Enabling Grids for E-science in Europe

www.eu-egee.org

JRA1 All Hands Meeting, 28-30-June-2004

Summary

Erwin Laure
EGEE Deputy Middleware Manager

EGEE is a project funded by the European Union under contract IST-2003-508833

How to proceed further …

Enabling Grids for E-science in Europe

- Very useful and productive meeting
- The following slides try to summarize the discussions (not the content of the talks – cf. agenda page http://agenda.cern.ch/fullAgenda.php?ida=a041852 for original talks)
- Issues are highlighted in red

- Out of the issues, actions will be identified and followed up by the EMT

Integration

Enabling Grids for E-science in Europe

- Common build system using ant, CruiseControl, Maven
  - Common layout and targets
  - Supports different languages and build tools
    - Java, c/c++, perl,… ant, make, autotools, …
  - Non standard building possible
    - But talk to the integration team
  - Components should be single language – this seems to be an unnecessary restriction; for jni this is not feasible at all.
    - So far this problem has not been seen – it's anyway only a recommendation and can be changed (e.g. put it on subsystem level).
  - Unit tests and packaging part of the build system (.tar – msi, rpm)
    - Need definition of the services, their deployment, run-time requirements, etc.
    - Binary packages are built from cvs – source packages should be checked as well (by building binaries from them)
  - QA: Coding guidelines, documentation, unit tests coverage
    - Coding guidelines missing
    - C++ auditing tool missing – codewizzard needs license

- Weekly builds & continuous integration
  - RHEL v3, winXP
  - Builds triggered by CVS changes
  - Build and notification intervals configurable
- Is secure access to web-based tools needed?
- 3 level hierarchy
  - Global – subsystem – component
- Integration started
  - 7 subsystems; 57 components
- Scripts for adding subsystems/components available
  - These should be checked automatically for consistency (not done yet)

- Who will provide service startup/monitoring scripts
  - If possible should come from integration team
  - See common components session

Integration
External Dependencies

Enabling Grids for E-science in Europe

- Need to agree on a common set of dependencies – will be provided in a common repository in pre-compiled version per platform
- Scalability with many dependencies?
- Hosting environment for repository?
- Some inconsistencies among different subsystems already detected – needs input from developers!
- Modified external components (patched ones) will become a new version and needs to be approved by CCB. Preferably, the patch should be pushed to the original provider and a new version should come from them.
- Internal dependencies will be put automatically in the rpm – external dependencies need to be added by provider
- Could dependencies be put into a configure script?
  - To allow services to discover them at installation time
- What to do with binary tarballs?
  - These should be specified in a dependencies file

Integration Configuration

Enabling Grids for E-science in Europe

- Common structure for gLite system
  - Common places; file type; used variables
- Guide users in selecting correct values
  - Structure configuration according to “to be changed” type
- Investigations of currently used configuration systems underway
- Try to come up with a common structure
- Common configuration vs. individual service deployment

Summary

Enabling Grids for E-science in Europe

- Documentation/information flow good enough?
- More training needed?
- Regular ITeam meetings needed

The integration team needs your help!

Testing progress

Enabling Grids for E-science in Europe

- Understand deployment procedures
  - Produce installation/configuration nodes
- RAL & NIKHEF installing components of the prototype
- Basic testing working
  - Functional testing not yet quite advanced
- Lots of testing on prototype testbed
- Savannah bugs in ‘ready for test’ state are being tested
- Continue installation testing
- In parallel start system testing

- (Bi)weekly testing of autobuild baseline

- Automatic installation for testing testbed?
  - Could kick-start be used for the moment until we converge to a more stable software stack?
  - We could already start with external dependencies using e.g. quattor.

Testing – Unit testing

Enabling Grids for E-science in Europe

- So far only IT/CZ started with unit tests – everybody needs to do that
- Naming conventions have been identified
- No interfaces defined yet – impacts interface testing
  - Should be based on tagged versions of interface files
  - Not based on xunit – could that be used?
  - Details to be discussed
- Documentation largely missing
- Delay in email response

Testing tools

Enabling Grids for E-science in Europe

- Framework candidates:
  - QMtest, LCG-tstg suite, xunit
  - No final decision yet – mid July; feedback is solicited
  - Since we need to be independent of frameworks – do we actually need one?

- Automatic deployment testing being set up (all services in a single machine – to check conflicts, incompatibilities)
  - Installation instructions not sufficient at the moment
    - AliEn, for instance, has automated testing environment that could be reused
- Prototype hard to integrate into testing environment - acts like a shell; not clear which APIs, how to do job storms, set up cron jobs etc.
- Testbed hard to install based on prototype
  - not yet working!

Prototype testing

Enabling Grids for E-science in Europe

- Testing team has show-stopping bugs on the prototype still open – impact on progress of tstg team
  - Testing team should be first customer!
- Documentation insufficient
- Synchronization with the prototype testbed difficult
- Savannah misses functionality – can we fix that or switch to something else (bugzilla?)
- Standard SCM cycle is badly needed!

Security

Enabling Grids for E-science in Europe

- Missing piece: authorization service (which is actually not a service) – Policy decision point (PDP), policy enforcement point (PEP)
- All functionality must be integrated in the services implementation
  - Languages? – start with java
  - Could be library called from the service or part of the container (extension of port type)
    - Should use container extension; but start with library which will evolve to that; this will give us more flexibility
- Need volunteers from JRA1 to work together with security in closing the gaps
  - MWSCG
  - Should start with one service per cluster – details to be defined soon
- Main issues
  - How to do delegation
  - Authentication of the client

- TLS: SOAP over HTTPS
  - Axis handler, reuse edg java-sec software + CoG
  - gSOAP plugin from IT (license?)
- MLS: should be WS-I compliant
  - Java: WSS4J from Apache
  - C: ???
- AuthZ framework
  - Refactorization of edg java-sec and LCAS/LCMAPS framework
- Mutual AuthZ
  - Upcoming
  - Should be configurable
- Encrypted data (esp. biomed)
  - Add-on service that re-uses DM infrastructure for key mgmt
  - Could also be used for licensed software

- Site proxy:
  - Routing outgoing requests to WN – would effectively be a software firewall – not feasible
  - Rather controlling of existing firewalls
  - How to deal with old globus-ssl-utils? How to evolve existing GSI connections (like WMS to LB)?
    - could evolve as GSI over HTTP? Not easily – better go straight to SOAP.
    - Short term: move from globus-ssl-utils to plain gss
- Security testing
  - Need collaboration between tstg team and JRA3 (contacts established)

Information & Monitoring

Enabling Grids for E-science in Europe

- Service discovery
  - Services announce their availability by publishing to a service table
    - Includes links to WSDL and documentation
  - Dynamic status of the service is published to the service status table
  - UDDI doesn't fit well what is needed – quite heavy
- Job scheduling
  - Push model heavily relies on I&M
  - Pull model does not rely that heavily on it since the CE will publish a classad to the WMS – needs to adhere to a schema
- Logging and bookkeeping:
  - Status updates are logged from RB/CE
  - Requires guaranteed delivery mechanisms
    - Add capabilities and attributes
  - Prototype end of the year
  - Extended towards job provenance and accounting
- Network monitoring
  - Network statistics published via I&M

- Accounting
  - Being developed for LCG
  - How does this related to proposed accounting schema
    - Could be a building block (also what is being done with L&B)
  - Need to take care not to diverge from LCG

- Job Monitoring
  - Capture the output of the job itself by either
    - Job calling I&M (e.g. using log4j)
      - Difficult – job needs to be modified
    - Job monitoring service which forwards messages to I&M
  - Not clear about use case and approach

Service bootstrapping

Enabling Grids for E-science in Europe

- Client needs to know producer/consumer service
  - Configured on installation
- Producer/Consumer services need to locate registry and schema services
  - Central server pushing information to nodes
    - Requires knowledge of all nodes at server
  - Individual nodes pull from central server
    - Requires knowledge of central server at all nodes
- Better: Configuration files configured on installation

- This is per VO – how does that confirm to lightweight VOs?
- Is there any crawling mechanism?
  - Might have consistency problems; but configuration files also cause problems
- Should be more like DNS

Matchmaking & DM

Enabling Grids for E-science in Europe

- Should input files be taken into account at all during matchmaking time?
  - Or just submit job and worry about locality later?
    - That would be too expensive in most cases
  - In any case, the system needs to survive if data is not in place anymore
    - SRM will be able to pin files, but only in mid term
- Where is fine grained authorization on files checked?
  - While matchmaking on the catalog – if the user is not allowed, no matching SEs/CEs are found. No need for the broker to check ACLs on its own.
- Where are ACLs stored
  - On the replica catalog (which will be also locally on the SE if needed)
- Does the WMS move data?
  - Yes, depending on policies
- **Is there a "central" instance knowing about replica locations (at least sites) or is this information only local at sites?**
  - **Some "central" instance is needed for the WMS!**

- What will WMS use? LFN/GUID/Metadata/SURL?
  - The WMS is just another user – so it will be LFN or GUID
- Transfer based on metadata (in particular bulk transfer)
- Output file handling
  - Done by the job wrapper calling copyAndRegister
- Local space management (on WN)
  - Could ‘tactical’ SE help?
- Is metadata handled by the job?
- Output data merging – who does that?

I/O on SE

Enabling Grids for E-science in Europe

- Provide library for POSIX I/O
  - Daemon authorizes access
- Problem with writing files: problem with synchronization
- Problem with new files: how to resolve names?
  - Should be possible with giving LFNs

Advance reservation

Enabling Grids for E-science in Europe

- Interesting, relevant work
- No reservable resources available at the moment
  - SRM implementations still don't provide pinning
  - Networks at the end of this year, earliest
- to be revisited at second year

GAS

Enabling Grids for E-science in Europe

- GAS represents the user entry point to a set of core services
  - Delegates method invocations to appropriate services
- In the long term, interfaces (GAS and service) need to converge – GAS is no excuse to freely change service interfaces
- Services are also directly accessible – services should talk directly to each other
- Requires controller service for managing the services offered to the user
  - Should be done by standard mechanisms (WSRF?) eventually
  - What's the role of the information system in that scenario
- Does GAS require additional things to normal clients/WSDL?
  - No it uses whatever interface the service provides (currently perl API or SOAP)
- Nickname use to overcome problems of changing certificates

Package Manager

Enabling Grids for E-science in Europe

- Extension of traditional package mgmt systems to distributed environments
  - Existing systems used at backend
- Only deals with VO code – not responsible for middleware and system software.
- Is there any difference between software packages and data?
  - Could use data mgmt system (would be actually advantageous) , but this might clash with some of the existing package mgmt systems
  - Could help also with licensed software
- Operation requirements need to be taken into account
- Could it be used for user code as well?

Common components

Enabling Grids for E-science in Europe

- Faults & Versioning
  - Should have a common fault base – GliteBaseFault
  - Or should we just have an inventory?
  - Backwards compatibility is required (within gLite)
    - Need policy what to do with non-understood extensions
  - How to expose the version of the service?
  - Must comply with WS-I
    - Testsuites available
- Messaging
  - Should use existing tools – which ones?
  - Relationship to techniques for overcoming outbounding connectivity restrictions
  - Who are the customers of it?
  - Needs to be integrated in security model (e.g. message level security – but costly)
- Logging
  - Logging is messaging with specific QoS
  - Need common message format
  - Need common logging levels
  - Where does logging information go
  - Are there additional requirements for accounting?

- Service administration and mgmt
  - Need common state mgmt mechanisms
  - Need common live configuration mgmt mechanisms
  - Common status monitoring, alarm plugins
  - Common administration port type per service?
  - Need to be integrated in existing ways of doing it on different platforms
  - Packaging influences what will be done

- Who is taking care of that?
  - Integration team would be the natural focal point as well as for configuration