SAMGrid as a Stakeholder of FermiGrid

Valeria Bartsch
Computing Division Fermilab
Overview

- uses
  - CMS
  - CDF
  - DZero

- example: SAMGrid used by 2 stakeholders
- interfaces to our campus grid Fermigrid
Overview of Fermilab

- **Mission:** High-Energy Physics, the science of matter, space and time.
- **Accomplishments:** Research at Fermilab has led to scientific discoveries and technological advances: discovery of the top, bottom quark, ...
- **Current Experiments:** CDF, DZero, Minos
- **Contributions to:** CMS, SDSC, Pierre Auger, ...
Requirements for our GRID

⇒ Data collected by each running experiment
1 PetaByte p.a.
⇒ experiments with a High Energy physics user community

Consequences for our GRID:
✓ data driven GRID
✓ interoperability possible
Basic SAMGrid Requirements

► Transfer enormous amounts of data needed for different activities (scalable)
► ... sometimes over large distances and with commodity hardware (robust)
► Maintain knowledge of what we are doing and what we did (monitoring and bookkeeping)
► Maximize use of our resources (efficient)
► Don’t want to know the details [where files sit, where jobs run] (helpful)

Solution...
► A data handling and job management system
► **SAM + JIM = SAMGrid**
► SAM used by CDF, SAMGrid used and developed by DZero
The Data Handling: SAM

Job Management

Information and Monitoring

DH Resource Management

Data Delivery and Caching

Implementation Or Library

Principal Component

Service

Information
World wide distribution of SAM stations

- main consumption of data still central
- remote use on the rise

DZero and CDF:
- 10k/20k Files declared/day
- 15k Files consumed/day
- 8 TByte of Files cons./day
The Job Management (JIM)

Job Management
- JH Client
- Condor MMS
- Job Scheduler
- Site Gatekeeper
- Batch System
- Principal Component
- Service
- Condor-G
- Condor MMS
- Request Broker
- (All) Job Status Updates
- GRAM
- Compute Element Resource
- Grid sensors

Information and Monitoring

Data Handling

Implementation Or Library

Information
Overview

- CMS
- CDF
- DZero
- general purpose
- storage

- example of SAMGrid for 2 stakeholders
- interfaces to our campus grid Fermigrid
Bilateral Interoperability Matrix

<table>
<thead>
<tr>
<th>resource stakeholder</th>
<th>CDF</th>
<th>US CMS</th>
<th>DZero</th>
<th>General Purpose Farm</th>
<th>OSG</th>
<th>Fermilab Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDF</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>progress</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>USCMS</td>
<td>X</td>
<td>√</td>
<td></td>
<td>progress</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>DZero</td>
<td>X</td>
<td>√</td>
<td>√</td>
<td>progress</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>GP Farm</td>
<td>X</td>
<td>√</td>
<td></td>
<td>progress</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>OSG</td>
<td>X</td>
<td>√</td>
<td></td>
<td>progress</td>
<td>v</td>
<td>v</td>
</tr>
</tbody>
</table>

⇒ work to do for a joint use of resource
⇒ will focus on the CDF/DZero effort which use a common data handling system
FermiGrid – 4 Components

- Common Grid Services
- Fostering Stakeholder Bilateral Interoperability
- Development of Open Science Grid (OSG) Interfaces
- Exposure of the Permanent Storage System to the Open Science Grid
FermiGrid – Common Grid Services

- accept jobs
- schedule these jobs for execution
- utilize several Common Grid Services

- VOMS server
  - maps UID
  - site control
- Gatekeeper
  - GUMS identity mapping service
  - SAZ site authorization service
- Job manager
- Job scheduler
FermiGrid – stakeholder bilateral interoperability

Fermilab VO Users → FermiGrid Gateway & Central Services → OSG “guest” VO Users

resource head node
worker nodes

resource head node
worker nodes

resource head node
worker nodes

resource head node
worker nodes
Testbed of SAMGrid forwarding mechanism / LCG

Forwarding Node
- Gatekeeper
- Job Managers
- Batch system adapter
- LCG handler
- LCG User Interface

GRAM head node

LCG Resource Broker

SAM
FermiGrid: OSG gateway

Forwarding Node
- Gatekeeper
- Job Managers
- Batch system adapter
- Job Scheduler Handler

Condor SchedD

GRAM head node

Condor Match Maker
FermiGrid – Milestones & Time Line

Work done:
- End Feb 2005: Hardware installation
- May 2005: Common Grid Services available
- Summer 2005: Gridify the General Purpose Farm

Current Work:
- Work with DZero to transition from static gridmap files to use FermiGrid
- Common Grid Services

Plans 2006:
- Install general purpose desktop cluster
- Add general availability storage
- Service failover
Acknowledgement

We are standing on the shoulders of giants.

Current members of the GRID development teams:

Running experiments
Krzysztof Genser       Andrew Baranovski       Steve Sherwood
Stephen White         Dehong Zhang         Rober Illingworth
Adam Lyon             Art Kreymer           Randolph Herber

Grid software
Gabriele Garzoglio    Parag Mhashilkar      Sinisa Veseli
Lauri Loebel-Carpenter

students
Anoop Rajendra        Sudhamsh Reddy

DZero LCG integration effort
Remote SAM shifters and of course our predecessors
Additional slides

This part of the slides not officially released by Computing division, just reminder in case questions
Computing Model

Remote Farms → Central Farms

Data Handling Services

Remote Analysis Systems

Central Analysis Systems

Central Storage

User Desktops

Raw Data
Reco Data
Reco MC
User Data
Stakeholder of our campus grid

- running experiments: CDF and DZero
- experiment in preparation (for CERN, CH): US-CMS
- smaller experiments*: e.g. Minos
- storage systems
- OSG (Open Science Grid)

⇒ Common interfaces for stakeholders needed
⇒ Management on the level above each system
⇒ first step: SamGrid common for CDF, DZero and Minos

* From the data handling point of view
Fermilab: example of computing resources

- **CPU:**
  - CDF analysis farms: 3300 GHz CPU, 370TByte disks
  - DZero analysis farms: 400 dual processing, however Condor system on desktops
  - CMS Tier1 center cur.: 3000 S12K, 1100TByte disks

- **General Purpose Farm:**
  - Permanent storage: using DCache / Enstore
  - 3PByte on tape
  - Daily usage ~10TByte

![Graph showing data transfer](image.png)
Global Collaboration

Remote Facilities

SAM station

✅ provides transfer mechanism
✅ handles disks

Central Systems

SAM station

✅ provides transfer mechanism
✅ handles disks

copy file

store file

gets location

Sequential Access Via Metadata

db servers

database

✅ monitors all locations
✅ metadata of files
✅ bookkeeping of analysis jobs
Components of a SAM Station

SAM is a distributed data movement and management service: data replication is achieved by the use of disk caches during file routing.

SAM is a fully functional meta-data catalog.
Remote Facilities

- provides user analysis, MC generation, reprocessing for DZero
- different stages of services: for users at own institutes, for users of own experiment, opportunistic use of GRID systems

Central Storage

- dCache: developed in collaboration with DESY (Hamburg)
- enstore robots

Central Systems

- still major facilities for user analysis
- CDF: 1000 GHz CPU, DZero: ....
- CDF: reprocessing farms

Sequential Access Via Metadata & Job&Information Monitoring
SAM-Grid Diagram

Flow of: job data meta-data

User Interface

Submission
Global Job Queue
Grid Client

Resource Selector
Match Making

Cluster
Data Handling
SAM Station (+other servs)
Worker Nodes

Local Job Handling
Grid Gateway
Grid/Fabric Interface
JIM Advertise

Site
Info Manager
MDS
XML DB server

Global DH Services
SAM DB Server

Web Serv
Grid Monitoring
User Tools
CAF for CDF

- Condor batch system
- kerberos authentification
- input/output tarball
- new development: use Condor-Glideins (condor_g)

⇒ utilization of LCG resource possible without software installations on worker nodes
⇒ still CDF software needs to be around but very flexible for users, authorization a bit awkward
⇒ at the moment no thoughts of interoperability
⇒ very good monitoring available
Condor-G to Globus 2

Submit Machine
- Schedd
- Gridmanager
- GAHP
- Submit

Remote Machine
- inedt
- Gatekeeper
- Jobmanager
- Local Batch
CDF deployment challenges

<table>
<thead>
<tr>
<th></th>
<th>CDF requirements / s</th>
<th>DZero max / s</th>
</tr>
</thead>
<tbody>
<tr>
<td>project starts</td>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>process starts</td>
<td>1</td>
<td>0.09</td>
</tr>
<tr>
<td>file delivery</td>
<td>8</td>
<td>0.44</td>
</tr>
<tr>
<td>file storage</td>
<td>50</td>
<td>0.01</td>
</tr>
</tbody>
</table>

DZero started with SAMGrid as sole data handling (DH) system
⇒ DH was allowed evolution
CDF had another DH system
⇒ specific requirements on SAM for the central systems
⇒ lesson learnt: hard to change DH system on the fly
Monitoring and Information: the glue