The Architecture of the European DataGrid

The European DataGrid Project Team
http://www.eu-datagrid.org/

Erwin.Laure@cern.ch
Contents

- Overview on the EDG Project
- EDG structure (work packages and tasks)
- Security
- Integration on the Testbed
- Future improvements
EDG Overview

- DataGrid is a project funded by European Union whose objective is to **exploit and build the next generation computing infrastructure** providing intensive computation and analysis of shared large-scale databases.

- Enable data intensive sciences by **providing world wide Grid test beds** to large distributed scientific organizations ("Virtual Organizations, VOs")

- **Start (Kick off): Jan 1, 2001**  **End: Dec 31, 2003**

- Applications/End Users Communities:
  - HEP, Earth Observation, Biology
Specific Project Objectives

- Middleware for fabric & grid management
- Large scale testbed
- Production quality demonstrations
- Collaborate and coordinate with other projects
  - Globus, Condor, CrossGrid, DataTAG, etc
- Contribute to Open Standards and international bodies
  - GGF, Industry & Research forum
DataGrid Main Partners

- CERN - International (Switzerland/France)
- CNRS - France
- ESA/ESRIN - International (Italy)
- INFN - Italy
- NIKHEF - The Netherlands
- PPARC - UK
Assistant Partners

Industrial Partners

- Datamat (Italy)
- IBM-UK (UK)
- CS-SI (France)

Research and Academic Institutes

- CESNET (Czech Republic)
- Commissariat à l'énergie atomique (CEA) – France
- Computer and Automation Research Institute, Hungarian Academy of Sciences (MTA SZTAKI)
- Consiglio Nazionale delle Ricerche (Italy)
- Helsinki Institute of Physics – Finland
- Institut de Fisica d'Altes Energies (IFAE) - Spain
- Istituto Trentino di Cultura (IRST) – Italy
- Konrad-Zuse-Zentrum für Informationstechnik Berlin - Germany
- Royal Netherlands Meteorological Institute (KNMI)
- Ruprecht-Karls-Universität Heidelberg - Germany
- Stichting Academisch Rekencentrum Amsterdam (SARA) – Netherlands
- Swedish Research Council - Sweden
Project Schedule

- Project started on 1/Jan/2001

- Testbed 0 (early 2001)
  - International test bed 0 infrastructure deployed
    - Globus 1 only - no EDG middleware

- Testbed 1 (2002)
  - First release of EU DataGrid software to defined users within the project:
    - HEP experiments (WP 8), Earth Observation (WP 9), Biomedical applications (WP 10)

- Testbed 2 (End 2002)
  - Builds on Testbed 1 to extend facilities of DataGrid
  - Focus on production quality

- EDG very successfully passed its 2nd annual EU review on February 4-5 2003

  - Advanced functionality; currently being deployed.

- Project stops on 31/Dec/2003
The EDG collaboration is structured in 12 Work Packages

- WP1: Work Load Management System
- WP2: Data Management
- WP3: Grid Monitoring / Grid Information Systems
- WP4: Fabric Management
- WP5: Storage Element
- WP6: Testbed and demonstrators - Production quality International Infrastructure
- WP7: Network Monitoring
- WP8: High Energy Physics Applications
- WP9: Earth Observation
- WP10: Biology
- WP11: Dissemination
- WP12: Management
DataGrid in Numbers

People
>350 registered users
12 Virtual Organisations
16 Certificate Authorities
>200 people trained
278 man-years of effort
100 years funded

Software
50 use cases
18 software releases
Current release 1.4.3
>300K lines of code

Testbeds
>15 regular sites
>40 sites using EDG sw
>10’000s jobs submitted
>1000 CPUs
>15 TeraBytes disk
3 Mass Storage Systems

Scientific applications
5 Earth Obs institutes
9 bio-informatics apps
6 HEP experiments
The Workload Management System (WP1)

Goals

Maximize use of resources by efficient scheduling of user jobs

The WMS is currently composed by the following parts:

- **User Interface (UI)**: access point for the user to the GRID (using JDL)
- **Resource Broker (RB)**: the broker of GRID resources, matchmaking
- **Job Submission System (JSS)**: Condor-G; interfacing batch systems
- **Information Index (II)**: an LDAP server used as a filter to select resources
- **Logging and Bookkeeping services (LB)**: MySQL databases to store Job Info
Data Management (WP2)

- **Goals**

  *Coherently manage and share petabyte-scale information volumes in high-throughput production-quality grid environments*

- **Data Management components:**
  - **Replica Catalog**: a GRID service used to resolve Logical File Names into a set of corresponding Physical File Names - **Globus Replica Catalog**
  - **Replica Manager**: the main manager for triggering replica execution all over the GRID, including replica optimization and interfacing the replica catalog service
  - **GDMP**: the GRID Data Mirroring Package, used to create replicas of any filetype all over the GRID Storage Elements in a synchronized way, by automatic updating the replica catalog
  - **Spitfire**: provides a Grid enabled middleware service for access to relational databases: it consists of the Spitfire Server module and the Spitfire Client libraries and command line executables.
GRID Monitoring and Info Providers (WP3)

- Goals

Provide information system for discovering resources and monitoring status

- Provide information about

  - The Grid itself
    This includes information about resources (ComputingElements, StorageElements and the Network), for which the Globus MDS is a common solution; and job status information (as implemented by WP1's Logging and Bookkeeping).

  - Grid applications
    This is information published by user jobs. This is used for performance monitoring.

- Globus MDS + caching

- R-GMA
Goals

manage clusters (~thousands) of nodes

Fabric Management Components

- **User Job Control and Management** (Grid and local jobs) on fabric batch and/or interactive CPU services
  - Gridification - Grid interface to fabric resources
  - Resource Management - manage underlying batch services

- **Automated System Administration** for Computing Fabric Elements (LCFG). *These subsystems are reserved for system administrators and operators for performing system maintenance*
  - Configuration Management
  - Installation Management
  - Fabric Monitoring
Mass Storage Management (WP5)

Goals

Provide common user and data export/import interfaces to existing local mass storage systems

Storage Element (SE):

- Interface to underlying Mass Storage Systems or simple storage services.
- Storage management
  - Lifetime
  - Staging
  - Access control
Network Services (WP7)

Goals
- Review the network service requirements for DataGrid
- Establish and manage the DataGrid network facilities
- Monitor the traffic and performance of the network
- Deal with the distributed security aspects

Network Monitoring Tools:
- PingER
- Udpmon
- Iperf
- NWS
Testbed (WP6)

Goals

- Deploy testbeds for the end-to-end application experiments & demos
- Integrate successive releases of the software components

Tasks

- Testing and integration of the Globus package
- Exact definition of RPM lists (components) for the various testbed machine profiles
- Perform preliminary centrally (CERN) managed tests on EDG m/w before green light for spread EDG testbed sites deployment
- Provide, update end user documentation for installers/site managers, developers and end users
- Define EDG release policies, coordinate the integration team staff with the various WorkPackage managers - keep high inter-coordination.
- Assign the reported bugs to the corresponding developers/site managers (BugZilla)
Applications (WP8-10)

Earth Observation Science Applications

Biomedical Applications

High Energy Physics
## Application Testbed Users

<table>
<thead>
<tr>
<th>VO</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMS</td>
<td>106</td>
</tr>
<tr>
<td>WP6</td>
<td>87</td>
</tr>
<tr>
<td>ALICE</td>
<td>63</td>
</tr>
<tr>
<td>ATLAS</td>
<td>55</td>
</tr>
<tr>
<td>Earth Obs.</td>
<td>29</td>
</tr>
<tr>
<td>BaBar</td>
<td>29</td>
</tr>
<tr>
<td>LHCb</td>
<td>28</td>
</tr>
<tr>
<td>ITeam</td>
<td>22</td>
</tr>
<tr>
<td>Genomic</td>
<td>22</td>
</tr>
<tr>
<td>TSTG</td>
<td>16</td>
</tr>
<tr>
<td>Medical Img.</td>
<td>6</td>
</tr>
<tr>
<td>DO</td>
<td>3</td>
</tr>
</tbody>
</table>

### Certificate Authorities Group
- Evaluates & approves new CAs
- 16 currently approved.
- Collaborating w/ other grid proj.
- More on the way...
  - Cyprus
  - US FNAL (KCA)
  - Belgium
  - Taiwan

### Virtual Organizations
- Also for Storage Elements
- Guidelines (EDG rules)

### Course-grained Authorization.

<table>
<thead>
<tr>
<th>CA</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFN (IT)</td>
<td>113</td>
</tr>
<tr>
<td>CNRS (FR)</td>
<td>71</td>
</tr>
<tr>
<td>UK</td>
<td>58</td>
</tr>
<tr>
<td>CERN (CH)</td>
<td>44</td>
</tr>
<tr>
<td>NIKHEF (NL)</td>
<td>19</td>
</tr>
<tr>
<td>Russia</td>
<td>15</td>
</tr>
<tr>
<td>US DOE</td>
<td>10</td>
</tr>
<tr>
<td>Spain</td>
<td>8</td>
</tr>
<tr>
<td>FZK (D)</td>
<td>5</td>
</tr>
<tr>
<td>Czech Rep.</td>
<td>3</td>
</tr>
<tr>
<td>Portugal</td>
<td>3</td>
</tr>
<tr>
<td>NorduGrid</td>
<td>2</td>
</tr>
<tr>
<td>Poland</td>
<td>1</td>
</tr>
<tr>
<td>Canada</td>
<td>0</td>
</tr>
<tr>
<td>Greece</td>
<td>0</td>
</tr>
<tr>
<td>Slovakia</td>
<td>0</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>352</strong></td>
</tr>
</tbody>
</table>
Certificate Request

Once in every two-three years
Certificate Signing

user

grid-cert-request

cert-request
certificate

CA

cert signing

VO

service
Registration

Once for the lifetime of the VO - you may change the certificate keys!

Usage guidelines

Account Registration

 cert-request
 cert-request
 certificate
 cert.pkcs12

convert
 registration

grid-cert-request

CA

VO
Starting a Session

- grid-cert-request
- cert-request
- certificate
- cert.pkcs12
- proxy-cert
- grid-proxy-init
- cert signing
- convert
- registration
- VO
- service
- every 12/24 hours
Certificate Request for a Host

- User
  - cert-request
  - certificate
  - cert.pkcs12
  - proxy-cert

- CA
  - cert signing

- VO
  - registration
  - convert
  - grid-proxy-init

- Service
  - host-request
  - grid-cert-request

Once in every two-three years
Signing the Certificate

- grid-cert-request
- cert-request
- certificate
- cert.pkcs12
- proxy-cert
- cert signing
- convert
- registration
- VO
- grid-proxy-init
- service
- host-request
- host-cert
- grid-cert-request
Configuration on the Server

- CA
  - grid-cert-request
  - cert signing
  - cert/crl update

- VO-LDAP
  - convert
  - registration
  - grid-proxy-init

- user
  - cert-request
  - certificate
  - cert.pkcs12

- service
  - host-request
    - host-cert
    - ca-certificate
cr

- cert signing

automatically updated every night/week
Authorization Information

CA

grid-cert-request

cert signing

cert signing

cert/crl update

user

cert-request

certificate

cert.pkcs12

proxy-cert

VO-LDAP

convert

registration

mkgridmap

grid-proxy-init

automatically updated every night/week

service

host-request

host-cert

ca-certificate

crl

crl

gridmap

convert

cert-request

The Architecture of the European DataGrid - n 28
Using a Service

The Architecture of the European DataGrid - n 29

Using a Service

The Architecture of the European DataGrid - n 29
Authorization Improvements

The Architecture of the European DataGrid - n 30

host/proxy certs and authorization information exchanged

Spitfire

LCAS

/grid

plans
Application testbed:

More than 1000 CPUs
15 Terabyte of storage

EDG sw installed at more than 40 sites

Taiwan about to join!
EDG Logical Machine Types

1. User Interface (UI)
2. Resource Broker (RB)
3. Information Service (IS)
4. Computing Element (CE)
   - Gatekeeper (Front-end Node)
   - Worker Nodes (WN)
5. Storage Element (SE)
6. Replica Catalog (RC)
dg-job-submit myjob.jdl

Myjob.jdl

Executable = "$(CMS)/exe/sum.exe";
InputData = "LF:testbed0-00019";
ReplicaCatalog = "ldap://sunlab2g.cnaf.infn.it:2010/rc=WP2 INFN Test Replica Catalog,dc=sunlab2g, dc=cnaf, dc=infn, dc=it";
DataAccessProtocol = "gridftp";
InputSandbox = {"/home/user/WP1testC", "/home/file*", "/home/user/DATA/*"};
OutputSandbox = {"sim.err", "test.out", "sim.log"};
Requirements = other.Architecture == "INTEL" && other.OpSys == "LINUX Red Hat 6.2";
Rank = other.FreeCPUs;
The Architecture of the European DataGrid - 38
The Architecture of the European DataGrid - n 38
The Architecture of the European DataGrid - n 39
The Architecture of the European DataGrid - n 30
### Grid aspects covered by EDG testbed

<table>
<thead>
<tr>
<th>VO servers</th>
<th>LDAP directory for mapping users (with certificates) to correct VO</th>
<th>Storage Element</th>
<th>Grid-aware storage area, situated close to a CE</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface</td>
<td>Submit &amp; monitor jobs, retrieve output</td>
<td>Replica Manager</td>
<td>Replicates data to one or more CEs</td>
</tr>
<tr>
<td>Job Submission Service</td>
<td>Manages submission of jobs to Res. Broker</td>
<td>Replica Catalog</td>
<td>Keeps track of multiple data files “replicated” on different CEs</td>
</tr>
<tr>
<td>Information index</td>
<td>Provides info about grid resources via GIIS/GRIS hierarchy</td>
<td>Information &amp; Monitoring</td>
<td>Provides info on resource utilization &amp; performance</td>
</tr>
<tr>
<td>Resource Broker</td>
<td>Uses Info Index to discover &amp; select resources based on job requirements</td>
<td>Grid Fabric Mgmt</td>
<td>Configure, installs &amp; maintains grid sw packages and environ.</td>
</tr>
<tr>
<td>Logging and Bookkeeping</td>
<td>Collects resource usage &amp; job status</td>
<td>Network performance, security and monitoring</td>
<td>Provides efficient network transport, security &amp; bandwidth monitoring</td>
</tr>
<tr>
<td>Computing Element</td>
<td>Gatekeeper to a grid computing resource</td>
<td>Testbed admin.</td>
<td>Certificate auth., user reg., usage policy etc.</td>
</tr>
</tbody>
</table>
Current Release 1.4

EDG 1.4.5 deployed on Application Testbed.

Positive Signs:

- Large increase in users.
- Many sites interested in joining.
- Pushing real jobs through the system.

EDG 1.4.5 deployed on Application Testbed.

HEP Simulation

Disk Usage

More Advanced MW being Deployed for Release 2.0 (May 2003)
Workload Management System

- WMS architecture reviewed
  - To apply the “lessons” learned and addressing the problems emerged with the first release of the software, deployed in the EDG testbed
    - E.g. reduce of persistent job info repositories, delegate some functionalities to pluggable modules, etc.
  - To make easier the interoperability with other Grid frameworks
  - To increase the reliability of the system
  - To support new functionalities:
    - User APIs (including a Java GUI)
    - Job checkpointing
      - Shown at the WP1 demo
    - Parallel jobs
    - Interactive jobs
    - Design and initial implementations for
      - Job partitioning; Dependencies of jobs
      - Grid Accounting
      - Resource reservation and co-allocation
Replica Management Services

- Optimization
- Replica Metadata
- Subscription

EDG-ReplicaManager

Client

File Transfer

Replica Location
Replica Location Service

**RLS:** A Framework for Constructing Scalable Replica Location Services

- Joint collaboration between WP2 and Globus
- Paper submitted to SC2002

- Independent local state maintained in Local Replica Catalogues: \( \text{LRCs} \)
- Unreliable collective state maintained in Replica Location Indices: \( \text{RLIs} \)
- Soft state maintenance of RLI state
  - relaxed consistency in the RLI, full state information in LRC

- Compression of soft states
  - compress LFN information based on knowledge of logical collections

- Membership and partitioning information maintenance
  - RLS components change over time: failure, new components added
  - Service discovery and system policies
The design of the SE follows a layered model with a central core handling all paths between client and MSS.

Core is flexible and extensible making it easy to support new protocols, features and MSS.
Information & Monitoring: R-GMA

- Use the GMA from GGF
- A relational implementation
  - Relational model is better able to describe real world systems than the hierarchical
- Applied to both information and monitoring
- Creates impression that you have one RDBMS per VO
- Not a general distributed RDBMS system, but a way to use the relational model in a distributed environment where global consistency is not important.
- Supports streaming
- System allows you to publish what you want and to find out what you want

• Producers
  • announce: SQL “CREATE”
• Producers
  • publish: SQL “INSERT”
• Consumers
  • collect: SQL “SELECT”
Fabric Monitoring & Fault Tolerance

Local Node

Central Repository

DB

Consumer

Sensor

Collector
agent

Cache

monitoring

Consumer

Actuator

Actuator

Actuator

Rule
cfg

Decision
unit

Actuator
gagent

Rule
cfg

Decision
unit

Actuator
gagent

Rule
cfg

Decision
unit

Actuator
gagent

Rule
cfg

Decision
unit

Actuator
gagent

Rule
cfg

Decision
unit

Actuator
gagent

Rule
cfg
Efficient multi-cluster job scheduling architecture
- Minimal support by batch systems required

Seamless integration with GRAM

---

Diagram:

- Maui-Scheduler
  - Retrieve node & job information
  - Start/cancel jobs

- Runtime Control System
  - Proxy Interface
    - Reserve nodes
    - Start/cancel job
    - Node, job info

- Batch System A
  - Start/cancel job
  - Node, job info

- Batch System X
  - Start/cancel job
  - Node, job info

1 2 3 = Job flow
A B C = Maintenance actions

- Set node state at batch systems

---

Work of Thomas Röblitz, Florian Schintke, Alexander Reinefeld

---

The Architecture of the European DataGrid - n 31
Installation & Configuration

- **LCFGng tailored to EDG** ("EDG LCFGng"):  
  - Component guidelines for EDG middleware developers (enforce/support use of Unix/Linux standards)
  - Produced new system management and Grid middleware components
  - Ported new client configuration access library to LCFGng
  - Non-intrusive version of the package installation subsystem (updaterpms)

- **Configuration task: first production releases of**
  - High-Level Language specification and compiler
    - Presented at USENIX LISA ’02
  - Central Configuration Database
  - Client configuration access modules
  - High-Level configuration schema

---

**Configuration Architecture**

- **admins**
- **Central Config DB**
- **HDL compiler**
- **XML**
- **cache**
- **Access API**
- **Applications**
- **Client node**
- **Client node**
- **Client node**

The Architecture of the European DataGrid - n 52
Advances are planned for all aspects of the EDG middleware.

The project is following the development of the OGSA paradigm for distributed computing.

- WebService based tools for
  - Data Management
  - Information Services, Monitoring
  - (Mass) Storage Management
Through links with sister projects, there is the potential for a truly global scientific applications grid

Demonstrated at IST2002 and SC2002 in November
More Information

- EDG Homepage
  http://www.edg.org

- WP Homepages
  http://www.edg.org/Intranet_Home.htm