European DataGrid (EDG)

- Overview over the DataGrid Project
- DataGrid Components
- A Job Submission Example
The EDG Project

- **Specific Project Objectives:**
  - Middleware for Computing:
    - Job Submission and Data Management
  - Information Systems:
    - GRID & Network Monitoring
  - Large scale testbed
  - Production quality demonstrations
  - Contribution to open Standards and international bodies

- **Applications:**
  - HEP (4 x LHC), Earth Observation, Biology

- **Start (Kick off): Jan 2001 - End: Dec 2003**
The EDG Project – Status

- **Current EDG middleware services:**
  - Job & Data Management
  - GRID & Network monitoring
  - Security: Authorization
  - Fabric & GRID management
    - Site installation and configuration tools
    - Set of common services

- **Current middleware development on:**
  - Features for job partitioning and check-pointing, billing and accounting
  - VO Management tool (VOMS)
The EDG Project – Status

- 6 sites are currently in the EDG-Testbed:
  - CERN, RAL, NIKHEF, CNAF, CC-Lyon, FZK
- Will be deployed on other EDG testbed sites
  - 37 sites wait to join
- Runs on RedHat 6.2
  - RedHat 7.2 in progress
  - Solaris in far future
- Application Status:
  - ATLAS ran production data challenge on EDG in mid September
  - CMS, LHCb, ALICE, Earth Obs. & Bio-Info. will follow ATLAS in demonstrating productions

Grid Kolloquium, 29th Oktober 2002 - Marcus.Hardt@hik.fzk.de
• DataGrid architecture functional blocks:

- **High level GRID middleware**
  - Basic Services
  - OS & Net services
  - This Talk

- **VOs common application layer**
  - LHC
  - Other apps

- **Specific application layer**
  - ALICE
  - ATLAS
  - CMS
  - LHCb
  - Other apps

- **DataGrid middleware**
- **GLOBUS 2.0r21**

**Authentication**
- Authorization
- Replica Catalog
- File transfer
- Information Providing
The EDG WMS

- The user interacts with Grid via a Workload Management System (WMS)
- The Goal of WMS is the distributed scheduling and resource management in a Grid environment.
- This allows Grid users to
  - submit/execute their jobs
  - get information about the job-status
  - retrieve the job-output
- The WMS tries to optimize the usage of resources
WMS Components

- **WMS** is composed of the following parts:
  - **User Interface (UI):** access point for the user to the GRID
  - **Resource Broker (RB):** manager of GRID resources, performing the match-making
  - **Job Submission System (JSS):** provides a reliable submission system
  - **Information Index (II):** a specialized Globus GIIS (LDAP) used by the Resource Broker to select resources
  - **Logging and Bookkeeping services (LB):** stores and provides Job Information
The RB is the core component of WMS
- It has to find the best suitable CE where the job will be executed
- Decision is based on information supplied by Data Management Service (DMS) and Information Service (IS)
- The CE chosen by RB matches the job requirements (e.g. runtime environment, data access requirements) specified via Condor CLASSADS in a JDL-file
EDG Replica Catalog (RC)

- Stores LFN/PFN mappings and additional information (e.g. filesize):
  - PFN - Physical File Name: <host><path+file name>
  - LFN - Logical File Name: logical name that may be resolved to PFNs
  - LFN : PFN = 1 : n, n>0
- Only files on storage elements may be registered
- Based upon the Globus LDAP Replica Catalog
The EDG Information Services (IS)

- The EDG have produced information providers:
  - Site information
  - The Computing Element
  - The Storage Element
  - Network Monitoring

- All of the EDG data objects are dynamic, they have a time stamp and a time to live (used by the cache mechanism) associated with them.
A Job Submission Example
A Job Submission Example

- User logs in on the UI
- User issues a **grid-proxy-init** to create a Globus proxy
- User sets up his or her JDL file
- Example of HelloWorld.JDL file:

```plaintext
Executable = ""/bin/gridtest";  
InputData = "LF:testbed0-00019";
ReplicaCatalog = "ldap://<host>/rc=test, dc=infn, dc=it"
DataAccessProtocol - "gridftp"
StdOutput = "Message.txt"
StdError = "stderr.log"
InputSandbox {"stdin.txt"};
OutputSandbox {"stdout.txt","stderr.log"};
Rank = "other.MaxCpuTime"
Requirements = other.Architecture=="INTEL" &&
other.OpSys=="LINUX" && other.FreeCpus>=4;
```
A Job Submission Example

- User issues `dg-job-submit HelloWorld.jdl`
  - and gets back from the system a unique Job Identifier (`JobId`)
- User issues `dg-job-status JobId`
  - to get logging information about the current status of his Job
- When the “OutputReady” status is reached, the user can issue `dg-job-get-output JobId`
  - and the system returns the name of the temporary directory where the job output can be found on the UI machine.
A Job Submission Example

- Replica Catalogue (RC)
- Information Service (IS)
- Resource Broker (RB)
- Job Submission Service (JSS)
- Storage Element (SE)
- Compute Element (CE)
- Logging & Book-keeping (LB)

Grid Kolloquium, 29th Oktober 2002 - Marcus.Hardt@hik.fzk.de
Matchmaking: List of PFNs (RC), CEs & SEs (IS)

Replica Catalogue (RC)

Information Service (IS)

Resource Broker (RB)

Job Submission Service (JSS)

Storage Element (SE)

Compute Element (CE)

Logging & Book-keeping (LB)

UI JDL

submitted

waiting

Grid Kolloquium, 29th Oktober 2002 - Marcus.Hardt@hik.fzk.de
Job handover to JSS; LB notification

UI JDL

Logging & Book-keeping (LB)

Replica Catalogue (RC)

Information Service (IS)

Resource Broker (RB)

Job Submission Service (JSS)

Compute Element (CE)

Storage Element (SE)
JSS: submits to CE, LB notification; BrokerInfo

Logging & Book-keeping (LB)

Compute Element (CE)

Storage Element (SE)

BrokerInfo

Resource Broker (RB)

Job Submission Service (JSS)

Replica Catalogue (RC)

Information Service (IS)

submitted

waiting

ready

scheduled

UI JDL
CE: get InputSandbox from RB, submit to farm

Replica Catalogue (RC)

Information Service (IS)

Input Sandbox

Resource Broker (RB)

Job Submission Service (JSS)

Storage Element (SE)

Compute Element (CE)

Logging & Book-keeping (LB)

submitted

waiting

ready

scheduled

running
CE keep LB up to date; UI: Query LB

Logging & Book-keeping (LB)

Replica Catalogue (RC)

Information Service (IS)

Resource Broker (RB)

Job Submission Service (JSS)

Storage Element (SE)

Compute Element (CE)

submitted

waiting

ready

scheduled

running

Job Status

Grid Kolloquium, 29th Oktober 2002 - Marcus.Hardt@hik.fzk.de
Job done

Replica Catalogue

Information Service

Resource Broker

Job Submission Service

Storage Element

Compute Element

Logging & Book-keeping

Job Status
CE: Send OutputSandbox to RB

UI JDL

Replica Catalogue

Resource Broker

Job Submission Service

Output Sandbox

Information Service

Compute Element

Storage Element

Job Status

Logging & Book-keeping

submitted

waiting

ready

scheduled

running

done

outputready
Replica Catalogue (RC)

Information Service (IS)

Resource Broker (RB)

Job Submission Service (JS)

Compute Element (CE)

Storage Element (SE)

Logging & Book-keeping (LB)

Output Sandbox

submitted

waiting

ready

scheduled

running

done

outputready

cleared

dg-job-get-output <job-id>
[hardt@ui001:~]$ **dg-job-submit HelloWorld.jdl**

Connecting to host lxshare0381.cern.ch, port 7771

Logging to host lxshare0381.cern.ch, port 15830

******************************************************************************

**JOB SUBMIT OUTCOME**

The job has been successfully submitted to the Resource Broker.

Use `dg-job-status` command to check job current status. Your job identifier (dg_jobId) is:


******************************************************************************

Retrieving Information from LB server https://lxshare0381.cern.ch:7846

Please wait: this operation could take some seconds.

BOOKKEEPING INFORMATION:

Printing status info for the Job:


Status = OutputReady


Job Destination = ce001.crossgrid.fzk.de:2119/jobmanager-pbs-workq

Status Reason = terminated

Job Owner = /C=DE/O=GermanGrid/OU=CrossGrid/CN=Marcus Hardt


**************************************************************************

JOB GET OUTPUT OUTCOME

Output sandbox files for the job:


have been successfully retrieved and stored in the directory:

/shift/lxshare072d/data01/Ulhome/hardt/result/12183940774010

**************************************************************************

[hardt@ui001:~]$ more result/12183940774010/stdout.txt

Hello World

[hardt@ui001:~]$ more result/12183940774010/stderr.log
Further Information

- The EDG Tutorial
  

- The EDG User’s Guide

- WMS and JDL
  http://www.infn.it/workload-grid

- ClassAd
  https://www.cs.wisc.edu/condor/classad

- This Talk
  https://gridportal.fzk.de/talks
• BACKUP SLIDES
User Interface Commands (UI)

- `dg-job-list-match`
  - lists resources matching a job description
- `dg-job-submit`
  - submits a job
- `dg-job-cancel`
  - cancels a given job
- `dg-job-status`
  - Displays the status of the job (submitted, waiting, ready, scheduled, running, chkpt, done, outputready, aborted, cleared)
- `dg-job-get-output`
  - returns the job-output to the user
- `dg-job-get-logging-info`
  - displays logging information about submitted jobs
- `dg-job-id-info`
  - is a utility for the user to display job info in a formatted style
The EDG Replica Manager (UI)

- UserInterface-Tool for replication and registration of files in the RC
- Keeps RC consistent with stored data
- Commands:
  - grid-info-search
  - CopyAndRegister
  - ReplicateFile
  - DeleteFile(s)
  - DeleteLFN
**Globus-url-copy**

- **Low level tool for secure copying**

  globus-url-copy <protocol>://<source file> \
  <protocol>://<destination file>

- **Main Protocols:**
  - gsiftp – for secure transfer, only available on SE and CE
  - file – for accessing files stored locally on UI, WN

  globus-url-copy file://`pwd`/file1.dat \
  gsiftp://lxshare0222.cern.ch/ \
  flatfiles/SE1/EDGTutorial/file1.dat
What have HEP experiments already done on the EDG testbed

- The EDG User Community has actively contributed to the validation of the first EDG testbed (Nov 2001 - Feb 2002)
- All four LHC experiments have ran their software (although in some cases in a preliminary version) to perform the basics operations supported by the testbed 1 features provided by the EDG middleware
- Validation included job submission (JDL), output retrieval, job status query, basic data management operations (file replication, register into replica catalogs), check of possible s/w dependencies or incompatibility (e.g. missing libs, rpms) problems
- Everything has been reported in

  “testbed 1 assessment by HEP applications” (D8.2)
  DataGrid-08-D8.2-0111-3-1

The first ALICE simulated event on the testbed
An (incomplete) list of the HEP-related executables

- Aliroot : generate, Display ALICE events
- DICE : generate ATLAS events
- Phythia, CMSIM, ORCA: generate CMS events
- Brunel, GAUDI, SICBMC: generate LHCb events
- PAW, PATCHY, CERNlibs: use CERN common lib analysis programs
- ROOT : object oriented framework for data analysis and data access, storage
- Objectivity : OODBMS
- GEANT3 : event reconstruction for simulated data