Easy Access to Grid Infrastructures

Dr. Harald Kornmayer (NEC Laboratories Europe)

On behalf of the g-Eclipse consortium

WP11 Grid Workshop
Grenoble, France
09th of December 2008
About me!

- Background in astro particle physics
- 2001: Build a Risk Management system (SOA with JAVA)
- Forschungszentrum Karlsruhe:
  - 2003: CrossGrid
  - 2004: EGEE
  - 2006: D-Grid
  - 2006: g-Eclipse
- Since 2007:
  NEC Laboratories
  - IT Research Group
g-Eclipse – the idea

• Users want easy access to the system

• Users act within different roles
  – Grid applications users
  – Grid resource providers and operators
  – Grid application developers

• Users are middleware agnostic
  → Build a middleware independent framework

• Provide a general UI framework/eco system for the different Grid actors based on a reliable platform
  → (re-)use Eclipse and contribute!
  → gain OS independence (by using JAVA!)
• Started in 2001
  – IBM donated their Java development framework as Open Source
  – Pure Java development
  – First industrial partners joined

• Eclipse Foundation started in 2004
  – Independent non-profit organization
    • Eclipse management organization
    • Councils (requirements, architecture and planning)
    • 10 projects (including > 50 subprojects)
  – With Eclipse 3.0 not only a Java IDE, but a general framework build for extension
Can we learn from others?

• Are there other distributed systems supported by different middleware systems?
  → J2EE
    – (enterprise beans, JMS, web services, …)
• Based on specifications
  – implemented by different Vendors
    • Websphere, BEA, Oracle, JBoss, Geronimo, …
• Customers don’t want to be vendor dependent
  – Use a generic middleware independent development tool
  → Web Tools project (WTP) at Eclipse.org
    – www.eclipse.org/webtools
    – Wizards, Editors, etc to simplify the development of Web applications
    – Independent on the underlying infrastructure
g-Eclipse – projects

- www.geclipse.eu
- Project funded by the European Commission (INFSO-32347)
- 8 partners
- Until December 2008
- www.eclipse.org/geclipse
- Technology project at Eclipse Foundation
- Release 1.0.0 with stable API scheduled for next week (December 2008)
- Gathering community
  - i.e. in Grid communities
  - In Eclipse community
The reality for the Grid user

- Infrastructure for scientists were built in the past years
- Many application domains start using Grid infrastructures
- But…
  - Grid technology is complex
    - Different middleware systems are used
      - gLite, Globus, GRIA, UNICORE, …
  - Different programming paradigms
    - Batch type systems vs. service oriented systems
    - Many programming languages

→ The threshold is too high for the “standard” user!
Grid application life cycle

- In most cases, e-Users (e-Scientists, e-Engineers, e-Stock Traders) have their application(s)
  - Legacy code written in different languages (FORTRAN, C, C++, …)

- e-Users want to collaborate
  - A Virtual Organisation is build around a Virtual Computing Center on existing (and new) infrastructure

- e-Users create Grid projects

- e-Users want to interact with the Grid
  - without knowing all details!!
    (development, deployment, testing, management, …)

- Tooling is necessary!!
  - Wizards, Editors, …
  - Hide the complexity!!
Grid Middleware

- Connects resources (computing, storage, network) at different sites to one Grid infrastructure.
- Provides services to access an infrastructure.
- Common basic functionalities of middlewares:
  - Security layer for authentication and authorization.
  - Transfer protocols for accessing and managing data.
  - Brokers to distribute computing jobs on the infrastructure.
  - Higher level services (SLAs, accounting, ...).
- Many different middlewares are available:
  - gLite, Globus Toolkit, UNICORE, GRIA, ...
- g-Eclipse supports
Demo 1

- Just use Amazon

- Create a Grid project
  - Including a VO

- Access to data
  - Edit a file remotely

- Rent a machine in a few seconds
  - Login there
Use cases

user

developer

operator

submit jobs

monitor jobs

check jobs

organize workflow

replicate data

create/delete data

visualize data

build workflow

check resources

monitor resources

configure resources

benchmark resources

manage VO

debug/test application

code application

monitor application

deploy application

check jobs

check resources

check jobs

check resources

check jobs

check resources

check jobs

check resources

check jobs

check resources
Roles and Contexts

• Grid user plays different roles
  – Grid applications user
  – Grid resource provider and operator
  – Grid application developer
  – ...

• Grid user acts in different contexts
  – Virtual Organizations
  – Projects
  – ...

• g-Eclipse supports “Contextualization”
  – Depending on the user role/context a different set of tools is used by/presented to the user

• g-Eclipse supports “Customization”
  – Build the user-preferred workbench
  – Persistent over sessions
The Architecture

- Two layers:
  - Abstraction layer (Grid model):
    - Authentication/Authorization
    - Job management
    - Data management
    - Services
    - ...
  - Implementation layer:
    - Implements the model for specific middlewares/Grids
- UI is based on abstraction layer
  → UI looks the same for all middlewares
Grid project

Resource Broker

Replica Manager

Computing

Storage

Infrastructure

Programming languages

Applications

Domain services

Batch apps

Interactive apps

Workflows

Web services

Infrastructure Providers

Policies

Service Providers

Virtual Organisations

Members

Roles

Virtual Organisations
How to integrate the Grid into the Eclipse workbench?

IGridModel

IGridContainer

IGridResource

IGridElement

+ dispose(): void
+ getFileStore(): IFileStore
+ getName(): String
+ getParent(): IGridContainer
+ getPath(): IPath
+ getProject(): IGridProject
+ getResource(): IResource
+ isLocal(): boolean
+ isVirtual(): boolean

IGridRoot

IGridProject

IGridJobDescription

IGridElementManager

IVirtualOrganisation

IGridConnectionElement

IGridJob

IGridService

IGridStorage

IGridComputing
Structuring the Grid

Mounted File Systems

Virtual Organisation

Computing Elements

Services

Storage Elements
User perspective

- Grid project view
- Data Connection
- Job Descriptions
- Jobs
- VO resources
- VO computing resources
- VO services
- JSDL Editor
- VO storage resources
- Glue Info view
- GGUS Web view
- Job Status View
- Auth Token View
Demo 2

- Just use EGEE
- Create a Grid project
  - Including a VO
- Access to data
  - Edit a file remotely
- Rent a machine in a few seconds
  - Login there
Grid Resource Provider

• How can a site A support a new VO with computing resources?
  Set up a queuing system:
  – Old: Know all the details of the queuing system
  – New: Configure the batch system on site with g-Eclipse
    • Set up a VO specific queue
    • Drain queues
    • Manage Cluster nodes
Operator perspective
Grid Application development

How to “gridify” a Legacy application?
1. Develop them on your local computer as a separate JDT/CDT project
2. After a code change, compile them locally and on a remote Grid resource
3. (if needed debug them locally or on a remote Grid site)
4. Deploy the application

Usage of gLogin introduce some firewall issues!!!
Developer perspective
Developer perspective II
Visualisation

---

g-Eclipse

---

Empowered by innovation NEC
Visualisation

![Image of a visualisation software interface]
Standards

• JSDL editor
  – Multitpage editor following the OGF JSDL standard
  – Submission to different middleware possible
    - gLite:
      XSLT transformation

• GLUE schema browser
  – Browse through your resources

• Eclipse is based on OSGi
  – Enables dynamic code deployment
Interoperability

- g-Eclipse workspace can contain project with different Grid flavors
- g-Eclipse Authentication framework manage the “single sign on” on request
- g-Eclipse enables transfer from/to different Grid flavors and from/to local computer

  - By using the EFS (Eclipse Files System) implementation
Manage Complexity

- By providing solution to common problems on Grid infrastructures
  - g-Eclipse provides an extended problem reporting mechanism based on the Eclipse core exception
  - Problems have associated solutions
  - Solutions may be
    - passive: just a descriptive text
    - active: provide an action that helps the user to solve the problem, e.g. open an associated preference page
Current project status

- g-Eclipse is an official Eclipse Technology Project
- First release (0.5) available since September 2007
- Currently working in 1.0 milestone phase (RC3 available)
- Final 1.0 release planned for December 2008

- Architecture and Grid model stable since mid of 2007
- First middleware implementation (gLite) stable since end of 2007
- Second middleware implementation (GRIA) started in the beginning of 2008
- Third middleware implementation (AWS) started in Spring 2008
User communities

- Interest from many other project
  - Contact with China for GT4 implementation

- g-Eclipse will be used for university lecture in Romania

- Collaboration with other EC projects
  - i.e. SIMDAT, EGEE, DORII, IS-ENES, ...

- Collaboration with other Eclipse projects
  - Parallel Tool Platform (support for MPI on HPC resources)
  - SOA Tool Platform
  - Swordfish Runtime
Short summary for 1.0 release

• Finish second middleware implementation
  → Prove of middleware independent conception
• Implement some first industry relevant applications
  on top of g-Eclipse
  → BAE application for fluid dynamics
  → Pharmaceutical application by NEC
• Provide an interface for Amazon's EC2 and S3
  → Be not only middleware-independent but also Grid-independent
  → Introduce the world of cloud computing in Eclipse
Contribute

• Use our tool and send us feedback!
  – We do it the Eclipse way!
  – Webpage www.eclipse.org/geclipse or www.geclipse.eu
  – Newsgroup
    • http://dev.eclipse.org/newslists/news.eclipse.technology.g-eclipse/
  – Developer mailing list
    • https://dev.eclipse.org/mailman/listinfo/geclipse-dev
  – Bugzilla
    • https://bugs.eclipse.org/bugs

• Bring your application!!!!
  – Contact {at} geclipse.eu
Outlook – g-Eclipse

• g-Eclipse has the potential to become as fundamental on the Grid scientist’s desktop as the web browser was for the internet

• By making no difference between academic Grids and commercial Clouds, g-Eclipse is the tools for **Groud** computing  
  (Grid + Cloud = Groud)

• With the help of the 1919 Eclipse the theory of relativity was verified,  
g-Eclipse will help to prove …???
Outlook – Grid in general

• How will Grids will look in the 64-Core-CPU time?
  – Service instead of Batch!

• The challenge will be the management on different levels in the XaaS world (XaaS = Everything as a Service)

• Cloud/Utility will be based on Virtualization with a lot of Grid behind the scene.

→ The Heaven starts beyond the Cloud
Empowered by Innovation