Worker Node to add to the user cloud resource pool. Each job will submit a single jar file representing the entire CLEVER middleware.

There are only few requirements to satisfy in order to move a Grid site able to be used with the CLEVER middleware:

- There should be at least one configured XMPP server required for CLEVER components inter communications;
- The Sedna Database should be installed and configured at the Grid site side;
- A Hypervisor should be installed and configured at Grid job execution node (i.e. WorkerNode) that will be used for instantiating the requested VMs.

There are no requirements about the storage resources due to the CLEVER StorageManager plugin architecture that can be easily customized in order to use the relevant StorageElement users have access to. A prototype plugin has been developed for handling VMs disk images stored on StoRM[17] storage manager, which provides data management capabilities in a Grid environment to share, access, transfer data among heterogeneous and distributed data centers. Grid users, using the CleverAdministration client may upload, clone, and list VMs disk images stored within a given path on the relevant StorageElement. Authentication and Authorization are enforced using the X509 proxy certificate security mechanism, which the Globus Security Infrastructure (GSI[18]) supplies with.

Our purpose is to be able to provide isolated private clouds on-demand to any requesting authorized user. Therefore, one of the key points of deploying CLEVER private clouds on top of Grid resources is uniquely binding clever instances, namely the JAR instances, to user accessing the Grid for submitting the HMs/CMs, actually creating its own private cloud and ensuring isolation from other ones, as well. This task is accomplished thanks to the XMPP room isolation capabilities: inside the same XMPP domain, a separated room will be created for each private cloud. Therefore, in order to allow the CLEVER HMs instances running over the Grid to take part of the same private Cloud bound to the same user who has submitted the jobs running CLEVER, both the XMPP login-names, identifying each HMs/CMs running on different Grid WorkerNodes, and XMPP room names are prefixed by a string representing the MD5 of the *Issuer* plus the *SerialNumber* taken from the user's X509 proxy certificate: *(hm/cm)_<hostname>_MD5(Issuer(X509-proxy) + SerialNumber(X509-proxy)* and clever_ MD5(Issuer(X509-proxy) + SerialNumber(X509-proxy) respectively.

After the grid authentication and the submission phases are correctly accomplished, that is, the CLEVER jobs are running, we wait for the HM initialization. Using the CleverAdministration tool users should wait for the HM entrance within the CLEVER XMPP room and finally the VM can be created issuing the relevant commands. The first step to be executed consists in the VM disk-image retrieval: through the ImageManager component of an HM. The VM disk-image is downloaded from the repository (i.e. StoRM Storage Element) where it has been previously registered. Once the disk-image has been downloaded, the VM can be created and, if the VM has been correctly created, it is ready to be started.

Thanks to the CLEVER features of modularity, auto-configuration and self-contained packaging, Grid users are able to increase the number of computing resources of their private cloud infrastructure in a few minutes. If we suppose the employment of different cloud middleware that do not provide any "auto configuration" feature for and need a preliminary installation phase, the whole time needed for performing the deployment of the new cloud computing nodes on the Grid will be certainly greater than the one we employed using CLEVER.

6. Conclusions and future works

In this work, we presented an integration example of both Grid and Cloud technologies where cluster and/or Grid resources are used to easily deploy on-demand Clouds using the CLEVER innovative Cloud middleware. We described how CLEVER simplifies the access management of private/hybrid Clouds providing simple and easily accessible interfaces. Due to its design, CLEVER allows an easy setup in different administration domains. This is an important feature if we consider GRID computation resource.

In the previous section we described how CLEVER instances (i.e. HMs/CMs) could be bound in a unique way to the host on which they run and to the user running it. For example, when a room is created by the first instanced CLEVER node within the XMPP server, it should be bound to the host it is running on and to the submitting user, taking into account a hash generated from the X509 user proxy. In this way grid users may share the same Grid WN running multiple CLEVER instances bound to different users ensuring that on a given host and for a given user, there exits only one instance of CLEVER running: there is no reason to run another HM/CM on the same host for the same user.

Thanks to the wide usage of the XMPP protocol, CLEVER overcomes the issues related to the GRID networks to handle the communication among the CLEVER components.

As a future work, for achieving better integration with the Grid, the Storage Manager plugin should be enhanced so that it could be transparently used as Virtual Machine Image repository supporting image file upload/download to/from the gLite Storage Elements like LUSTRE/DPM/DCACHE/GPFS. Furthermore, the plugin should also support VMs disks snapshot management to improve the performance of deployment as well as save disk space. Also, clever Administration console commands involving interaction between Virtual Machines and Storage should handle the most common Grid Storage Protocols. To such aim collaboration with the Institute of High Energy Physics at the Chinese Academy of Sciences in Beijing has already been started.

One of the problem encountered in our test-bed is relevant to the accounting. Since the resources (i.e. CPU time and Memory) consumed by the VMs running on the HyperVisors at the Grid WN are not taken into account, and thus correctly accounted by the LocalResourceManager (e.g. PBS, LFS) executing CLEVER instances running at the Grid WorkerNodend. Hence, as a future work, to distribute resources reasonably, a robust accounting plugin should be realized, which allows for monitoring, accounting and billing of grid environment and records detail information of VMs including creation time, owner, lifetime and host on which VMs run.

References

[1] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P. Massonet, H. Munoz, and G. Toffetti, "Reservoir - when one cloud is not enough" Computer, vol. 44, pp. 44–51, 2011

[2] Xen Hypervisor - Leading open source hypervisor for servers. Retrieved June 10, 2012, from http://www.xen.org/

[3] Virtualization: the essential catalyst for enabling the transition to secure cloud computing. Retrieved June 10, 2012, from http://www.vmware.com/it/

[4] KVM: Kernel Based Virtual Machine, from http://www.linux-kvm.org/

[5] F. Tusa, M. Paone, M. Villari, and A. Puliafito., "CLEVER: A CLoud-Enabled Virtual EnviRonment," in 15th IEEE Symposium on Computers and CommunicationsS Computing and Communications, 2010. ISCC '10. Riccione, June 2010

[6] gLite - Lightweight Middleware for Grid Computing, user documentation. Retrieved June 10, 2012, from http://glite.cern.ch/end/

[7] EGEE to EGI. Retrieved on June 10 2012, from http://www.gridtalk.org/briefings/whatisegi.pdf

[8] STRATUS LAB. Retrieved January, 2012 http://www.egi.eu/community/collaborations/StratusLab.html

[9] OpenNebula: The Open Source Solution for Data Center. Retrieved June 10, 2012, from http://opennebula.org/

[10] MORFEO Claudia Project. Retrieved June 10, 2012, from http://claudia.morfeo-project.org/

[11] F. INRIA, “Deploy a virtualized environment in grid 5000.” Deploy a virtualized environment in GRID 5000. Retrieved January, 2012 *https://www.grid5000.fr/mediawiki/index.php/*

[12] G. Andronico, R. Barbera, A. Fornaia, S. Monforte. “An attempt to integrate Clouds in Grids” Proceedings of the International Symposium on Grids and Clouds and the Open Grid Forum (ISGC 2011 & OGF 31). March 19 - 25, 2011. Academia Sinica, Taipei, Taiwan. Published online at http://pos.sissa.it/cgi-bin/reader/conf.cgi?confid=133, id.45

[13] The Extensible Messaging and Presence Protocol (XMPP) protocol. Retrieved June 10, 2012, from http://tools.ietf.org/html/rfc3920/

[14] Ejabberd, the Erlang Jabber/XMPP. Retrieved June 10, 2012, from http://www.ejabberd.im/

[15] Sedna, Native XML Database System. Retrieved June 10, 2012, from http://http://www.sedna.org/

[16] F. Tusa, M. Paone, A. Puliafito, M. Villari, “CLEVER: a Cloud Cross-Computing Platform leveraging GRID resources” 2011 Fourth IEEE International Conference on Utility and Cloud Computing

[17] StoRM, Storage Resource Manager. Retrieved June 10, 2012, from http://storm.forge.cnaf.infn.it/

[18] GSI: the Grid Security Infrastructure. Retrieved June 10, 2012, from http://www.globus.org/security/overview.html