Alice Grid Activities

Peter Malzacher (GSI) for the Alice Grid Project
P.Malzacher@gsi.de

Seminar Datenverarbeitung in der Hochenergiephysik
DESY, 27.5.2002
ALICE GRID Project

G. Andronico (CT), T. Anticic (IRB), R. Barbera (CT),
I. J. Bloodworth (Birmingham), M. Botje (NIKHEF), P. Buncic (CERN),
F. Carminati (CERN), P. Cerello (TO), G. Chabratova (DUBNA),
J. Cleymans (Capetown), J. G. Contreras (MERIDA), D. DiBari (BA),
DeGirolamo (BA), K. Flurchick (OSC), J. Gosset (SACLAY),
A. Grigoryan (Yerevan), D. Johnson (OSC), M. L. Luvisetto (BO),
P. Malzacher (GSI), M. Masera (CERN/TO), A. Masoni (CA),
K. Mkoyan (Yerevan), D. Mura (CA), L. Nellen (UNAM), B. Nilsen (OSU),
G. Odinyek (LBNL), D. Olson (LBNL), S. K. Pal (Calcutta), F. Pierella (BO),
F. Rademakers (CERN), I. Sacreida (NERSC), P. Saiz (CERN),
Y. Schutz (SUBATECH), K. Schwarz (GSI), E. Scomparin (TO),
M. Sitta (TO), R. Turrisi (PD), D. Vicinanza (SA)

38 people, 21 institutions

http://www.to.infn.it/activities/experiments/alice-grid
Overview

- **LHC computing challenge**
  - MONARC, EU DG, LCG

- **The Grid metaphor**
  - Globus

- **Alice Grid Activities**
  - The Alice Offline Project
  - Remote job submission and monitoring
  - AliEn
  - Interface Root-Globus, Root-Alien, Root-...
  - Distributed analysis with proof

- **Summary**
Overview

- LHC computing challenge
  - MONARC, EU DG, LCG

- The Grid metaphor
  - Globus

- Alice Grid Activities
  - The Alice Offline Project
  - Remote job submission and monitoring
  - AliEn
  - Interface Root-Globus, Root-Alien, Root-...
  - Distributed analysis with proof

- Summary
LEP (end 2000) -> LHC (start 2007)
The LHC Detectors

ATLAS

CMS

LHCb

Alice
### Summary of Computing Capacity Required for all LHC Experiments in 2007


*(ATLAS with 270Hz trigger)*

<table>
<thead>
<tr>
<th></th>
<th>CERN</th>
<th>Regional</th>
<th>Grand</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Tier 0</strong></td>
<td><strong>Tier 1</strong></td>
<td><strong>Total</strong></td>
</tr>
<tr>
<td>Processing (K SI95)</td>
<td>1,727</td>
<td>832</td>
<td>2,559</td>
</tr>
<tr>
<td>Disk (PB)</td>
<td>1.2</td>
<td>1.2</td>
<td>2.4</td>
</tr>
<tr>
<td>Magnetic tape (PB)</td>
<td>16.3</td>
<td>1.2</td>
<td>17.6</td>
</tr>
</tbody>
</table>

1 SPECint95 = 10 CERNunits = 40 MIPS
today dual PIII ~ 100 SPECint95
The MONARC RC Topology

CERN - Tier 0

Tier 1
- FNAL
  - Lab a

Tier 2
- RC...
  - Uni b
  - Lab c
  - Uni n

IN2P3

Bandwidths:
- 2.5 Gbps
- 622 Mbps
- 155 Mbps

Department

Desktop
HEP Data Grid Initiative

- **European level coordination of national initiatives & projects**
- **Principal goals:**
  - Middleware for fabric & Grid management
  - Large scale testbed - major fraction of one LHC experiment
  - Production quality HEP demonstrations
    - "mock data", simulation analysis, current experiments
  - Other science demonstrations
- **Three year phased developments & demos**
- **Complementary to other GRID projects**
  - **EuroGrid**: Uniform access to parallel supercomputing resources
- **Synergy to be developed (GRID Forum, Industry and Research Forum)**
## Work Packages

<table>
<thead>
<tr>
<th>WP</th>
<th>Work Package</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Grid Workload Management</td>
<td>C.Vistoli/INFN</td>
</tr>
<tr>
<td>2</td>
<td>Grid Data Management</td>
<td>B.Segal/CERN</td>
</tr>
<tr>
<td>3</td>
<td>Grid Monitoring services</td>
<td>R.Middleton/PPARC</td>
</tr>
<tr>
<td>4</td>
<td>Fabric Management</td>
<td>T.Smith/CERN</td>
</tr>
<tr>
<td>5</td>
<td>Mass Storage Management</td>
<td>J.Gordon/PPARC</td>
</tr>
<tr>
<td>6</td>
<td>Integration Testbed</td>
<td>F.Etienne/CNRS</td>
</tr>
<tr>
<td>7</td>
<td>Network Services</td>
<td>C.Michau/CNRS</td>
</tr>
<tr>
<td>8</td>
<td>HEP Applications</td>
<td>F.Carminati/CERN</td>
</tr>
<tr>
<td>9</td>
<td>EO Science Applications</td>
<td>C.Michau/CNRS</td>
</tr>
<tr>
<td>10</td>
<td>Biology Applications</td>
<td>C.Michau/CNRS</td>
</tr>
<tr>
<td>11</td>
<td>Dissemination</td>
<td>G.Mascari/CNR</td>
</tr>
<tr>
<td>12</td>
<td>Project Management</td>
<td>F.Gagliardi/CERN</td>
</tr>
</tbody>
</table>
More realistically - a Grid Topology

- CERN – Tier 0
- Tier 1: FNAL
- Tier 2: Lab a, Uni b, Lab c, Uni n
- Department: α, β, γ
- Desktop

Connections:
- 622 Mbps between CERN and Tier 1 sites
- 2.5 Gbps between Tier 1 and Tier 2 sites
- 155 Mbps within Tier 2 sites
LHC Computing Review

- **Report issued on February 2001**
- **Recognition of**
  - Size of LHC computing: 240MCHF initially and every 3y
  - Understaffing of both IT and Offline core teams
  - Distributed hierarchical computing model (MONARC)
- **Recommendations**
  - GRID technology development
  - Common projects
  - Establishment of proper management structure
  - Continued support for common libraries and introduction of the CERN support for FLUKA & ROOT
  - Data challenges on common Prototype testbed
  - Improvement of the planning of the Off-line Projects
  - Drafting of software MOU
The LHC Computing Grid Project Structure

LHCC

Common Computing RRB

Reports

Reviews

Resource Matters

Project Overview Board

RTAG

Common

Computing

Matters

Other Computing Grid Projects

Other HEP Grid Projects

EU DataGrid Project

CERN

Other Labs

Project Execution Board

Requirements, Monitoring

Software and Computing Committee (SC2)

implementation teams
LHC GRID Computing Phase 1

- Technology development and data challenges
- Production prototype at CERN, MS&NMS 2001-2004
  - ~50% in complexity of one LHC experiment
- Training and technology opportunities
- Approved in the Council Meeting of 20-Sept-01
  - ~ 40 FTEs for three years (more pledged)
  - ~ 30 MCHF (>10 pledged from various sources)
- Critical level for “approval to start” reached
- Need still more funds for the CERN prototyping
  - Contributions in kind, EU-FP6, industry contributions
- Discussion with M & NM States, agencies, industries
  - Work out convention with each funding agency
The Problem - Summary

- **Scalability** → cost → management
  - Thousands of processors, thousands of disks, Petabytes of data, Terabits/second of I/O bandwidth, ....

- **Wide-area distribution**
  - WANs are and will be 1% of LANs
  - Distribute, replicate, cache, synchronize the data
  - Multiple ownership, policies, ....
  - Integration of this amorphous collection of Regional Centers ..
    - .. with some attempt at optimization

- **Adaptability**
  - We shall only know how analysis is done once the data arrives
Overview

- **LHC computing challenge**
  - MONARC, EU DG, LCG

- **The Grid metaphor**
  - Globus

- **Alice Grid Activities**
  - The Alice Offline Project
  - Remote job submission and monitoring
  - AliEn
  - Interface Root-Globus, Root-Alien, Root-...
  - Distributed analysis with proof

- **Summary**
The GRID metaphor

- Unlimited ubiquitous distributed computing
- Transparent access to multipetabyte distributed data bases
- Easy to plug in
- Hidden complexity of the infrastructure
- Analogy with the electrical power GRID

Five Emerging Models of Networked Computing from

- Distributed Computing
  - synchronous processing
- High-Throughput Computing
  - asynchronous processing
- On-Demand Computing
  - dynamic resources
- Data-Intensive Computing
  - databases
- Collaborative Computing
  - scientists
How to build a Grid Application?

- **Do It Yourself**
- **Legion = MetaOS on top of local OS**
  “Legion is a reflective object-based wide-area system designed from first principles to address the difficult technical challenges of using distributed resources (cycles, data, people, applications)”
  - [www.legion.virginia.edu](http://www.legion.virginia.edu)
- **Globus = Protocol/Toolkit approach**
  “The Globus Toolkit provides a set of standard services for authentication, resource location, resource allocation, configuration, communication, file access, fault detection, and executable management. These services can be incorporated into applications and/or programming tools in a "mix-and-match" fashion to provide access to needed capabilities”
  - [www.globus.org](http://www.globus.org)
Globus Approach

- **Focus on architecture issues**
  - Propose set of core services as basic infrastructure
  - Use to construct high-level, domain-specific solutions

- **Design principles**
  - Keep participation cost low
  - Enable local control
  - Support for adaptation

![Diagram showing applications and core Globus services connected to local OS]
“Data Grid” Architecture Elements

**APPLICATIONS**

<table>
<thead>
<tr>
<th>Task mgmt (Condor-G)</th>
<th>Data request management</th>
<th>Caching</th>
<th>Virtual Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliable replication</td>
<td>Replica selection</td>
<td>Attribute-based lookup</td>
<td></td>
</tr>
<tr>
<td>Location cataloging</td>
<td>Metadata cataloging</td>
<td></td>
<td>Virtual Data cataloging</td>
</tr>
</tbody>
</table>

Enquiry (LDAP)
Access (GRAM)

CPU
CPU resource manager

Storage
Storage resource manager

…”
The Grid from a Services View

Applications
- Chemistry
- Cosmology
- Environment
- Biology
- High Energy Physics

Application Toolkits
- Distributed Computing Toolkit
- Data-Intensive Applications Toolkit
- Collaborative Applications Toolkit
- Remote Visualization Applications Toolkit
- Problem Solving Applications Toolkit
- Remote Instrumentation Applications Toolkit

Grid Services (Middleware)
Resource-independent and application-independent services
- authentication, authorization, resource location, resource allocation, events, accounting, remote data access, information, policy, fault detection

Grid Fabric (Resources)
Resource-specific implementations of basic services
- E.g., Transport protocols, name servers, differentiated services, CPU schedulers, public key infrastructure, site accounting, directory service, OS bypass
The Globus Advantage

- **Single sign-on for all resources**
  - No need for user to keep track of accounts and passwords at multiple sites
  - No plaintext passwords
- **Uniform interface to various local scheduling mechanisms**
  - PBS, Condor, LSF, NQE, LoadLeveler, fork, etc.
  - No need to learn and remember obscure command sequences at different sites
- **Support for staging, etc.**
Overview

- **LHC computing challenge**
  - MONARC, EU DG, LCG

- **The Grid metaphor**
  - Globus

→ **Alice Grid Activities**
  - The Alice Offline Project
  - Remote job submission and monitoring
  - AliEn
  - Interface Root-Globus, Root-Alien, Root-...
  - Distributed analysis with proof

- **Summary**
ALICE Collaboration

ALICE Collaboration statistics

937 members (63% from CERN MS)
77 Institutions, 28 Countries
ALICE Computing

- **1800 SI95, 2.8PB tapes/y, >1 PB disk**
  - Same order of magnitude than ATLAS or CMS
- **Major strategic decisions already taken**
  - Move to C++ completed beginning 1999
    - TDRs all produced with the new framework
    - PPR production being done with AliEn/AliRoot
  - Adoption of the ROOT framework
  - Tightly knit Off-line team – single development line
  - Physics performance and computing in a single team
  - Ambitious DC program on the LHC prototype
  - ALICE DAQ/Computing integration realised during data challenges in collaboration with ALICE DAQ team and IT division
Alice Offline Framework

- > 50 active users participate in the development of AliRoot from the different detector groups
  - 71% of the code developed outside, 29% by the core Offline team

- AliRoot framework
  - C++: 400kLOC + 225kLOC (generated) + macros: 77kLOC
  - FORTRAN: 13kLOC (ALICE) + 914kLOC (external packages)
  - Maintained on Linux (any version!), HP-UX, DEC Unix, Solaris

- Two packages to install (ROOT+AliRoot)
  - Less that 1 second to link (thanks to 37 shared libs)
  - 1-click-away install: download and make (non-recursive makefile)

- AliEn
  - 50kLOC of PERL5 (ALICE) + 1MLOC of PERL5 (external packages)

- Installed on more than 30 sites by physicists
## Offline Milestones

### Physics Challenge
- Determine readiness of the offline framework for data processing
- Verify the computing model

### Data Challenge
- Verify integration of Offline and DAQ
- Assess technologies and models
- Determine readiness of DAQ system

<table>
<thead>
<tr>
<th>PDC Period (milestone) (Size)</th>
<th>Physics Objective</th>
<th>ADC Milestone (Data to MSS)</th>
<th>Offline milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>06/01-06/02 5%</td>
<td>First test of the complete chain from simulation to reconstruction for the PPR Simple analysis. ROOT digits.</td>
<td>10/2002 200TB</td>
<td>200 MB/s</td>
</tr>
<tr>
<td>01/04-06/04 10%</td>
<td>Complete chain used for trigger studies. Prototype of the analysis tools. Comparison with parameterised MonteCarlo. Simulated raw data.</td>
<td>5/2003 300TB</td>
<td>300 MB/s</td>
</tr>
<tr>
<td>01/06-06/06 20%</td>
<td>Test of the final system for reconstruction and analysis.</td>
<td>5/2004 450TB</td>
<td>450 MB/s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5/2006 1250TB</td>
<td>1250 MB/s</td>
</tr>
</tbody>
</table>
ALICE/GRID Strategy

- **Test / Use Globus Tools**
- **Physics Performance Report (Step 1)**
  - Involving progressively more sites (Catania, CERN, GSI/FZK, Lyon, OSC, NIKHEF, NERSC, Torino)
- **AliKit + AliEn (+ Globus & MRTG & …)**
  - Don't wait for EU-DG1
- **Integration with DataGRID (Testbed 1)**
  - AliKit into EU-DG1: straightforward
  - Evaluate EU-DG1: Services
  - AliEn into EU-DG1: interaction with middleware WP's
- **ALICE Data Challenge IV**
- **Development of distributed parallel analysis tools (PROOF)**
ALICE Physics Performance Report

- Evaluation of acceptance, efficiency, resolution for signals
- **Step 1**: Simulation of ~10,000 central Pb-Pb events
  - Hits: 2 GB/event = 20 TB
  - Summable Digits: 1 GB/event = 10 TB
  - AliEn (Globus Authentication + Data Catalog) 1 user (Production Manager)
- **Step 2**: Signal Superposition + Reconstruction 100,000 events
  - Reconstructed Events = 10 TB x n_{signals}
  - ROOT + Data Manager for access to distributed input: n_{signals} users (Production Managers)
- **Step 3**: Event Analysis: Analysis Object Data = 1 TB x n_{signals}
  - PROOF + Data Manager + Workload Manager for chaotic access to distributed analysis input: 200 users
Remote Job Submission and Monitoring

- Web interface to login on "The Grid"
- Remote Job Submission and Monitoring
- Web interface for job submission
- CPU Load
- Disk space
- Network availability

Bookkeeping system

Only at Lyon

LDAP server for ALICE (only in Italy)
ALICE GRID: Monitoring

Monitor of:
- CPU load
- Disk availability
- Network load

Client 07 CPU load

System: ali07
Maintainer: Mario.Sitts@infn.it

The statistics were last updated: Mon Aug 13 09:59:10 2001

'Daily' graph (5 Minute Average)

'Weekly' Graph (30 Minute Average)

Max Cpu: 41 U/sec, Average Cpu: 3 U/sec, Current Cpu: 0 U/sec.

'Weekly' graph (30 Minute Average)
### ALICE GRID August Production

- **Bookkeeping included:**
  - Check production & keep track of the event parameters for selection
  - Central Server (Catania) + mirrors (Lyon, Bologna)
  - MySQL based, ORACLE interface available

### ALICE Physics Performance Report: Simulation Status

<table>
<thead>
<tr>
<th>RUN EVENT</th>
<th>SUBMITTER</th>
<th>WORKER</th>
<th>STATUS</th>
<th>START</th>
<th>STOP</th>
<th>FTP</th>
<th>ROOT CONFIG OUT</th>
<th>ERR</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>336</td>
<td>grid1.ct.infn.it</td>
<td>alifatur9.ct.infn.it</td>
<td>DONE</td>
<td>2001-08-26 13:48:03</td>
<td>2001-08-26 13:49:18</td>
<td>ERROR</td>
<td>disk</td>
<td>show</td>
</tr>
<tr>
<td>1</td>
<td>337</td>
<td>grid1.ct.infn.it</td>
<td>alifatur9.ct.infn.it</td>
<td>DONE</td>
<td>2001-08-26 14:08:14</td>
<td>2001-08-26 14:09:13</td>
<td>ERROR</td>
<td>disk</td>
<td>show</td>
</tr>
<tr>
<td>1</td>
<td>338</td>
<td>hxbatch028.cern.ch</td>
<td>hxbatch028.cern.ch</td>
<td>DONE</td>
<td>2001-08-27 10:47:54</td>
<td>2001-08-27 10:49:40</td>
<td>ERROR</td>
<td>disk</td>
<td>show</td>
</tr>
<tr>
<td>1</td>
<td>339</td>
<td>hxbatch106.cern.ch</td>
<td>hxbatch106.cern.ch</td>
<td>DONE</td>
<td>2001-08-27 15:42:02</td>
<td>2001-08-27 15:43:27</td>
<td>ERROR</td>
<td>disk</td>
<td>show</td>
</tr>
<tr>
<td>1</td>
<td>340</td>
<td>alifatur01.to.infn.it</td>
<td>alifatur01.to.infn.it</td>
<td>DONE</td>
<td>2001-08-28 12:31:05</td>
<td>2001-08-28 12:31:19</td>
<td>ERROR</td>
<td>disk</td>
<td>show</td>
</tr>
<tr>
<td>1</td>
<td>341</td>
<td>alifatur01.to.infn.it</td>
<td>alifatur01.to.infn.it</td>
<td>DONE</td>
<td>2001-08-28 12:33:58</td>
<td>2001-08-28 12:34:55</td>
<td>ERROR</td>
<td>disk</td>
<td>show</td>
</tr>
<tr>
<td>1</td>
<td>342</td>
<td>grid1.ct.infn.it</td>
<td>alifatur9.ct.infn.it</td>
<td>DONE</td>
<td>2001-08-28 12:40:28</td>
<td>2001-08-28 13:41:58</td>
<td>ERROR</td>
<td>disk</td>
<td>show</td>
</tr>
<tr>
<td>5</td>
<td>18</td>
<td>juno.mikheev.nl</td>
<td></td>
<td>ERROR</td>
<td>2001-08-28 14:15:28</td>
<td>2001-08-28 14:15:42</td>
<td>not done</td>
<td>file</td>
<td>show</td>
</tr>
<tr>
<td>1</td>
<td>343</td>
<td>testbed001.cern.ch</td>
<td></td>
<td>QUEUED</td>
<td>2001-08-28 14:47:38</td>
<td></td>
<td>not done</td>
<td>-</td>
<td>show</td>
</tr>
<tr>
<td>1</td>
<td>344</td>
<td>testbed001.cern.ch</td>
<td></td>
<td>QUEUED</td>
<td>2001-08-28 14:47:38</td>
<td></td>
<td>not done</td>
<td>-</td>
<td>show</td>
</tr>
</tbody>
</table>
ALICE in EU DG

- First HEP event produced on the DataGRID testbed released Dec 10, '01
- ALICE provides HEP coordination
Distributed file catalogue, simulating a file structure.

Each directory is a table in a SQL database, abstract interface to DB's

Ownership and privileges like in a UNIX system.

User shell interface, C/C++ API

Completely transparent access to mass storage systems.

Replication & caching

Automatic synchronization among all the databases.

150kL perl5, 95% from the WEB

SOAP for client/server and API implementation

Secure authentication
ALICE production distributed Environment
- Entirely ALICE developed
- File Catalogue as a global file system on a RDB
- TAG Catalogue, as extension
- Secure Authentication
  - Interface to Globus available
- Central Queue Manager ("pull" vs "push" model)
  - Interface to EDG Resource Broker available
- Monitoring infrastructure
  - The CORE GRID functionality
- Automatic software installation with AliKit
- http://alien.cern.ch
HI PPR Production Summary

- Entirely AliEn based!
- >5500 jobs validated (~24h each)
- 13 clusters actively used in production, 9 remote sites
- GSI, Karlsruhe, Dubna, Nantes, Budapest, Bari, Zagreb, Birmingham are joining
- Hijing param 8000/4000/2000 dN/dy
- Hijing centrality bins
- HIJING + Jet/Photon Trigger
  - Minimum Transverse Momentum: 25, 50, 75, 100GeV
Other productions

- **EMCAL production**
  - Design and optimisation of the proposed EM calorimeter
  - Entirely AliEn based
  - Decided, implemented and realised in two months
  - 2000 jobs, 4000h CPU

- **pp production august 2001**
  - ~ 10000 events generated at CERN in CASTOR
  - Transport + TPC digitization

- **pp production october 2001**
  - Vertex sampling in the diamond
  - > 10000 events - 80 GB in 3 days in Torino transferred to CERN with AliEn in ~20 hours (band-width saturated)
  - Test and tune tracking
The Grid:
ROOT with the help of Grid Services:
- Finds convenient places for it to be run
- Organises efficient access to your data
  - Caching, migration, replication of data and/or code
- Deals with authentication to the different sites that you will be using
- Interfaces to local site resource allocation mechanisms, policies
- Compiles/runs your scripts
- Monitors progress
- Recovers from problems

If there is scope for parallelism, it can also decompose your work into convenient execution units based on the available resources, data distribution, ...

This can work only, if there is a close cooperation of ROOT and Grid services!
Physicist

Show me the equation of state of quark gluon plasma

Root script

Use Case:(chaotic) Analysis
DataGrid & ROOT

TagDB

Selection
Parameters
Procedure

CPU

PROOF

Local

Remote

RDB

Proc.C

DB1

Proc.C

DB2

Proc.C

DB3

Proc.C

DB4

Proc.C

DB5

Proc.C

DB6

Bring the KB to the PB and not the PB to the KB
Parallel Script Execution

Local PC

$ root

root [0] .x ana.C

root [1] gROOT->Proof("remote")

root [2] gProof->Exec(".x ana.C")

proof = master server
proof = slave server

Remote PROOF Cluster

#proof.conf
slave node1
slave node2
slave node3
slave node4

node1
node2
node3
node4

*.root

TNetFile

TFile

TFile

TFile
Running a PROOF Job

gROOT->Proof();
TDSet *treeset = new TDSet("TTree", "AOD");
treeset->Add("lfn:/alien.cern.ch/alice/prod2002/file1");
. . .
gProof->Process(treeset, "myselector.C");


gROOT->Proof();
TDSet *objset = new TDSet("MyEvent", "+", "/events");
objset->Add("lfn:/alien.cern.ch/alice/prod2002/file1");
. . .
objset->Add(set2003);
gProof->Process(objset, "myselector.C");
class TGrid : public TObject { 
public:
    virtual Int_t AddFile(const char *lfn, const char *pfn) = 0;
    virtual Int_t DeleteFile(const char *lfn) = 0;
    virtual TGridResult *GetPhysicalFileNames(const char *lfn) = 0;
    virtual Int_t AddAttribute(const char *lfn,
                               const char *attrname,
                               const char *attrval) = 0;
    virtual Int_t DeleteAttribute(const char *lfn,
                                   const char *attrname) = 0;
    virtual TGridResult *GetAttributes(const char *lfn) = 0;
    virtual void Close(Option_t *option="") = 0;
    virtual const char *GetInfo() = 0;
    const char *GetGrid() const { return fGrid; }
    const char *GetHost() const { return fHost; }
    Int_t GetPort() const { return fPort; }
    virtual TGridResult *Query(const char *query) = 0;
    static TGrid *Connect(const char *grid, const char *uid = 0,
                           const char *pw = 0);
ClassDef(TGrid,0) // ABC defining interface to GRID services
};
Running PROOF Using AliEn

TGrid *alien = TGrid::Connect("alien://alien.cern.ch", rdm);
TGridResult *res;
res = alien->Query("lfn:///alice/simulation/2001-04/V0.6*.root");
TDSet *treeset = new TDSet("TTree", "AOD");
treeset->Add(res);

gROOT->Proof();
gProof->Process(treeset, "myselector.C");

// plot/write objects produced in myselector.C
...
**DataGrid & ROOT**

- **Selection parameters** → **TAG DB** → **Root** → **Grid RB**
  - LFN #hits
- **Grid log & monitor**
- **Grid replica manager**
- **Grid autenticate**
- **PROOF loop** → **Spawn PROOF tasks** → **best places**
  - output LFNs
- **Update Root RDB** → **Send results back**
  - Send results back
- **Grid replica catalog**
First Steps:

- Use GSI to provide a single sign-on via "grid-id" (prototype ready for proof)
- `globus-job-run` to start remote demons
  - we need jobmanager-fork or another way to launch interactive processes
- TLdap to access MDS
  - API: OpenLDAP
  - Information model: Globus, DataGrid
    - Mapping of events to files
    - Mapping of files to replica
    - Ressource broker
    - ...

MDS Approach

- **Based on LDAP**
  - Lightweight Directory Access Protocol v3 (LDAPv3)
  - Standard data model
  - Standard query protocol
- **Globus specific schema**
  - Host-centric representation
- **Globus specific tools**
  - GRIS, GIIS
  - Data discovery, publication, …
Overview

- **LHC computing challenge**
  - MONARC, EU DG, LCG

- **The Grid metaphor**
  - Globus

- **Alice Grid Activities**
  - The Alice Offline Project
  - Remote job submission and monitoring
  - AliEn
  - Interface Root-Globus, Root-Alien, Root-...
  - Distributed analysis with proof

➔ **Summary**
The Grid from a Services View

Applications

Chemistry
Cosmology
Environment
Biology
High Energy Physics

Application Toolkits

Distributed Computing Toolkit
Data-Intensive Applications Toolkit
Collaborative Applications Toolkit
Remote Visualization Applications Toolkit
Problem Solving Applications Toolkit
Remote Instrumentation Applications Toolkit

Grid Services (Middleware)

Resource-independent and application-independent services
authentication, authorization, resource location, resource allocation, events, accounting,
remote data access, information, policy, fault detection

Grid Fabric (Resources)

Resource-specific implementations of basic services
E.g., Transport protocols, name servers, differentiated services, CPU schedulers, public key
infrastructure, site accounting, directory service, OS bypass
# AliEn as a meta-GRID

<table>
<thead>
<tr>
<th>AliEn User Interface</th>
<th>iVDGL stack</th>
<th>AliEn stack</th>
<th>EDG stack</th>
</tr>
</thead>
</table>

![Map of North America](image1)  
![Map of Europe](image2)
Physicist executes a root (C++) script to analyse events

**root:**
- finds the name of the file(s) with the events
- locates the file (and replica)
- decides based on
  - network speed and load, CPUs/disks available and file usage pattern,
  - whether -to fetch the file -to use only parts (via rootd)
  - or -to execute the script at the remote side (via proof).