gLite, the middleware for EGEE – status and future

Bob Jones
ISCG 2006, Taipei, Taiwan

www.eu-egee.org
Grid middleware

- The Grid relies on advanced software, called **middleware**, which interfaces between resources and the applications.

- The GRID middleware:
  - Finds convenient places for the application to be run
  - Optimises use of resources
  - Organises efficient access to data
  - Deals with authentication to the different sites that are used
  - Runs the job & monitors progress
  - Recovers from problems
  - Transfers the result back to the user
The Grid Software Stack

Applications

Grid

Grid Application Layer

Collective Services

Underlying Grid Services

Infrastructure Fabric

Fabric services

APPLICATIONS

MIDDLEWARE

GLOBUS Condor (VDT)

APPLICATIONS

MIDDLEWARE

GLOBUS Condor (VDT)
Key success factors for production quality software:

- **Strict software process**
  - Use industry standard software engineering methods
    - Software configuration management, version control, defect tracking, automatic build system, …

- **Conservative in what software to use**
  - Avoid “cutting-edge” software
    - Deployment on over 100 sites cannot assume a homogenous environment
      - middleware needs to work with many underlying software flavors
  - Avoid evolving standards
    - Evolving standards change quickly (and sometime significantly cf. OGSI vs. WSRF) – impossible to keep pace on over 100 sites

You will not develop and deploy your PhD project on a production Grid infrastructure
There is a long (and tedious) path from prototypes to production
• gLite
  – Exploit experience and existing components from VDT (Condor, Globus), EDG/LCG, AliEn, and others
  – Develop a lightweight stack of generic middleware useful to EGEE applications (HEP and Biomedics are pilot applications).
    • Should eventually deploy dynamically (e.g. as a globus job)
    • Pluggable components – cater for different implementations
  – Focus is on re-engineering and hardening
• Requirements through database (>400 requirements) and task forces

• Applications involved in all stages
  – Evaluate service functionality on prototypes
  – Evaluate service reliability and performance on pre-production service
  – Use the services on the infrastructure

• Technical Coordination Group (TCG)
  – Started in November 2005
  – Brings together all technical activities

• Direct US participation in middleware design team (Condor and Globus)

• Middleware Security Group
  – Brings together EGEE and other projects

• Consolidate middleware release
  – Current LCG-2.7 and gLite 1.5 distributions will be consolidated and released as gLite 3.0

• Using VDT as common basis for EGEE and OSG is very important for interoperability
  – EGEE is not only user but also contributor
  – ETICS project started Jan 1st with the goal to have a common build and test infrastructure
  – Univ. of Wisconsin is a partner
Software stack and origin of services in release 1 (simplified)

- **Computing Element**
  - Gatekeeper (*Globus*)
  - Condor-C (*Condor*)
  - CE Monitor (*EGEE*)
  - Local batch system (*PBS, LSF, Condor*)

- **Workload Management**
  - WMS (*EDG*)
  - Logging and bookkeeping (*EDG*)
  - Condor-C (*Condor*)

- **Information and Monitoring**
  - R-GMA (*EDG*)

- **Storage Element**
  - gLite-I/O (*AliEn*)
  - Reliable File Transfer (*EGEE*)
  - GridFTP (*Globus*)
  - SRM: Castor (*CERN*), dCache (*FNAL, DESY*), other SRMs

- **Catalog**
  - File/Replica & Metadata Catalogs (*EGEE*)

- **Security**
  - GSI (*Globus*)
  - VOMS (*DataTAG/EDG*)
  - Authentication for C and Java based (web) services (*EDG*)

Now doing rigorous scalability and performance tests on pre-production service
CAs, Authentication, Authorization

**Authentication**
- **Use of GSI, X.509 certificates**
  - Generally issued by national certification authorities
- **Agreed network of trust:**
  - International Grid Trust Federation (IGTF)
    - EUGridPMA
    - APGridPMA
    - TAGPMA
  - All EGEE sites usually trust all IGTF root CAs

**Authorization**
- **Until LCG-2.7.0 via grid-map files only**
- **From LCG-2.7.0 using VOMS extended proxies**
  - Call-outs to local authorization services
  - Integration with grid services underway – compute elements, storage systems
  - The authorization will be a mixture of call-outs and grid-map files until all services understand extended proxies
Basic Services

Job Management:

• **Workload Management** –
  – Resource Broker
  – DLI/SI interface to catalogues for data-based scheduling
  – Bulk job submission (gLite-3.0)
  – DAGs (gLite-3.0)
  – Push/pull mode (pull untested – gLite-3.0)

• **Compute Element (CE):**
  – Globus/EDG/LCG + Condor_C (VO-based scheduling) in gLite-3.0

• **Logging & Bookkeeping**

• **Local Batch systems:**
  – LSF, PBS, Condor, (Sun Grid Engine)

• **Additional tools:**
  – Ability to “peek” at stdout/stderr of running jobs

Data Management

• **File and replica catalogues (LFC)**
  – Central or local (not distributed)
  – Replication via Oracle, or squid caches tested by LCG
  – Secure

• **File Transfer Service (FTS)**
  – Reliable data transfer
  – Uses gridftp or srmcopy as transport

• **Storage Elements based on SRM interface**
  – DPM: implements Posix ACLs, VOMS roles/groups (gLite-3.0)
  – Other available SEs: dCache, Castor
  – Deprecated: “Classic SE” – basically just gridftp

• **Metadata catalogue:**
  – AMGA (gLite-3.0 – partial support)
gLite Software Process

Development

Software Code

Fix

Integration

Deployment Packages

Pass

Integration Tests

Fail

Testing

Testbed Deployment

Functional Tests

Installation Guide, Release Notes, etc

Pass
gLite Key Concepts

- Centered around VOs
  - It’s ultimately the VO who gets resources allocated and need to decide how to best use them (share them among the VO users)

- Distinguish between *infrastructure* and *VO* services

- Infrastructure services
  - Operated and trusted by the resource administrator
  - Implement site policies
    - Including what share of the resources are allocated to a VO
VO services

- Implement intra-VO policies
  - Scheduling, priorities, etc.
- Managed and operated by a VO
  - Typically by sites on behalf of VOs
  - A service instance may serve multiple VOs
- Currently mostly higher level services
  - Resource brokers, catalogs, …

- There is the need of deploying VO services closer to the resource
  - Better information about the resource and better control about the resource
  - Downside: more and more services to be deployed at the sites – see discussion later on
Service Oriented Architecture

Guiding Principles

Interoperability
Portability
Modularity
Scalability

Web Services

Building on existing components in a lightweight manner

AliEn   LCG   Condor
Globus   SRM   ...

EGEE-II INFSO-RI-031688
Bob Jones, ISCG 2006
In principle, Grid security requirements are not different from standard security requirements

- Users want their data and application secured (including data transfer)
- Sites want access to their resources secured and audited

What makes it challenging are the different administrative domains interconnected on the Grid and the need to establish mutual trust
• Grid security is based on **X.509 PKI infrastructure**
  – **Certificate Authorities (CA)** issue (long lived) certificates identifying individuals (much like a passport)
  – Trust between CAs and sites is established (offline)
  – User identification is done by using (short lived) **proxies** of their certificates

• **Proxies can**
  – Be delegated to a service such that it can act on the user’s behalf
  – Include **additional attributes** (like VO information via the VO Membership Service VOMS)
Middleware Challenges: Data Management

- **Heterogeneity**
  - Data is stored on different storage systems using different access technologies
  - Need common interface to storage resources
    - Storage Resource Manager (SRM)

- **Distribution**
  - Data is stored in different locations – in most cases there is no shared file system or common namespace
  - Data needs to be moved between different locations
  - Need to keep track where data is stored
    - File and Replica Catalogs
  - Need scheduled, reliable file transfer
    - File transfer and placement services

- **Different Administrative Domains**
  - Data is stored at places you would normally have no access to
  - Security and auditing implications
  - Need a common security model
    - ACLs enforcement based on Grid identities – DNs
• Computational tasks of thousands of users need to be scheduled on the available Grid resources

• Grid (Meta)Scheduling consists of:
  – Resource Discovery/Brokering
    ß Find suitable resources
  – Matchmaking
    ß Assign a job to a resource that satisfies job requirements
  – Job execution
    ß Reliably execute the jobs and retrieve output
    ß Deal with error management

• Job execution requires to find the “right” Computing Element (computing resource)
  – with maybe boundary conditions (architecture, software installed, data accessible, etc.)
• **Installation Guide**
• **Release Notes**
  - **General**
  - **Individual Components**
• **User Manuals**
  - **With Quick Guide sections**
• **CLI Man pages**
• **API’s and WSDL**
• **Beginners Guide and Sample Code**
• **Bug Tracking System**
• **Mailing Lists**
  - **gLite-discuss**
  - **Pre-Production Service**
• **Other**
  - **Data Management (FTS) Wiki**
  - **Pre-Production Services Wiki**
    - **Public and Private**
  - **Presentations**