

Instant-Grid – A Toolkit for Demonstration, Test and Development of Grid-Infrastructure

Christian Boehme, Tim Ehlers, Jan Engelhardt, Andreas Félix, Oswald Haan,
Tibor Kálmán, Ulrich Schwardmann, Dietmar Sommerfeld, Alexander Willner

Abstract

Instant-Grid provides an environment for demonstration, test and development of grid applications. Using a Live-CD approach, cluster configuration and deployment of D-Grid compatible middleware is fully automated. The default installation of Instant-Grid results in a set of ready-to-use grid functionalities, which can be accessed in a beginner-friendly way via a web portal. It also includes a number of preconfigured applications, which demonstrate the uses and benefits of the grid.

1 Introduction

The grid has evolved as a promising technology for the integration of information processing in industry, science and society. In order to enhance the awareness for the possibilities of grids it is necessary to demonstrate the use of grids and the benefits of grid functionalities to potential new users and user communities.

This can be done by presentations and documentations, but it is often easier and more convincing to do this furthermore inside a test system where users can easily try the technology in a preconfigured environment directly at their working place.

In order to achieve a better awareness and a successful conviction for the possibilities of grids there are several preconditions that have to be fulfilled by such a test system. First of all the system has to be robust: it should work reliable under nearly all circumstances. The user will furthermore trust that every change to his familiar working environment will be completely reversible. The system has to be self-explanatory, it should presume no or only minimal technical knowledge of the unskilled user. Particularly it should not ask for any technical decisions. But it should be flexible enough to allow changes to the configuration for the more skillful users. And at the end clearly it should be complete in providing the main aspects of grid computing.

The Instant-Grid([1],[2]) Live-CD aims to be such a flexible and beginner-friendly demonstration, test, and development environment for grid technologies based on Globus Toolkit and especially there for the D-Grid approach.

With the Instant-Grid Live-CD computers in a local network can be rebooted and automatically configured to build a local grid (see figure 1). No previous knowledge of any of the configuration steps is assumed to form the local grid. In most cases, the setup of the boot server, compute nodes, network connection, grid middleware, and user portal work without requiring manual intervention. The Instant-Grid middleware – based on Globus Toolkit 4 (GT4)[3] – includes a complete security infrastructure and services for

job management, data handling, and information delivery. Using a Live-CD approach, cluster configuration and deployment of grid middleware is fully automated. A way to use Instant-Grid as a deployment tool for grid middleware is described in [4].

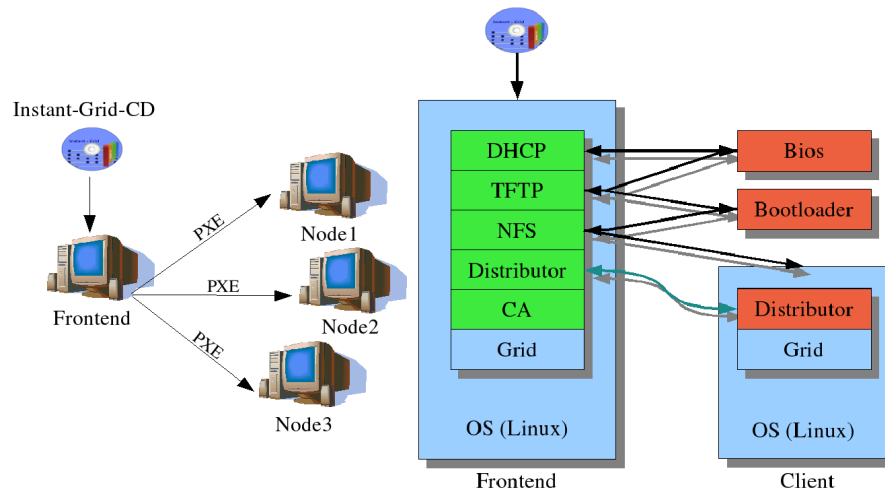


Figure 1: Configuration of an Instant-Grid

The default installation of Instant-Grid results in a set of ready-to-use grid functions, which can be accessed in a beginner-friendly way via a web portal. Integrated into the portal are interfaces for the start of applications. These applications have been chosen to be understandable for non-experts, the problems to be solved can be easily stated, and the solutions provided by the application are displayed in a self-explanatory way. Furthermore every application is a demonstration for the benefits of using grid resources and/or grid functionality. Monitoring of the availability of grid functionality and of the utilisation of the grid resources therefore is integrated in Instant-Grid.

Once started, the Instant-Grid functionalities are immediately available via a web portal. Integrated into the portal are interfaces for the start of applications. These applications have been chosen to be understandable for novices, the problems to be solved can easily be stated, and the solutions provided by the application are displayed in a self-explanatory way. Furthermore every application is a demonstration of the benefits of using grid resources and/or grid functionalities. Monitoring of the availability of grid functionalities and of the utilisation of the grid resources is therefore integrated.

The basic mechanism for the automated configuration of the hardware and the automated distribution and configuration of the grid middleware is summarized in the next section. Another focus of our present paper lies on the applications available in Instant-Grid. As shown in section 3 the applications demonstrate the value of grids as computational resources for numerical demanding applications, as a repository of distributed informations, as a tool for collaborative work, and as a virtual laboratory.

2 Features of Instant-Grid

The setup of Instant-Grid is based on the original Knoppix[5] Live-CD using the Debian GNU/linux operating system. This Live-CD serves as boot device for the frontend node of a local cluster that can be build in turn on top of the services provided by this frontend node.

2.1 Cluster and Network setup

Technically this redistribution of the operation system inside a local network is done by the well established PXE mechanism, as used in the original Knoppix as well as in Live-CD projects aimed at cluster setup. Generally, one computer boots from the CD (see figure 1) and then acts as a DHCP, TFTP, and NFS server to allow other machines on the same subnet to boot from the network and nfs-mount the CD in order to work as diskless clients.

In our implementation of this design, the main goal was to allow this process to adapt to common network setups without human intervention. This includes automatically choosing a suitable network interface firstly for the frontend node and later for the clients. All other network interfaces of the clients are deactivated so that they communicate with the internet via NAT only.

To the best of our knowledge, all other implementations of the PXE boot mechanism for clusters need additional configuration steps before starting the services.

2.2 Dynamic Resource Monitoring

An important feature of grids as opposed to clusters is that resources in a grid are assumed to be dynamic. This means that client nodes can enter or leave the grid at any point during the lifetime of the grid - in the case of the Instant-Grid as long as the frontend is available. As the basic infrastructure of the grid needs to be known on the frontend and all client nodes, a mechanism must be implemented to discover and distribute it.

For the Instant-Grid, we have developed a mechanism where the frontend probes the Instant-Grid network for available hosts, updates the configuration, and copies it to a special nfs-exported file system. On the client nodes this file system is periodically checked and so the client's knowledge of the cluster configuration is updated every minute.

2.3 Data Management

There is another important feature that uses this nfs-exported file system. An Instant-Grid user should be able to use the grid infrastructure for self defined problems and jobs. Since Instant-Grid is a Live-CD based project, the native system is left untouched and each node is usually notable to see any other than its local hard drive resources. In order to make the necessary software and data for the users own jobs available in a centralized way to the local grid, the user can use a special directory inside the hard drive partitions of the frontend node. This directory will be recognized and distributed

via the previously mentioned nfs-exported file system to all clients. Furthermore any changes on this file system, as results and other output, are still available on the frontend node, even after the instant grid is shut down and the nodes are started again with their native operating system.

On the grid level , two additional data managing services from the Globus Toolkit are installed. The first is GridFTP[6], which is an FTP server that is started automatically at boot time on all nodes of the Instant-Grid.

The second data managing service, the Reliable FileTransfer service[6] (RFT), is a web service which can monitor and control gridwide file transfers. RFT requires GridFTP for the file transfers and an SQL database for storing information about them.

Relying on the automatically generated and distributed credentials (see section 2.5), the user can easily copy files from one node to another (even using a third node) without any configuration and authentication needs by using either command line tools (grid-url-copy for GridFTP or rft for RFT) or by using a file browser portlet which is available through our GridSphere-based portal.

2.4 Portal

Some of the grid systems provide terminal based access only. In Instant-Grid the interface between the grid users and the grid middleware is based on a web portal which is hosted on the frontend. The main page automatically appears after the boot screen.

As an open-source portal framework, GridSphere[7] has been chosen . It provides a JSR-168 compliant portlet container and has a large user community. One of the key elements in GridSphere is enabling developers to use third party portlet web applications that can be run and administered within the GridSphere portlet container.

By default GridSphere is based on its own user database and has no connection with the user database of the operating system. Therefore the grid users need to be managed separately both in GridSphere and on the operating system level. Furthermore, the administrator of the grid needs to create the necessary mappings between them, and the CA administrator needs to issue certificates for the newcomers. After this the users generate their own grid credentials and load them into an online credential repository (MyProxy).

We have implemented a portlet application for grid account management, which gives the possibility to manage all of these tasks in an easy way: the visitors of the portal can create new Instant-Grid and portal accounts with one click - without any background knowledge of user management and Grid Security Infrastructure (GSI)[8].

Some other portlet based web applications are also integrated. Grid Portlets can be used for submitting jobs and managing data and credentials on the Instant-Grid. It is based on the non-webservice components of Globus Toolkit and provides many of the portlets needed to produce a production grid-portal.

2.5 Security

The resource pool and the user pool of an Instant-Grid change dynamically. Because of this, a difficult decision often has to be made between security and manageability. We tried to find a good compromise where these two goals are in conflict.

All nodes in the Instant-Grid use the Knoppix Live-CD operating system environment with the same security configuration (user database, services, etc). This makes it easier to manage the client nodes. System-wide changes (passwd file, etc.) are spread over the grid by our distribution mechanism (see section 2.1). It is common practice in Live-CDs to use a default user account. As usual, in Instant-Grid external login to this account is denied for security reasons.

The services (Globus Gatekeeper, Web Services Container, etc.) are configured to run under their own locked system account (globus). The Instant-Grid network is protected by a firewall. Furthermore the usage of a private network makes the direct access to the clients from outside impossible. However, clients are able to communicate with the outside world via NAT on the frontend (see section 2.1). Instant-Grid can be an isolated test environment and still work as an accessible grid site, too.

Grid Security Infrastructure (GSI)[8] is in use by Instant-Grid's middleware (Globus Toolkit). One of our goals is that new end entities (users, hosts, services), when they are entering into the grid, receive an X.509 end entity certificate issued and signed by the Instant-Grid Certificate Authority (CA) based on simpleCA[6]. This is made without any user interaction and fully automated (the certificate management works with OpenSSL CA functionality). When a new client boots up or when a new test user is generated, not only the hostfile and user database is updated, but also the new client or user gets its own X.509 certificate issued automatically by the Instant-Grid CA. Furthermore, the user's grid proxy credentials are generated automatically and stored in a MyProxy repository.

2.6 Information Services

Unlike cluster environments, a grid is a non-static environment where the available resources are dynamically changing during runtime. Information services are necessary to provide information about the status, resources and services of the grid. Special testing routines are provided to find out whether the formerly discovered resources and services are working correctly.

In Instant-Grid we have chosen Ganglia[9] for monitoring the hardware resources and the web service based Monitoring and Discovery System (MDS4) of the GlobusToolkit[6] to obtain information about the available services. Both Ganglia and MDS4 are started at boot time and run without user intervention.

Another monitoring tool, that is implemented as a portlet of the Instant-Grid portal, has an interface to the Grid Resource Database (GRDB) (see 2.8), and shows the availability and the load of the grid components.

2.7 Testing

We implemented a further monitoring test that checks for open ports expected in the current Instant-Grid implementation. This is a fast test for port responses that does not allow to make statements about the functionality of the services behind these ports. The results of this test are presented as a local web page.

For testing the resources and services that are discovered by the monitoring systems Instant-Grid provides a set of test applications. The output of these tests can be checked

for plausibility and correctness. Furthermore, this output is used to automatically provide the information in a uniform and concise manner to the user.

At the moment, there are two such test suites available with Instant-Grid. The first test checks the functionality of job submission using different services. It contains a job submission returning the hostname of the machine via the job submission command of Globus-GRAM[6]. Another job submission uses the web services based command of WS-GRAM, and a third task is a submission of an MPI job to check the cluster functionality.

A second test is an adaption of the grid status test developed by the Grid Computing Group at NCSA[10] to GT4 and to our needs. This test is a suite of several tests of grid services, the setup, and of the basic functionalities of Globus Toolkit.

2.8 Grid Workflow Management System

The "Fraunhofer Institut für Rechnerarchitektur und Softwaretechnik (FIRST)" developed in the context of its engagement in several grid projects the generic workflow management system (WFMS) as an enhancement of the grid job handler. Major parts of this workflow management system are implemented in Instant-Grid to enable user to develop and use workflows as description of complex grid tasks together with their dependences.

The universal internal tool to describe these workflows is the workflow description language (GWorkflowDL), a petri net based XML language for process flows. It is used on several levels of abstraction inside WFMS from simple control tasks to the specification of complete executable grid operations (see figure 2). These specifications are mainly stored inside the eXist XML database.

In Instant-Grid the workflows are visualised with the grid workflow user interface (GWUI). The Petri-Net structure of the workflow and the activation of the several components of the workflow is shown. The workflow can be started, stopped and suspended here. In the future this tool will be the main interface to choose the services needed and to design and manipulate complex workflows in a comfortable manner.

The execution of the workflow in the grid is controlled by the grid workflow execution service (GWES). It has furthermore components to analyze and verify the workflow execution.

The resource matcher chooses matching resources for execution. This is based on the properties given by the GWorkflowDL, which were described and written to the XML-DB before, and which have to be checked before execution.

Such a detailed description of the resources used in the grid is syntactically formulated with the grid resource description language (GRDL) and collected in the grid resource database (GRDB) developed for Instant-Grid by one of the authors (A. Willner)[11]. The grid resource description consists of two major components: the portlet that answers on requests of the user and the daemon that gets the relevant information about the resources from monitors like ganglia or MDS4. Both components communicate with the XML-DB via GRDL requests.

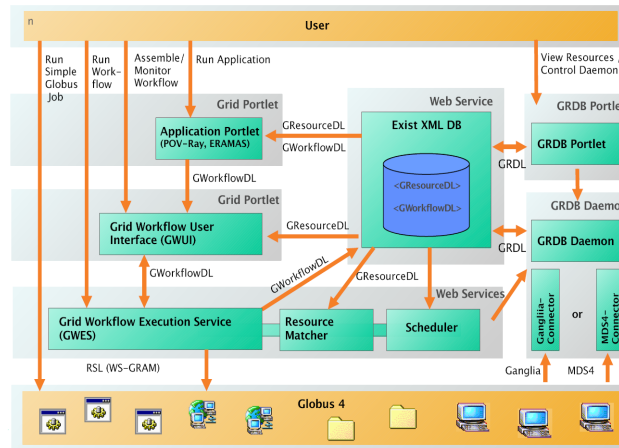


Figure 2: data flows in the workflow management system

2.9 Deployment

For every Globus specific service one can choose at boot time whether it should be started or not, or whether a package of default services should be activated. This way Instant-Grid works as a non-standard way of middleware deployment and gives the user the opportunity to adapt Instant-Grid to his own purposes at boot time. All Globus services started are automatically registered in an index service on the frontend by using Web Service Resource Framework (WSRF) compliant methods[6].

All Globus services, and hence all clients, reregister in the index service on the frontend automatically after approximately five minutes or are reregistered upon request from the frontend index service. For this the services have to be configured to know the frontend, which is also done directly in the CD image as mentioned above. The current status of the available services of the Instant-Grid therefore has a time resolution of at most five minutes.

3 Examples

3.1 Distributed Rendering

One application inside the Instant-Grid is "The Persistence of Vision Raytracer", a freely available raytracing program. Since raytracing is per se a rendering technique which can easily be parallelized, POV-Ray is often used in cluster environments for demonstration and benchmark purposes. In Instant-Grid POV-Ray is chosen to give the user an impression of the benefits of dynamic resource-management inside grids and the ability to easily setup a "render-farm" for complex 3D-scenes.

The parallel functionality of POV-Ray is given by the fact that all nodes of the grid can compute just a part of the image at the same time which can be assembled after computation to reconstruct the whole image. Communication between the user and

POV-Ray is realized by using a web-frontend based on Apache and PHP, where the user can give necessary parameters (scene describing file, resolution of the image, quality settings). All scheduling and file transfer tasks are done by the grid functionalities of Globus Toolkit.

3.2 Collaboration

As tools to describe the possibilities of collaborative work in grids two applications have been chosen and implemented in Instant-Grid.

Gobby is a realtime collaborative editor that communicates through encrypted channels and supports multiple documents in one session. In this framework each user in the grid has its own changeable color a name based on the hostname of the client such that each contribution can be identified by others. Documents can be synchronised on request.

Coccinella is a chat client with a built-in whiteboard. For this additionally a jabber server is implemented to establish a multiuser chat environment together with an inbox for instant messaging. All Instant-Grid clients are configured to form a group inside the jabber server and are automatically logged in with their hostname. The whiteboard accepts any fonts you may have installed on your system. In Instant-Grid a wide range of fonts including mathematical fonts are implemented, allowing the user to communicate mathematical formulas. The user can furthermore easily draw pictures and instantly communicate them to others. Using a library of applets it is possible to play games like chess or cards.

3.3 ERAMAS

The 'Environmental Risk and Management System' (ERAMAS) is developed by Fraunhofer FIRST together with other partners. In Instant-Grid a part of this software, the pollution transport in the atmosphere, is adapted and can be used as an example grid application. It is intended to implement this software as a workflow driven application in Instant-Grid, that results in a visualization of the expected pollution after an accident and after a certain amount of time.

3.4 GridSearch

This software is developed by FIZ-Chemie and implemented into Instant-Grid within a joint project. It builds an index of text corpora that are accessible in the grid. Whenever a new resource, like a CD or an USB-stick, is introduced into any of the clients of the Instant-Grid, the user is asked, whether an index of the texts on the media should be computed or not. Afterwards that index is accessible by every other user that collaborates in the grid and can be searched by keywords. Different formats like HTML, DOC, PDF and of course plain text are supported.

3.5 GridLabs

GridLabs is a development of the "Fernuniversität Hagen" as a system to allocate time slots of laboratory resources to users. The user interface is organized like an institute

with entry hall, reception, laboratory rooms and floors. It is possible to make a reservation of laboratories and to get into these rooms during this reserved time slot. With Instant-Grid the user can make an experiment with a small programmable computer chip inside the laboratory. Here a view of the state of the computer from outside via virtual webcam and from inside via a memory dump is possible.

4 Summary and Outlook

The Instant-Grid-CD combines the established Live-CD concept with the PXE network boot mechanism and an automated deployment of the Globus grid middleware to provide a pre-configured grid environment, usable for the development and testing of grid-applications. The CD is enhanced by a GridSphere based web portal and ready-to-use demonstration applications. The possible use of these features, without previous knowledge, broadens the audience of Instant-Grid to include unskilled users with an interest to learn more about the possible benefits of grid technology. Users can immediately get familiar with grid technology without the overhead of finding out how to install and configure the necessary software. They can build their own applications and make them work as grid applications inside Instant-Grid.

In order to fulfill the major requirement of grids to be dynamic with respect to changes of the resource and user pool, Instant-Grid has implemented update mechanisms to discover all changes of the local cluster and grid resources.

Monitoring systems are automatically working in the background of Instant-Grid. This task is done by Ganglia for the resources on the cluster level. On the grid level, the Globus MDS4 is started to monitor the resources and services.

Instant-Grid additionally has implemented several semi-automated testing suites to check the higher-level functionality of all the services that are established by the monitoring and discovery services.

For demonstration purposes, and to have a fully certified user to test the grid features, there is a completely automated authentication and authorization process of a predefined local grid user by the local Instant-Grid CA.

Instant-Grid has the ability to integrate user applications by use of a special local filesystem. The user assembles his programs before or after startup of Instant-Grid in this filesystem on the frontend. This filesystem is available on each cluster node via NFS, and therefore usable for cluster and grid applications.

Instant-Grid comes with an implemented workflow engine that uses a petri net based XML language to describe the workflows. It enables the user to start workflows and to observe the progression of the task.

There are several examples available with Instant-Grid that illustrate the broad area of possible grid applications. They include distributed rendering with POV-Ray, two applications showing collaborative editing, chat and whiteboard functionalities, a workflow based program for environmental risk management, a framework for indexing text corpora and a system to allocate laboratory resources to users.

An important issue is the requirement that these applications work – at least in principle and in a demo case – just inside the local Instant-Grid implementation. This is because we do not require internet connection or global grid connectivity, such that Instant-Grid

can work as a demo platform in nearly every environment.

On the other hand in case of an internet or global grid connection, the plan is to connect Instant-Grid to the grid and make it possible to the user working with Instant-Grid to use additional external grid resources. Here, the validity of server and user certificates and the firewall rules in such an environment is an important issue.

Acknowledgment

We gratefully acknowledge financial support for the Instant-Grid project by the Bundesministerium für Bildung und Forschung, contract number 01AK807 and we like to thank our project partners Fraunhofer Institut für Rechnerarchitektur und Softwaretechnik (Andreas Hoheisel, Thilo Ernst, Stefan Hinze, Matthias Hoche) and ed-media (Stefen Franke, Jochen Breuer) and Sergy Grebenschikov for their contributions to Instant-Grid.

References

1. Instant-Grid Project: Website of the Instant-Grid Project.
<http://www.instant-grid.de/> (2006)
2. C. Boehme, A. Felix, B. Neumair, U. Schwarzmair: Instant-Grid: Demonstration, Entwicklung und Test von Grid-Anwendungen. Praxis der Informationsverarbeitung und Kommunikation, 29.Jg., Heft 3, S. 171-177 (2006)
3. Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems. IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13 (2006)
4. C. Boehme, T. Ehlers, J. Engelhardt, A. Felix, O. Haan, T. Kalman, B. Neumair, U. Schwarzmair, D. Sommerfeld: Instant-Grid: Fully Automated Middleware-Deployment Using a Live-CD. icns, p. 70, International conference on Networking and Services (ICNS'06) (2006)
5. Knopper, K.: Website of Knoppix.
<http://www.knoppix.org/> (2006)
6. The Globus Alliance: Globus Toolkit Homepage.
<http://www.globus.org/toolkit/> (2006)
7. GridSphere: GridSphere Portal Framework.
<http://www.gridsphere.org/> (2006)
8. Global Grid Forum (GGF): Grid Security Infrastructure (GSI) Working Group Grid Forum Security Area.
<https://forge.gridforum.org/projects/gsi-wg/> (2004)
9. The Ganglia Team: Ganglia monitoring system.
<http://ganglia.sourceforge.net/> (2006)
10. Grid Computing Group at NCSA: Website of NCSA Grid Status Test.
<http://grid.ncsa.uiuc.edu/test/grid-status-test/tests.html> (2006)
11. Alexander Willner: Entwurf und Implementierung einer Ressourcen-Datenbank für das Instant-Grid-Projekt der GWDG. Masterarbeit im Studiengang Angewandte Informatik, ISSN 1612-6793, Nummer ZFI-BM-2006-36 (2006)