Grid Computing

Philippe Charpentier
PH Department
CERN, Geneva

Outline

- Disclaimer:
  - These lectures are not meant at teaching you how to compute on the Grid!
  - I hope it will give you a flavor on what Grid Computing is about…
  - We shall see how far we can go today…
- Today
  - Why Grid Computing, what is it?
  - Data Management system
- Tomorrow
  - Workload Management system
  - Using the Grid (production, user analysis)
  - What is being achieved?
  - Outlook

The challenge of LHC Computing

- Large amounts of data for each experiment
  - Typically 2-300 MB/s or real data (1 MB events)
    - $5\ 10^6$ s gives 1.5 PB/year
- Large processing needs
  - Reconstruction: 10-20 s/evt
    - Requires 2-4000 processors permanently
    - For reprocessing (in 2 months): 4 times more
  - Simulation: up to 15 mn /evt!!!
    - Requires few 10,000 processors
  - Analysis: fast, but many users and many events!
    - Similar to reconstruction in resource needs

An ideal world

- Huge Computing Center
  - Several 100,000 processors
  - Tens of PB of fast storage (disks)
  - Highly performing network connections
    - From the experimental site
    - From the collaborating institutes

- … and the associated risks
  - If it fails, all activities are stopped…

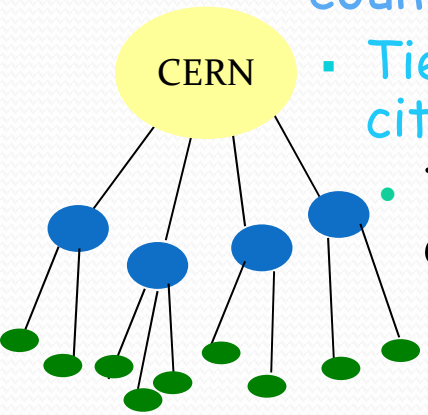
An alternative…

- Distributed Computing
  - Several large computing centers
    - Distributed over the world
  - Multiple replicas of datasets
    - Avoids single point of failure
  - High performance network connections
    - For data replication
  - Transparent job submission
    - Hide the complexity of the system to users

… The Computing Grid …

The Monarc model

- Hierarchical model of sites (dated 1999)
  - Tier0: where the real data are produced (CERN)
    - Tier1s : large Computing Centers, at the level of a country or a region
      - Tier2s : smaller Computing Centers, in large institutes or cities
        - Tier3s: computer clusters, at the level of a physics department
          - Tier4s : the physicist's desk/lap-top



- The data flows along the hierarchy
  - Hierarchy of processing steps
    - Reconstruction, global analysis, group analysis, final analysis

The LHC Computing Grid

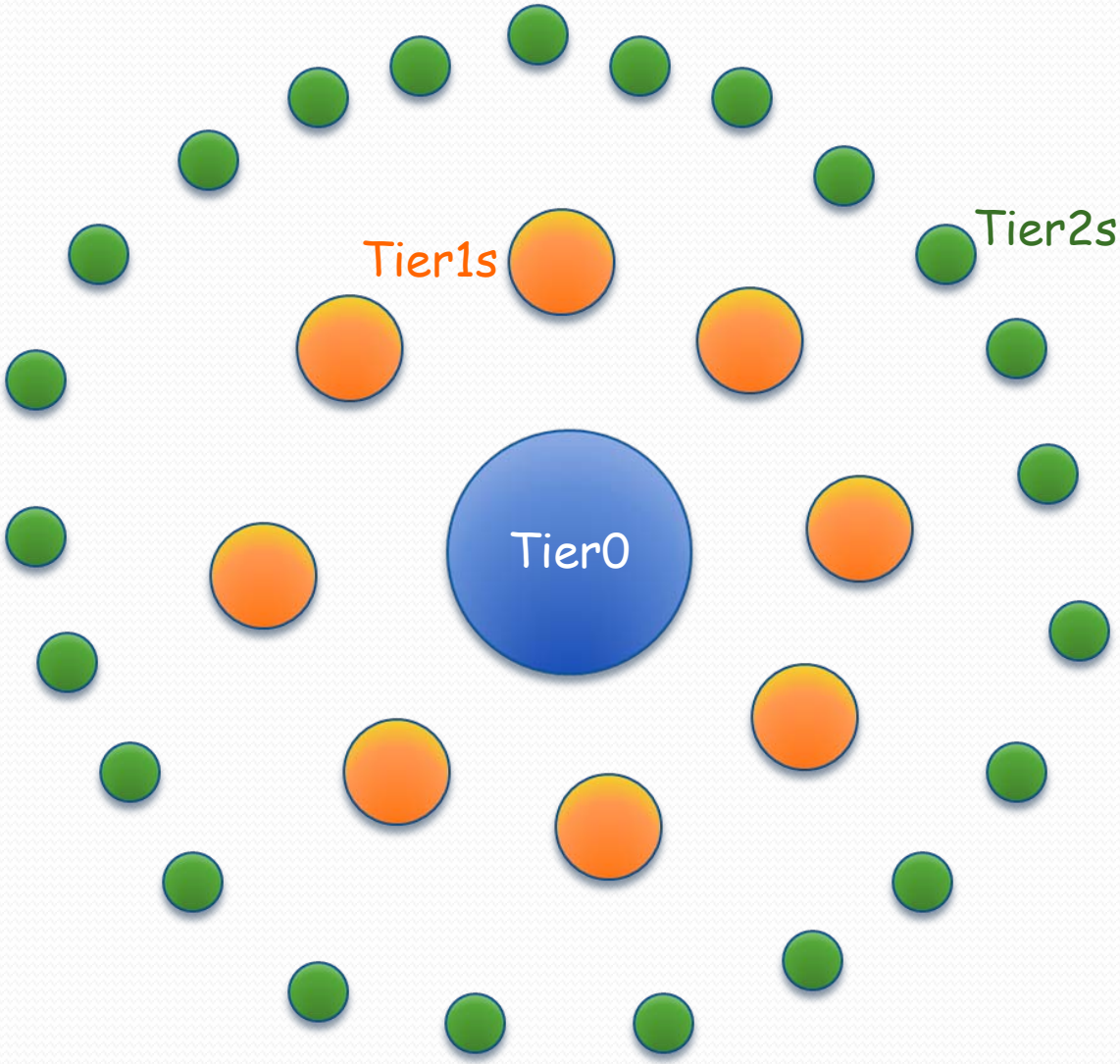
- Uses the same terminology as MONARC
  - … but "Tiers" reflect the level of service
  - It is an implementation of the Grid
  - Tier0: CERN
    - Tier1s: large capacity of mass storage ("tapes"), (multi-)national level, high bandwidth to Tier0
      - Tier2s: no mass storage, lower quality of service, regional level
        - Tier3s and Tier4s: not considered as part of the Grid (regular computers or clusters)
  - Not necessarily hierarchical dependency
    - Although Tier1s are usually supporting Tier2s
      - … but not necessarily strict link for data flow

The Computing Models

- Definition of functionality
  - Real data collection and recording
    - Real data replication for security
  - Real data initial reconstruction
    - Quasi real-time
  - Real data re-reconstruction
    - After months or years
  - Monte-Carlo simulation
    - No input data required, highly CPU demanding
  - Analysis
    - Large dataset analysis
    - Final (a.k.a.) end-user analysis
- The mapping to Tiers may differ
  - It is the experiment's choice…

Computing Models and the Grid

The Grid



Computing Models and the Grid

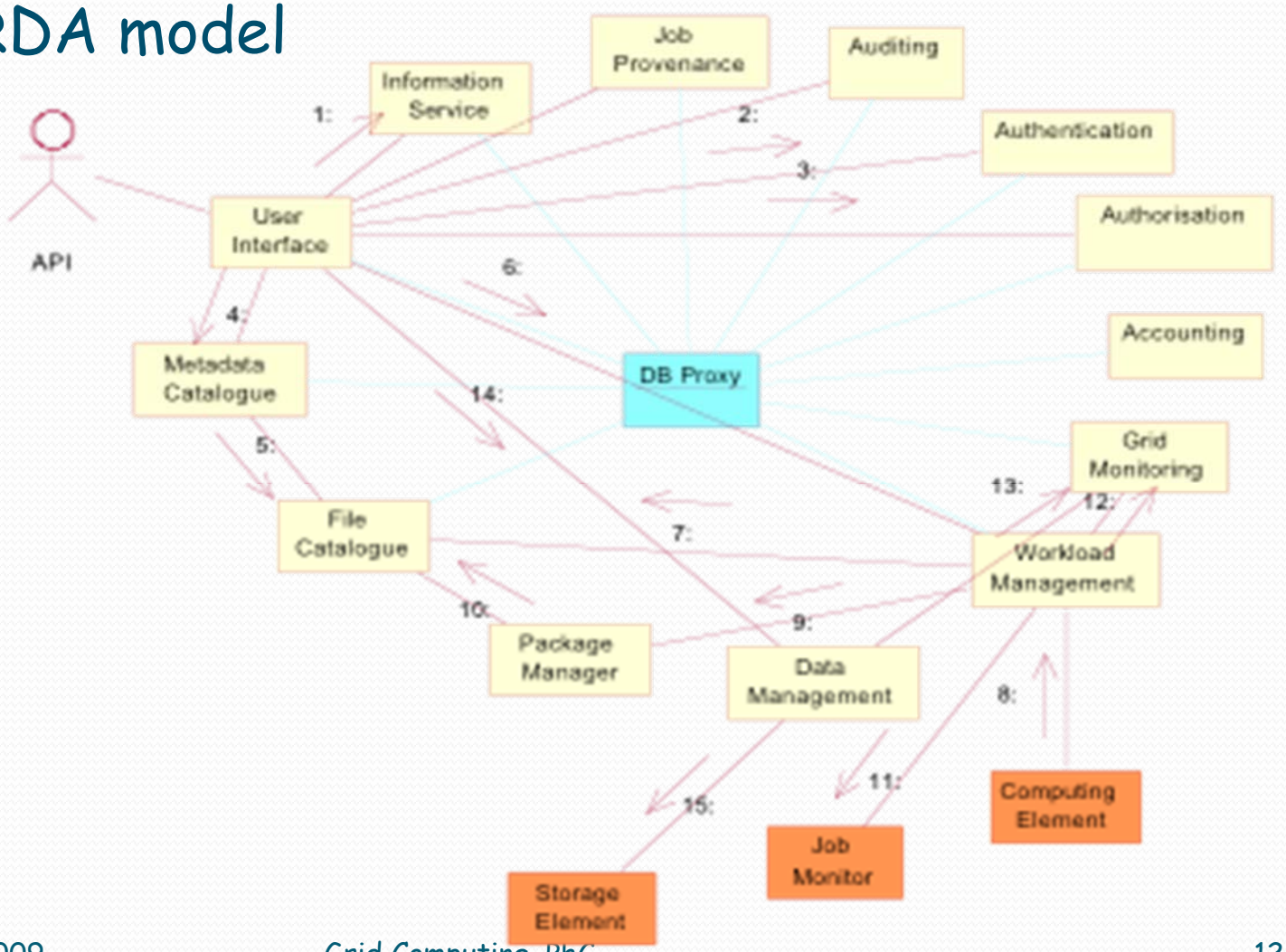
	Tier0	Tier1	Tier2
	Central data recording	Data mass storage	
ALICE	First event reconstruction Calibration Analysis	Event re-reconstruction	AOD storage Simulation Analysis
ATLAS			
CMS			
LHCb	Event reconstruction Analysis		Simulation

The Grid abstraction

- Large worldwide Computer Center
- Top requirements
  - Storage and data transfer (Data Management)
  - Job submission (Workflow Management)
  - Global infrastructure:
    - Computing Centers' and Users' support
      - Including problem reporting and solving
    - High bandwidth networks
- The Grid functionality is decomposed in terms of "services"

Possible breakdown of Grid services

- The ARDA model



Grid security (in a nutshell!)

- Important to be able to identify and authorise users
  - Possibly to enable/disable certain actions
- Using X509 certificates
  - The Grid passport, delivered by a certification authority
- For using the Grid, create short-lived "proxies"
  - Same information as the certificate
  - … but only valid for the time of the action
- Possibility to add "group" and "role" to a proxy
  - Using the VOMS extensions
  - Allows a same person to wear different hats (e.g. normal user or production manager)
- Your certificate is your passport, you should sign whenever you use it, don't give it away!
  - Less danger if a proxy is stolen (short lived)

Organisation of the Grid

- Infrastructure
  - Computing Centers, networks
    - In Europe, EGEE (Enabling Grids for E-SciencE)
      - In European Nordic countries, NorduGrid
    - In US: OSG (Open Science Grid)
  - Each infrastructure may deploy its own software (hereafter called middleware)
- Activity Grid initiatives
  - For LHC Computing, WLCG (Worldwide LHC Computing Grid)
    - Federates the infrastructures for a community
- Virtual Organisations (VOs)
  - Each set of users within a community
  - E.g. in WLCG: the LHC experiments
    - … but there are many more, in HEP and in e-Science

Grid infrastructures partners

Open Science Grid

Data Management

Data Management services

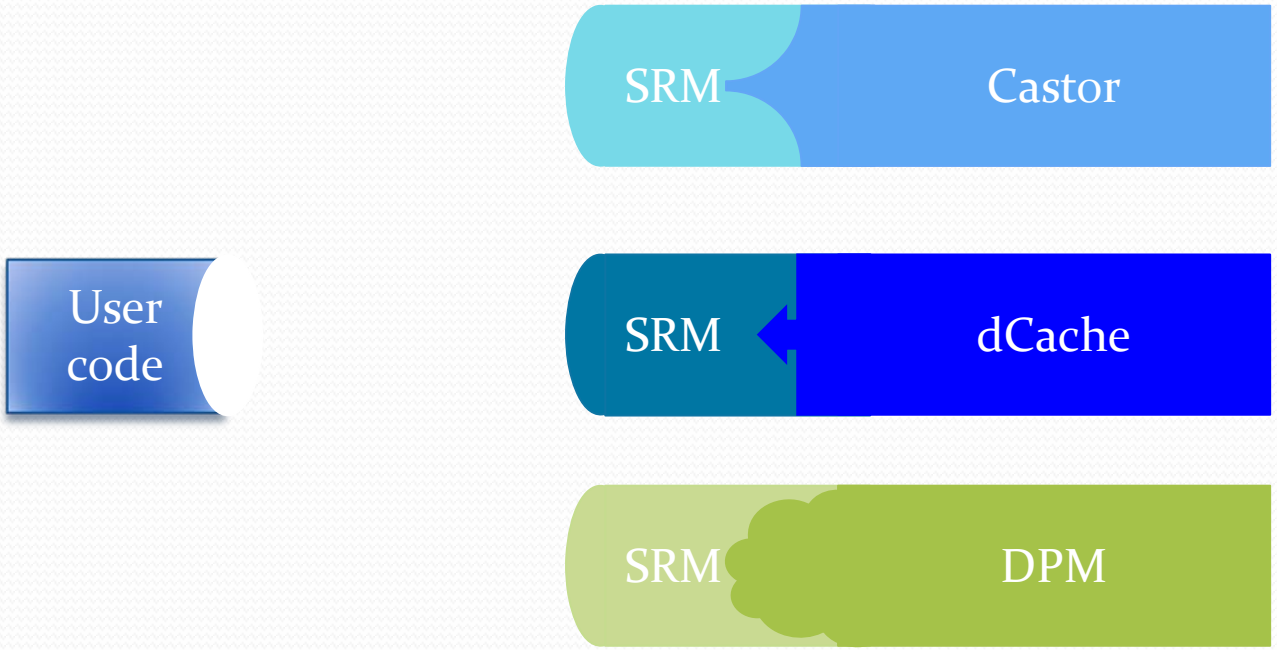
- Storage Element
  - Disk servers
  - Mass-storage (tape robots) for Tier0/1
- Catalogs
  - File catalog
    - Data location (SE) and file name (URL)
  - Metadata catalog
    - More experiment-specific way for users to get information on datasets
- File transfer services
- Data access libraries

Storage Elements (1)

- Sites have freedom of technology
- 4 basic storage technologies in WLCG
  - Castor
    - Disk and mass storage integrated
    - Used at CERN, RAL, CNAF and AGSC (Taiwan)
  - dCache
    - Disk caching system, can use multiple mass storage systems
    - Used at most other Tier1s and some Tier2s (no MSS)
  - DPM
    - Disk Pool Manager: can use any file system back-end
    - Used at most Tier2
  - STorM
    - Disk storage, can be coupled to e.g. Castor for MSS
    - Used at CNAF Tier1 and some Italian Tier2s

Storage Elements (2)

- 4 types of SEs
  - Need for an abstraction layer
  - Users use a single interface
- Storage Resource Manager (SRM)



SRM functionality (1)

- Provides a unified interface to SEs
- Enables partitioning of storage into "spaces"
  - Classified according to Service Classes
    - Defines accessibility and latency
    - Mainly 3 classes:
      - T1D0: mass storage but no permanent disk copy
      - T1D1: mass storage with permanent disk copy
      - T0D1: no mass storage, but permanent disk copy
      - Can be extended (e.g. T2, D2)
    - Spaces define a partitioning of the disk space
  - Access Control Lists for actions on files per space
    - E.g. disallow normal users to stage files from tape

SRM functionality (2)

- Standard format of Storage URL
  - srm://<end-point>:<port>/<base-path>/<path>
  - <base-path> may be VO-dependent
    - In principle independent on service class
- Returns transport UTL (tURL) for different access protocols
  - Data transfer: gsiftp
  - POSIX-like access: file, rfio, dcap, gsidcap, xrootd
  - tURL used by applications for opening or transferring files
- Allows directory and file management
  - srm_mkdir, srm_ls, srm_rm…

File access and operations

- Applications use tURLs, obtained from SURL through SRM
- Standard File Access Layer
  - gfal: POSIX-like library, SRM-enabled
  - xrootd: alternative to SRM+gfal for file access and transfer, but can be used as access protocol only too.
- High level tool:
  - lcg_utils
    - Based on gfal
    - See later functionality

File catalogs (1)

- Files may have more than one copy (replica)
  - Useful to uniquely represent files, independent on their location
  - Logical File Name
    - Structure like a path, but location-independent
      - Human readable
      - E.g. /grid/lhcb/MC/2009/DST/1234/1234_543_1.dst
  - Global Unique Identifier (GUID)
    - 128-bit binary indentifier
      - Not readable
      - Used for files cross-references (fixed size)
      - Attributed at file creation, unique by construction

File catalogs (2)

- Associates GUID, LFN and list of replicas (SURL)
  - Queries mainly by GUID or LFN
  - Contains file metadata (as a file system)
    - Size, date of creation
  - Returns list of replicas
- In LHC experiments 3 types
  - LCG File Catalog (LFC): used by ATLAS and LHCb
    - Database, hierarchical, generic
  - AliEn: developed and used by ALICE
  - Trivial File Catalog: used by CMS
    - Just construction rules, and list of sites per file (DB)

High level DM tools

- Integrating File Catalog, SRM and transfer
  - E.g. lcg_utils

```
[lxplus227] ~ > lfc-ls -l /grid/lhcb/user/p/phicharp/TestDVAssociators.root
-rwxr-sr-x   1 19503    2703  430 Feb 16 16:36
/grid/lhcb/user/p/phicharp/TestDVAssociators.root

[lxplus227] ~ > lcg-lr lfn:/grid/lhcb/user/p/phicharp/TestDVAssociators.root
srm://srm-lhcb.cern.ch/castor/cern.ch/grid/lhcb/user/p/phicharp/TestDVAssociators.root

[lxplus227] ~ > lcg-gt srm://srm-
lhcb.cern.ch/castor/cern.ch/grid/lhcb/user/p/phicharp/TestDVAssociators.root rfio
rfio://castorlhcb.cern.ch:9002/?svcClass=lhcbuser&castorVersion=2&path=/castor/cern.ch/grid/lhc
b/user/p/phicharp/TestDVAssociators.root

[lxplus227] ~ > lcg-gt srm://srm-
lhcb.cern.ch/castor/cern.ch/grid/lhcb/user/p/phicharp/TestDVAssociators.root gsiftp
gsiftp://lxfsrj0501.cern.ch:2811//castor/cern.ch/grid/lhcb/user/p/phicharp/TestDVAssociators.ro
ot

[lxplus227] ~ > lcg-cp srm://srm-
lhcb.cern.ch/castor/cern.ch/grid/lhcb/user/p/phicharp/TestDVAssociators.root file:test.root
```

From LFN to reading a file

```
[lxplus311] ~ $ lcg-lr lfn:/grid/lhcb/MC/MC09/DST/00005071/0000/00005071_00000004_1.dst
srm://ccsrm.in2p3.fr/pnfs/in2p3.fr/data/lhcb/MC/MC09/DST/00005071/0000/00005071_00000004_1.dst
srm://srm-
lhcb.cern.ch/castor/cern.ch/grid/lhcb/MC/MC09/DST/00005071/0000/00005071_00000004_1.dst
srm://srmlhcb.pic.es/pnfs/pic.es/data/lhcb/MC/MC09/DST/00005071/0000/00005071_00000004_1.dst
[lxplus311] ~ $ lcg-gt srm://srm-
lhcb.cern.ch/castor/cern.ch/grid/lhcb/MC/MC09/DST/00005071/0000/00005071_00000004_1.dst root
castor://castorlhcb.cern.ch:9002/?svcClass=lhcbdata&castorVersion=2&path=/castor/cern.ch/grid/l
hcb/MC/MC09/DST/00005071/0000/00005071_00000004_1.dst
[lxplus311] ~ $ root -b
  ****************************************
  *                                      *
  *       W E L C O M E  to  R O O T      *
  *                                      *
  *   Version  5.22/00c      27 June 2009  *
  *                                      *
  *  You are welcome to visit our Web site *
  *          http://root.cern.ch         *
  *                                      *
  ****************************************

ROOT 5.22/00c (branches/v5-22-00-patches@29026, Jun 29 2009, 14:34:00 on linuxx8664gcc)

CINT/ROOT C/C++ Interpreter version 5.16.29, Jan 08, 2008
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
TFile::Open("castor://castorlhcb.cern.ch:9002/?svcClass=lhcbdata&castorVersion=2&path=/castor/c
ern.ch/grid/lhcb/MC/MC09/DST/00005071/0000/00005071_00000004_1.dst")
```

Transferring large datasets: FTS

- Needs scheduling of network bandwidth
- Using dedicated physical paths
  - E.g. LHC Optical Network (LHCOPN)
- Transfers handled by a service (FTS)
  - An FTS service handles a channel (from A to B)
  - Manages share between VOs
  - Optimises bandwidth
  - Takes care of transfer failures and retries
- Yet to come: file placement service
  - Move this dataset to this SE (i.e. 100% of it)
  - Implemented within the VO frameworks

Other Data Management services

- Metadata catalog
  - Contains VO-specific metadata for files/datasets
  - Allows users to identify datasets
  - Example:
    - Type of data (e.g. AOD)
    - Date of data taking, type of simulated data…
    - Type of processing (e.g. reconstruction April 2010)
  - Too specific to be generic

VO frameworks

- Need to elaborate on Grid middleware
- Each LHC experiment has its own Data Management System
  - File Catalog
    - E.g. LFC, AliEn-FC…
  - Dataset definitions
    - Sets of files always grouped together
  - Metadata catalogs
    - User queries, returns datasets of LFNs
  - Access and transfer utilities
    - Use gridftp, lcg_cp or FTS

End of lecture 1

Workload Management System

The Great Picture

- Prepare your software, distribute it on the Grid
- Submit jobs, monitor them and retrieve "results"
- Not care where it ran, provided it does its job!

Workload Management System

The Great Picture

- Prepare your software, distribute it on the Grid
  - The Worker Node (WN) that will run the job must have this capability
- Submit jobs, monitor them and retrieve "results"
  - If the job needs data, it must be accessible
- Not care where it ran, provided it does its job!
  - … but care where it ran if it failed…

The components

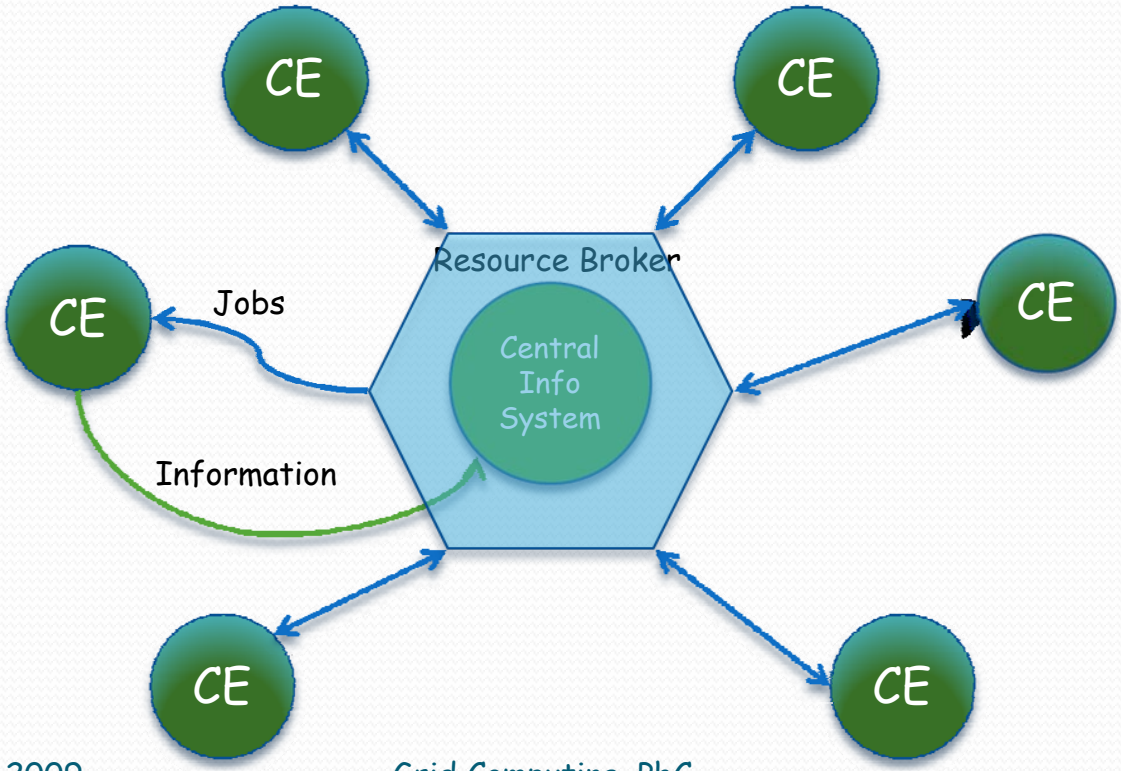
- Local batch system
  - Under the site's responsibility
  - Must ensure proper efficiency of CPU resources
  - Allows to apply shares between communities
  - Provides monitoring and accounting of resources
- Central job submission service
  - To which users send their tasks (a.k.a. jobs)
- Job description
  - Which resources are needed (Operating system, input data, CPU resources etc…)
  - Standard Job Description Language (JDL)
    - Rather cryptic…
  - Additional data can be added in a "sandbox"
    - Input and output sandboxes (limited in size)

Local Batch System

- Several flavors on the market
  - LSF, PBS, BQS, Condor etc…
  - Not practical for the Grid…
    - … define an abstraction…
    - **Computing Element (CE)**
      - Knows the details of all implementation
      - Provides a standard interface for:
        - Job submission
        - Job monitoring
        - Information on the CE
      - CE may depend on the infrastructure…
        - Need for interoperability

Central job submission

- Ideally:
  - A "Big Brother" that knows permanently the state of the whole system!
  - It decides on this knowledge what is the most "cost-effective" site to target, knowing the job's requirements



Resource Brokering

- Caveats with the ideal picture:
  - One entity can hardly support the load (few 100,000 job simultaneously)
    - Several instances
    - Each instance works independently
      - Changes the "great picture"
  - Information is not instantaneously available
    - Typical refresh rate of few minutes
      - The picture has time to change quite a lot…
  - How to evaluate the cost of a choice?
    - Estimated return time: how long shall I wait to get the job done?

The real life of WM (1)

- Several instances of RB per community
  - Typically handling up to 10,000 jobs each
- Dynamic information system (BDII)
  - Regularly updated by CEs
    - Typically 2-3 minutes
  - Used by RBs to make a decision based on the state of the universe
- Complex RB-CE interaction
  - What if no resources are indeed available?
    - E.g. due to out-of-date information
    - Job refused, sent back to RB, retry somewhere else…

The real life of WM (2)

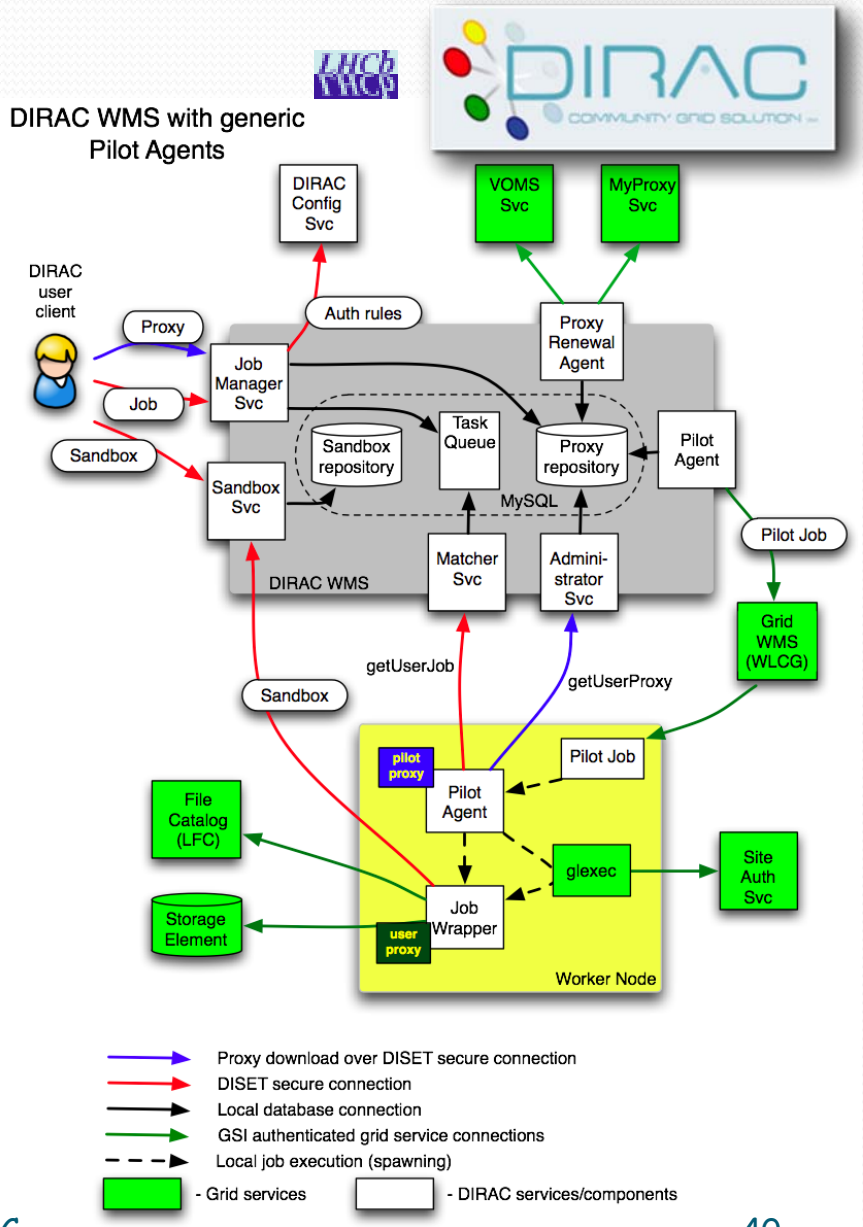
- Which OS, which software is deployed?
  - Publish "software tags", used in job matching
  - CEs must be homogeneous, as much as possible
- Specify max CPU time limit
  - Define units and apply CPU rating
    - Still in flux…
- User identity
  - User's ability to use resources should be checked
    - Users can be "banned' in case of misbehavior
  - Users are mapped to standard UNIX (uid,gid)
    - These are transient, but most sites make a one-to-one assignment (pool accounts, like <VO>_001, _002…)
- Applying priorities between users or Vos
  - Can be done using "fair shares"
    - Needs site intervention for changes

An alternative to the central broker

- Pilot jobs
  - Small jobs that do not carry the actual job to be executed
  - Knows about where to ask for work to be done (central task queues)
    - If nothing found, can just quit gracefully
  - Pilots are all identical for a given set of queues and framework
  - Implements a "pull" paradigm, opposed to a "push" paradigm (RB)
  - Pioneered by ALICE and LHCb (2003)
    - Now used by ATLAS and partly by CMS

The DIRAC example

- Late job binding
  - A job is fetched only when all is OK
    - Software and data present
    - Site capabilities matching
- Apply priorities centrally
  - No site intervention
  - Users and production jobs mixed
- Federating Grids
  - Same pilots can be submitted to multiple Grids
- Similar systems:
  - AliEn (ALICE)
  - PANDA (ATLAS)
  - CMSGlideins (CMS)



The Grid Experience (1)

- Basic infrastructure is there
  - Several flavors (EGEE, OSG, NorduGrid…)
- Basic middleware is there
  - Storage-ware (from HEP): Castor, dCache, DPM, StoRM, SRM
  - DM middleware: gfal, lcg_utils, FTS
  - WM middleware: gLite-WMS, LCG-CE (obsolescent, soon to be replaced by CREAM from gLite)
- Good coordination structures
  - EGEE, OSG
  - WLCG for LHC experiments (and HEP)
  - Complex relationships and dependencies…

The Grid Experience (2)

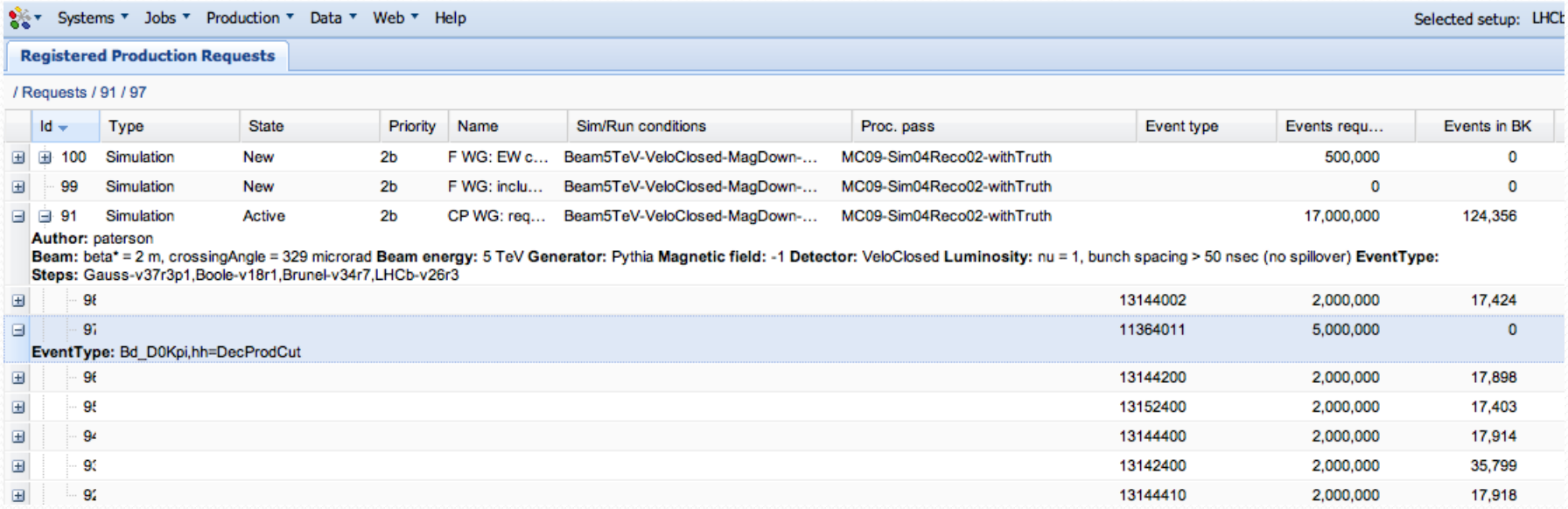
- Large VOs must elaborate complex frameworks
  - Integrating middleware
    - Data Management: File-catalog, SE, Metadata-catalog, FTS
  - Applying specific policies and addressing complex use cases
    - Dataset publication, replication
  - Develop alternative WM solutions
    - Pilot jobs frameworks, still using WM middleware for pilots submission
- Production and user analysis frameworks
  - Based on the Grid frameworks

Production systems

- Organised data processing
  - Decided at the VO-level
  - Very much CPU-intensive, repetitive tasks with different starting conditions
- Examples:
  - Real data reconstruction
  - Large samples Monte-Carlo simulation
  - Group analysis (data reduction)
- Workflow:
  - Define tasks to be performed
  - Create tasks
  - Generate jobs, submit and monitor them
  - Collect and publish results
- Each VO has its own dedicated and tailored production system (possibly systems)

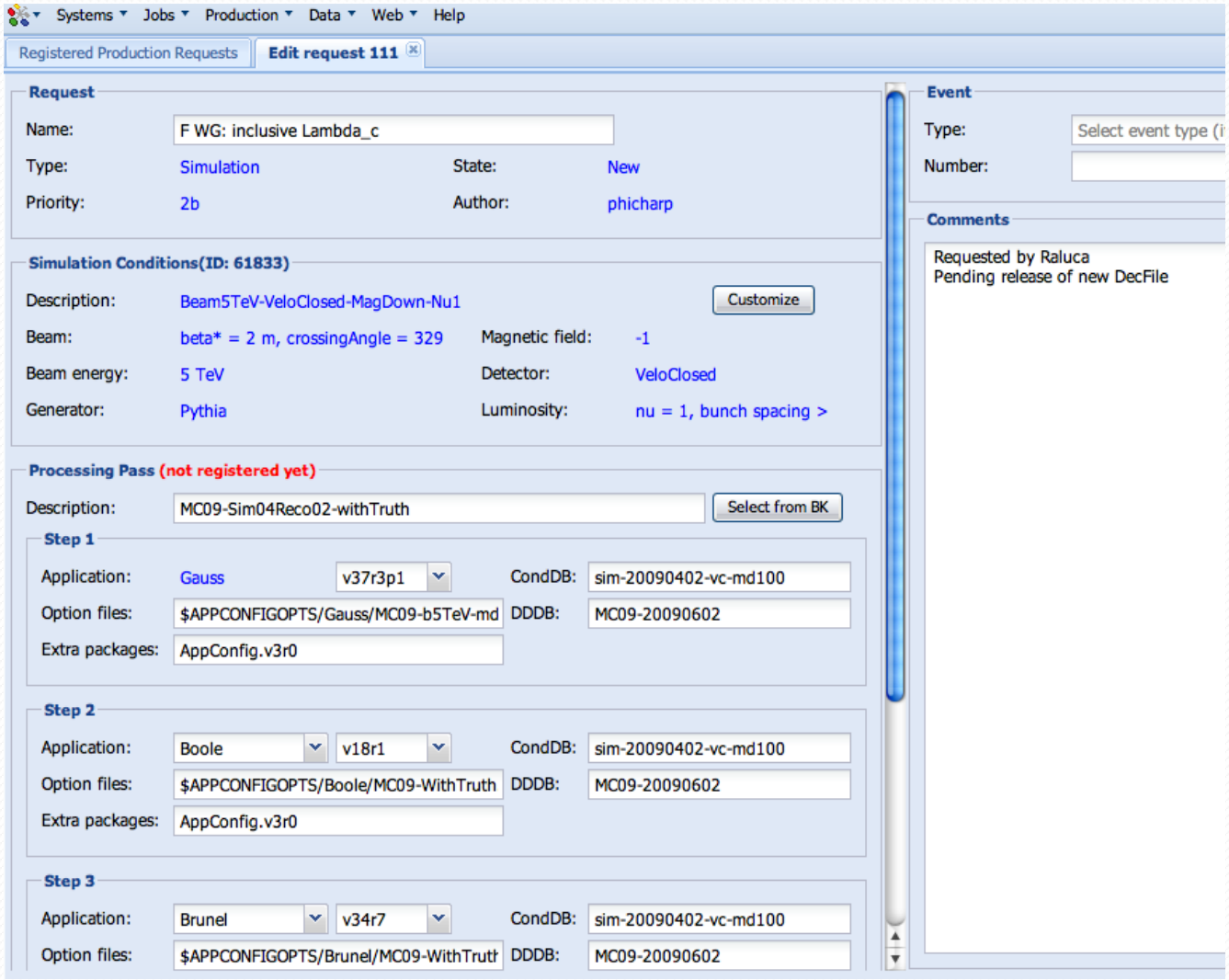
Example of production system

- From LHCb/DIRAC system
  - Users define their needs
  - Validation
    - Physics Planning Group: also sets priorities
    - Technical validation (application managers)
  - Production is launched
    - Progress is monitored



Id	Type	State	Priority	Name	Sim/Run conditions	Proc. pass	Event type	Events requ…	Events in BK
100	Simulation	New	2b	F WG: EW c…	Beam5TeV-VeloClosed-MagDown-…	MC09-Sim04Reco02-withTruth		500,000	0
99	Simulation	New	2b	F WG: inclu…	Beam5TeV-VeloClosed-MagDown-…	MC09-Sim04Reco02-withTruth		0	0
91	Simulation	Active	2b	CP WG: req…	Beam5TeV-VeloClosed-MagDown-…	MC09-Sim04Reco02-withTruth		17,000,000	124,356
Author: paterson **Beam:** beta* = 2 m, crossingAngle = 329 microrad **Beam energy:** 5 TeV **Generator:** Pythia **Magnetic field:** -1 **Detector:** VeloClosed **Luminosity:** nu = 1, bunch spacing > 50 nsec (no spillover) **EventType:** **Steps:** Gauss-v37r3p1,Boole-v18r1,Brunel-v34r7,LHCb-v26r3									
9							13144002	2,000,000	17,424
9							11364011	5,000,000	0
EventType: Bd_D0Kpi,hh=DecProdCut									
9							13144200	2,000,000	17,898
9							13152400	2,000,000	17,403
9							13144400	2,000,000	17,914
9							13142400	2,000,000	35,799
9							13144410	2,000,000	17,918

Production request example



Specificity of production

- Well organised job submission
  - Agreed by the collaboration
- Data placement following the Computing Model
  - Jobs upload their results
  - Datasets are then distributed to other sites
    - Transfer requests (associated to the production)
    - Dataset subscription (from sites)
    - User requests
- Also takes care of dataset archiving and deletion
  - Reduce number of replicas of old datasets
    - When unused, delete them

User Analysis support

- Much less organised!...
- Jobs may vary in resource requirements
  - SW development: rapid cycle, short jobs
  - Testing: longer, but still restricted dataset
  - Running on full dataset
- Output also varies
  - Histograms, printed output, Ntuple
    - Order kB to few MB
  - Event data output
    - Reduced dataset or elaborated data structure
    - Order GB

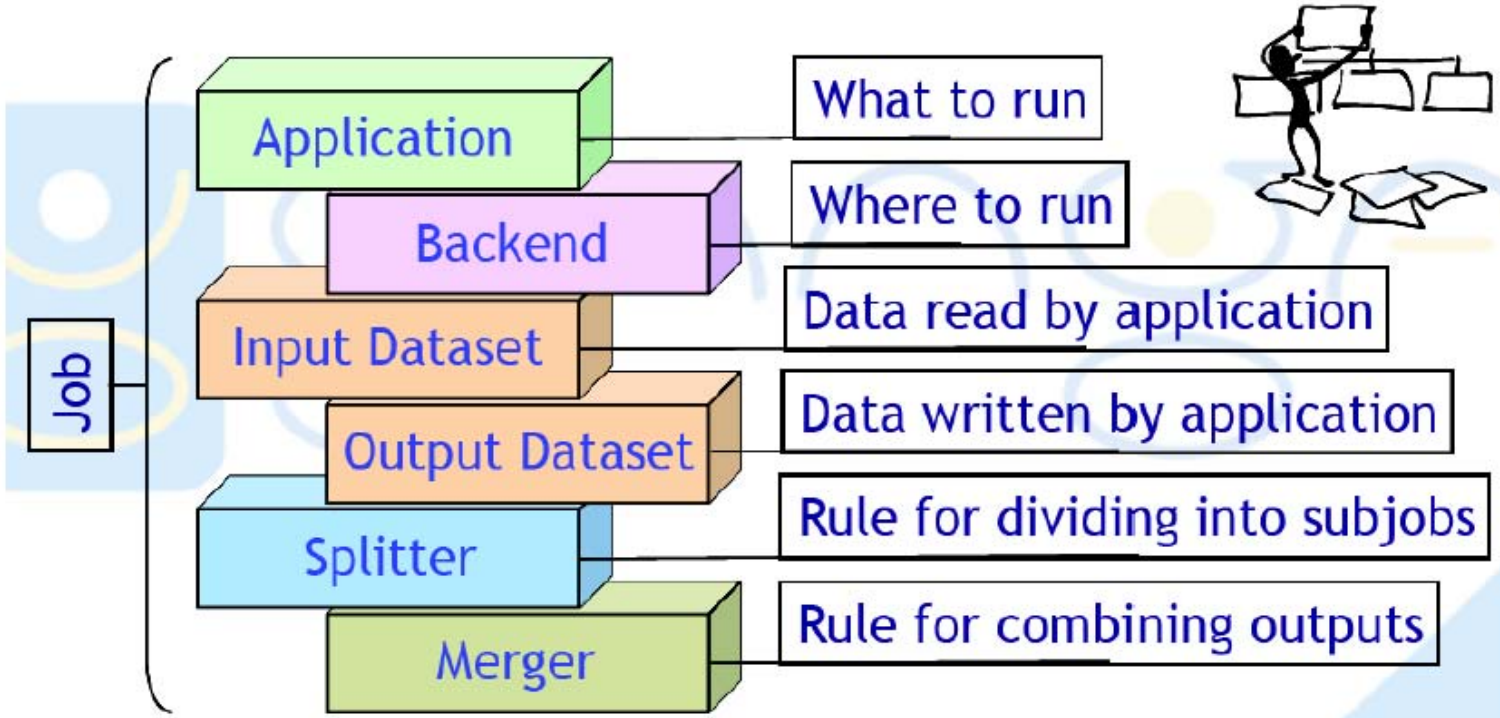
User Analysis

- For large datasets, users benefit from using the Grid
  - More potential resources
  - Automatic output data storage and registration
- For development and testing, still local running
- "End analysis"
  - From Ntuple, histograms
  - Making final plots
  - Outside the Grid
    - Desk/lap-top
    - Small clusters (Tier3)

Example of ganga

- Developed jointly by ATLAS and LHCb
- Allows users to configure their jobs, then run interactively, locally or on the Grid with just minor modifications
- Job repository: keeps history of jobs
- Job monitoring
- Hides complexity of Grid frameworks
  - Direct gLite-WMS, Panda (ATLAS), DIRAC (LHCb)

The Ganga Job Object



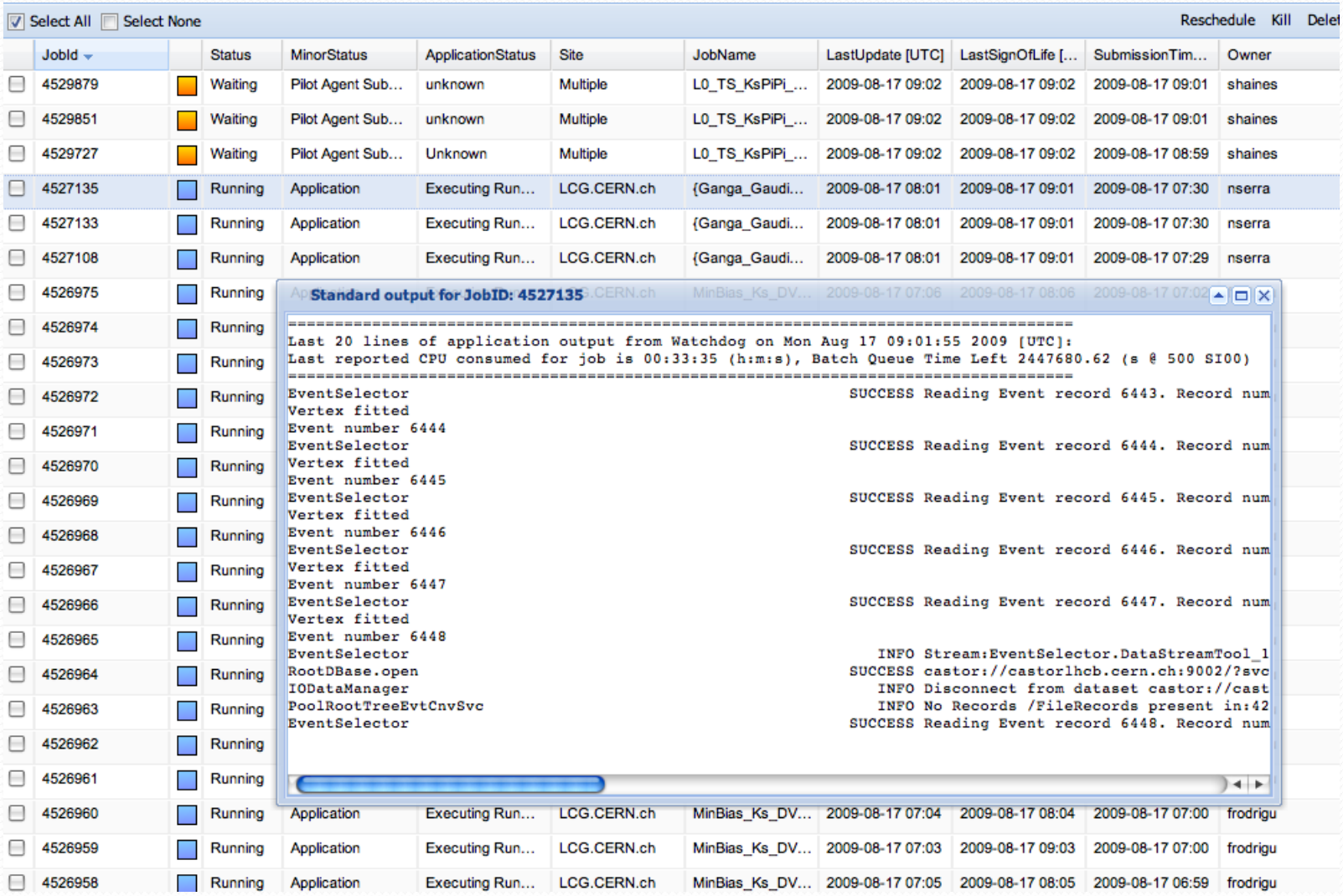
Job Create/Submit

```
[In:] j = Job()
[In:] j.application = Executable(exe='/bin/hostname')
[In:] j.name = "MyTest"
[In:] print j
[In:] j.submit()

# wait until job is completed and look
# at the output directory

[In:] j.status
[In:] j.peek()
[In:] j.peek('stdout')

# The syntax for peek is very flexible see
[In:] help(j.peek)
```

More on Job Submit

```
# once a job is submitted you cannot modify
# it. if you want to submit a new job you
# should create a new job object

[In:] j2 = j.copy()
[In:] j2.backend = Batch()#Default is LSF at CERN
[In:] j2.submit()


# if you have GRID certificate you can try

[In:] j3 = j.copy()
[In:] j3.backend = Dirac()
[In:] j3.submit()

# print jobs to see all your jobs
```

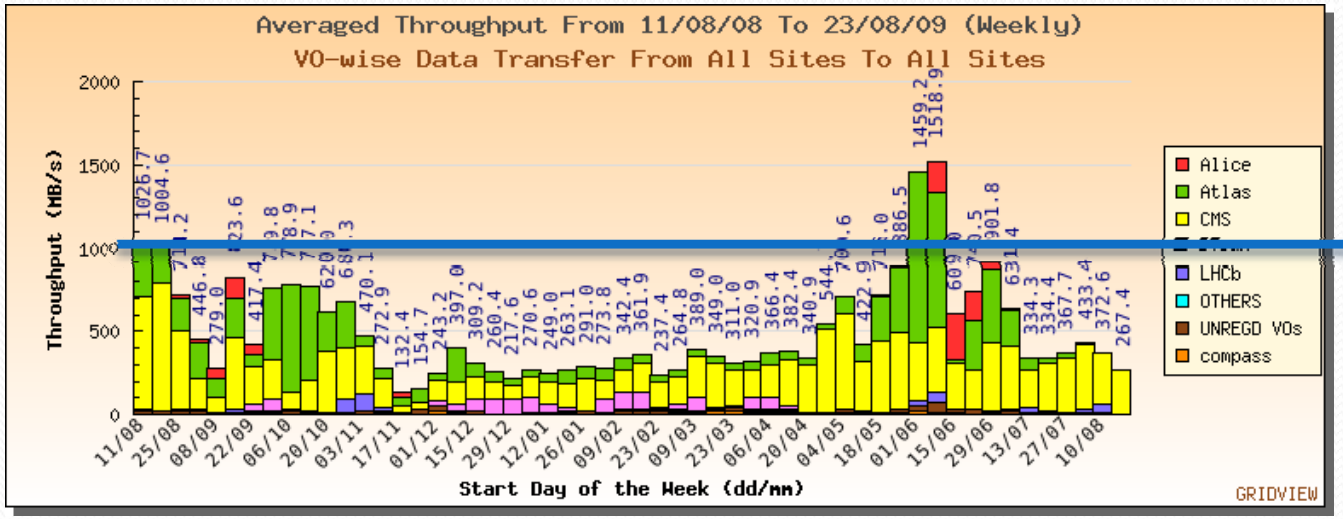
Job monitoring



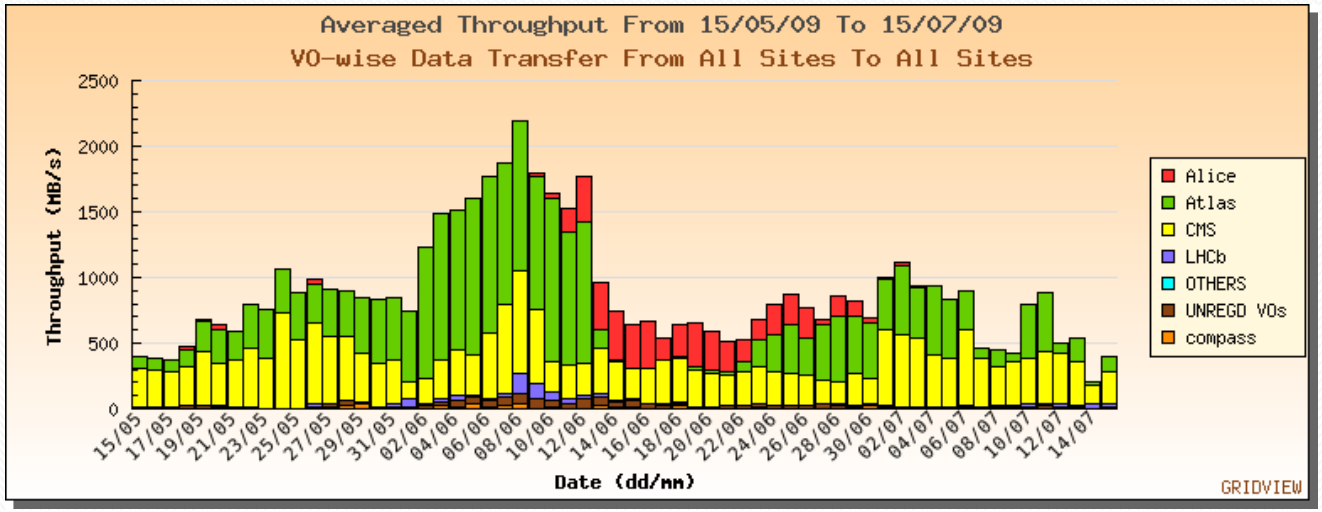
Does this all work?

- Many "challenges"
  - For years now, in order to improve permanently…
  - Data transfers, heavy load with jobs etc…
- Now the system is "in production"
  - Still needs testing of performance
  - Must be robust when fist data comes
  - Sites and experiments must learn
    - It will still take time to fully understand all operational issues
    - What happens when things go wrong?
- Many metrics used to assess performances
  - Global transfer rates
  - Individual sites' performance

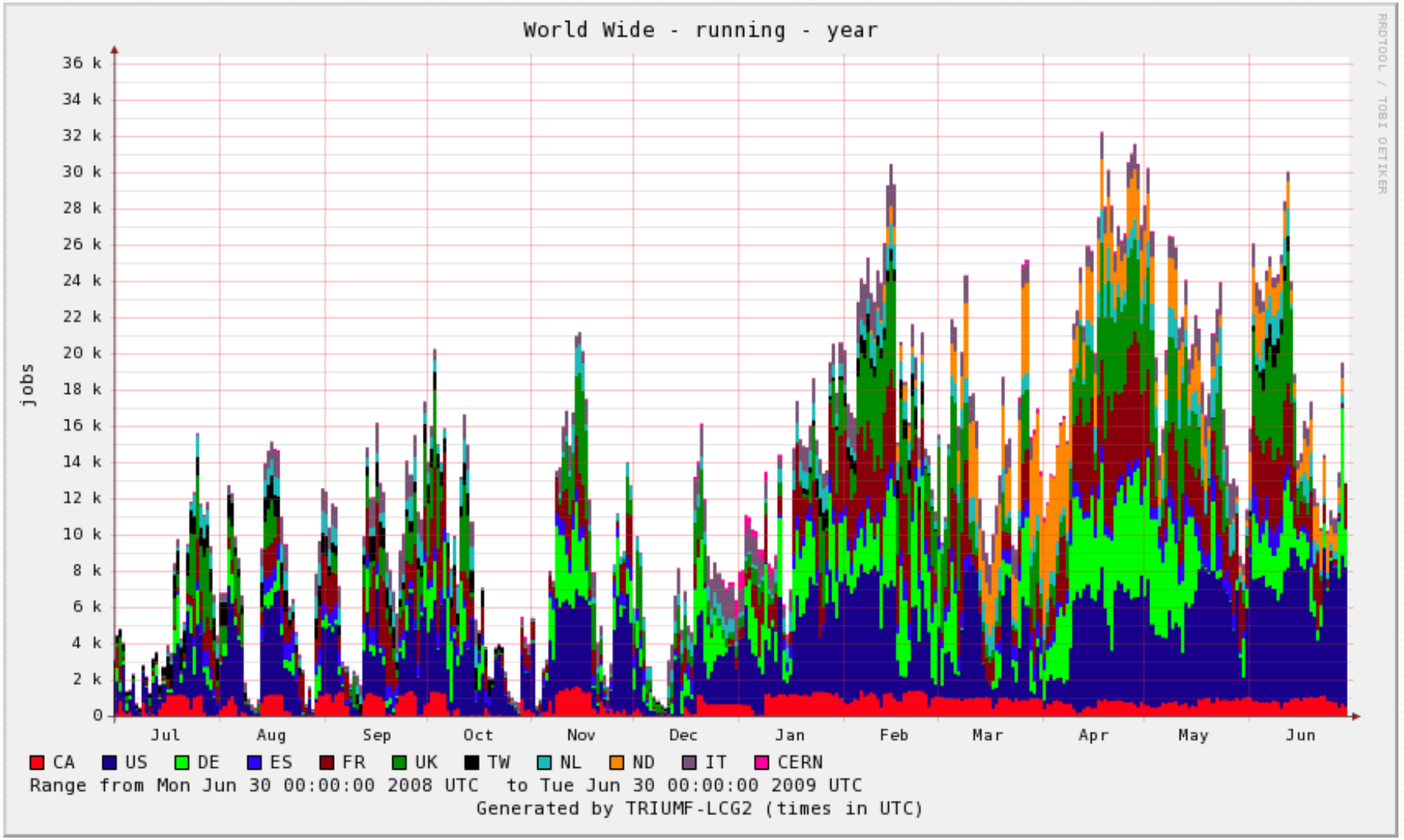
Data transfers



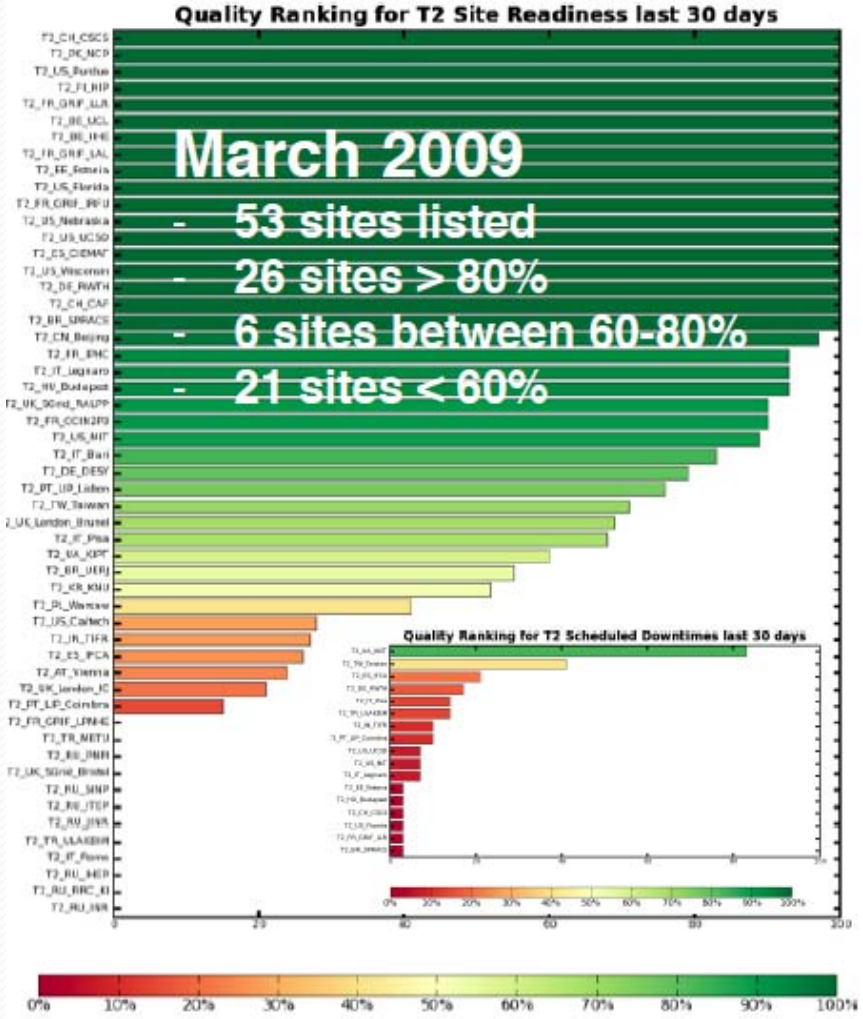
1 GB/s

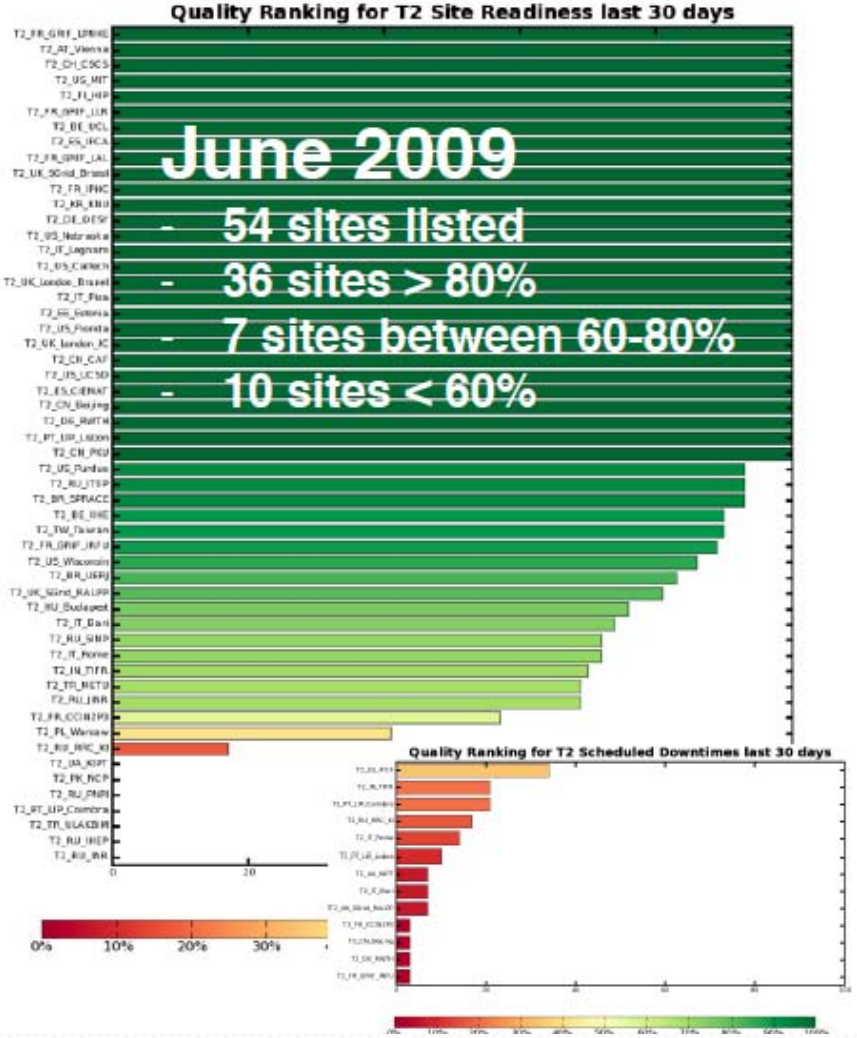


Jobs running on the Grid (ATLAS)



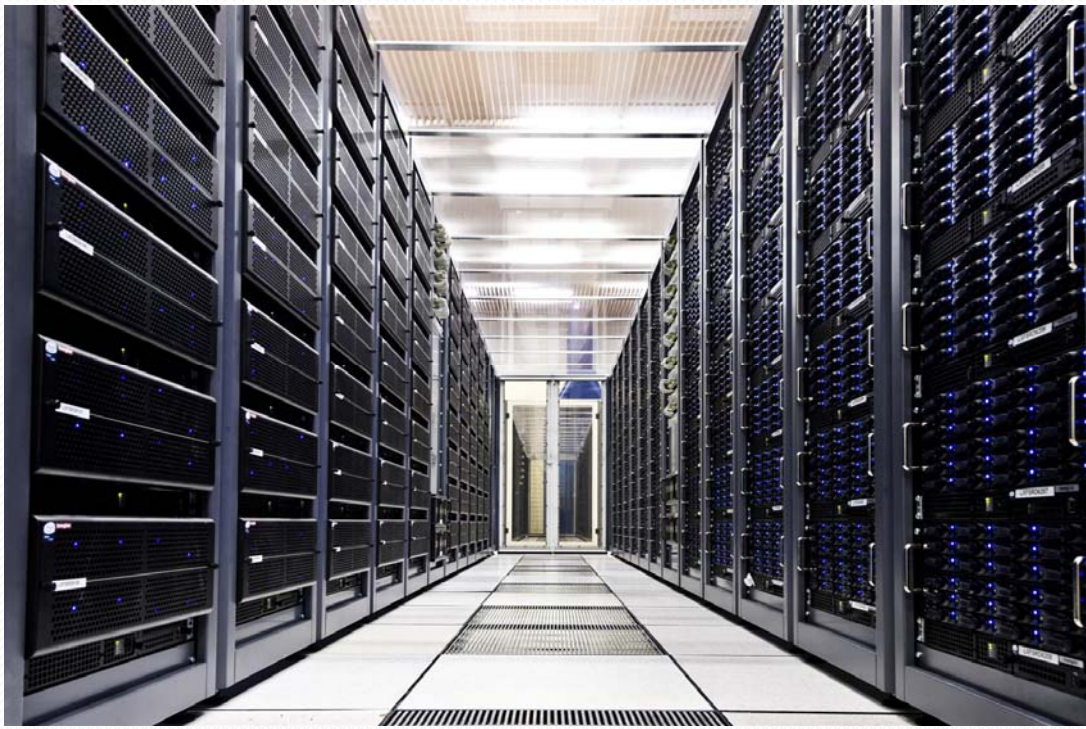
Site readiness T2: substantial improvement






eGee
Enabling Grids
for E-sciencE
UKI-SCOTGRID-ECDF
Edinburgh ECDF
Computing Elements:
ce.glite.ecdf.ed.ac.uk [6 / 1]
no Resource Brokers
mw05.ecdf.ed.ac.uk [9 / 13]
Graph(s) of the load for the Computing Elements at this site over the last week
Imperial College London
GridPP
UK Computing for Particle Physics
Google

Storage tape robot



CPU farm (CERN)

Outlook… rather than conclusions

- Grid Computing is a must for LHC experiments
  - The largest Computing Center in the world!
  - Data-centric (PB of data per year)
    - Unlike cloud computing (Amazon, Google..)
- Grid Computing requires huge efforts
  - From the sites, to be part of an ensemble
  - From the user communities (expts)
    - Flexibility, constant changes in services
- Nothing can be changed now
  - Concentrate on stability, stability, stability, stability, …

However…

Parallelisation

- All machines are now multi-core and soon many-core
  - Already 16 cores, soon 128 in a single machine
  - How to best use these architectures?
    - Adapt applications for using multiple cores
      - Saves a lot of memory in running the same application
    - Change way of allocating resources
  - Anyway mandatory for desktop usage
    - Why not for batch / Grid processing?

Virtualisation

- Always problems with customisation of environment on WNs
  - Operating system
  - Middleware, libraries, user software…
- Create "virtual machines" to be submitted to the Grid
  - Needs trust between users and resource providers (sites)
  - This is already the trend for cloud computing
- Workshop on parallelisation and virtualisation
  - http://indico.cern.ch/conferenceDisplay.py?confId=56353