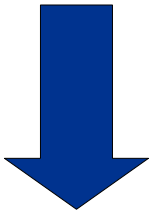
Experience of the running HEP experiments in using the EGEE/ LCG infrastructure

Simone Pagan Griso (University and INFN Padova)
Christoph Wissing (DESY)
Hartmut Stadie (University of Hamburg)
Vinicio Duic (University and INFN Trieste)

2nd EGEE User Forum,
10 May 2007 - Manchester, United Kingdom

Outline

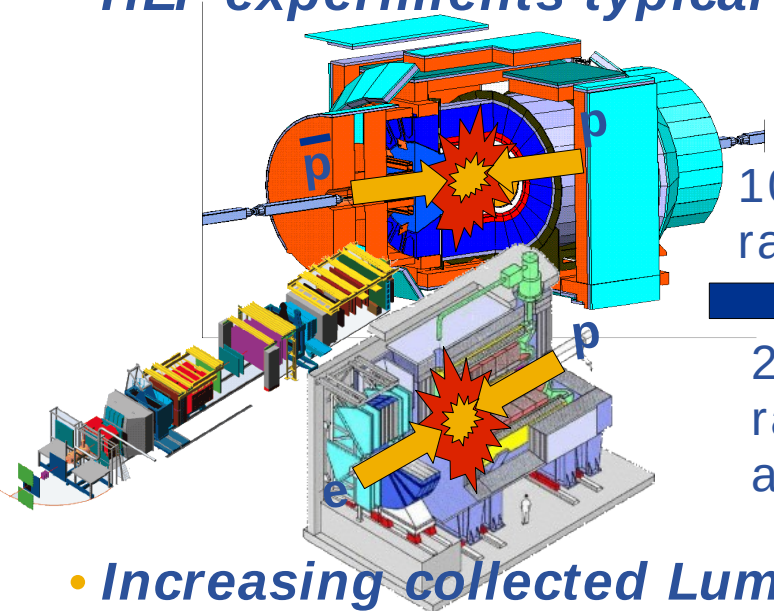
- Running High Energy Physics experiments had to adapt their computing model to Grid environment
- This talk will show the experience of four running experiments in using LCG Grid



- Introduction: HEP motivations for going to the Grid
- CDF (Tevatron) approach (Pagan Griso)
- HERA (H1 and ZEUS) approach (Wissing and Stadie)
- COMPASS approach (Duic)

Introduction – HEP experiments

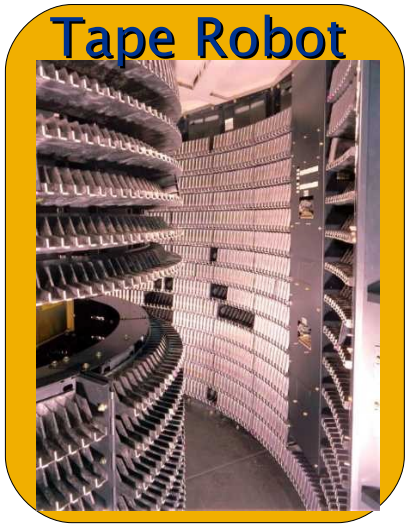
- ***HEP experiments typical data flow***



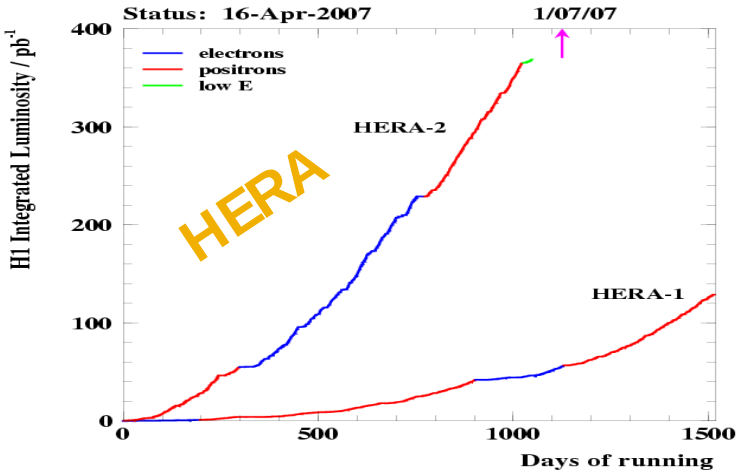
100 – 150 KB
raw event size

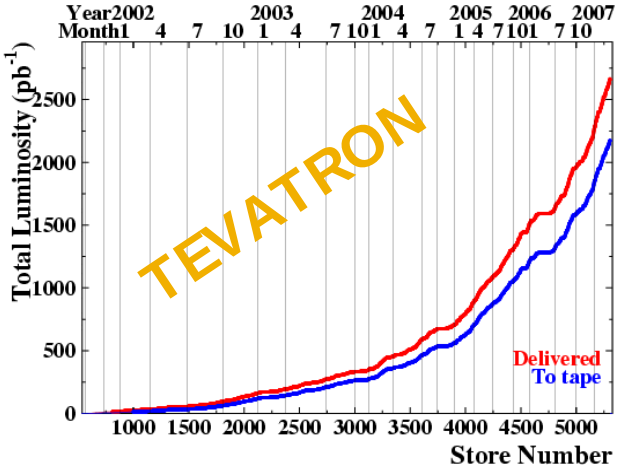
20-130 Hz
rate of events
after trigger

10-30 MB/s



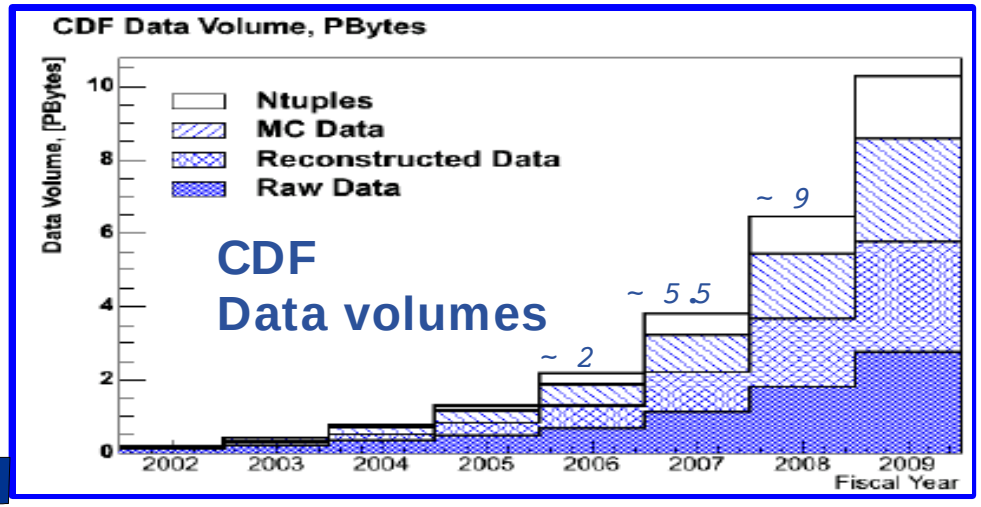
- ***Increasing collected Luminosity***





Introduction – Moving to Grid

- **Offline analysis require huge Monte Carlo simulations**



- **These simulations require a large amount of CPU power**

- **Dedicated pools are not sufficient anymore**
- **Need man power for maintenance**

Experiment	Power [KspecInt2K•s/ event]
CDF	~11
H1/ZEUS	2–10
COMPASS	~3

Started exploiting Grid resources

LcgCAF: CDF European approach to the LCG Grid

Authors and Co-Authors:
Dr. Pagan Griso, Simone (INFN and University of Padova)
Dr. Lucchesi, Donatella (INFN and University of Padova)
Dr. Compostella, Gabriele (INFN and University of Trento)
Dr. Sfiligoi, Igor (Laboratori Nazionali di Frascati)
Dr. Sarkar, Subir (INFN Pisa)
Dr. Jeans, Daniel (INFN National Center for Telematics and Informatics)
Dr. Delli Paoli, Francesco (INFN Padova)

- CDF also exploit American OSG Grid with a different method, named NamCaf; not shown in this talk

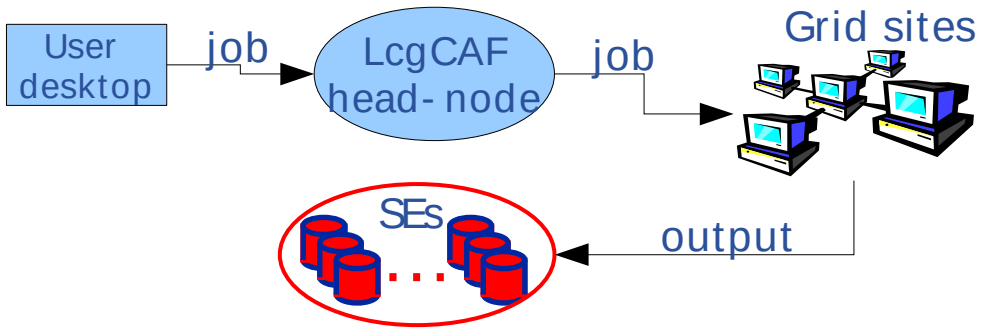
LcgCAF: CDF approach to Grid

http://www.pi.infn.it/cdf-italia/public/offline/lcgcaf.html

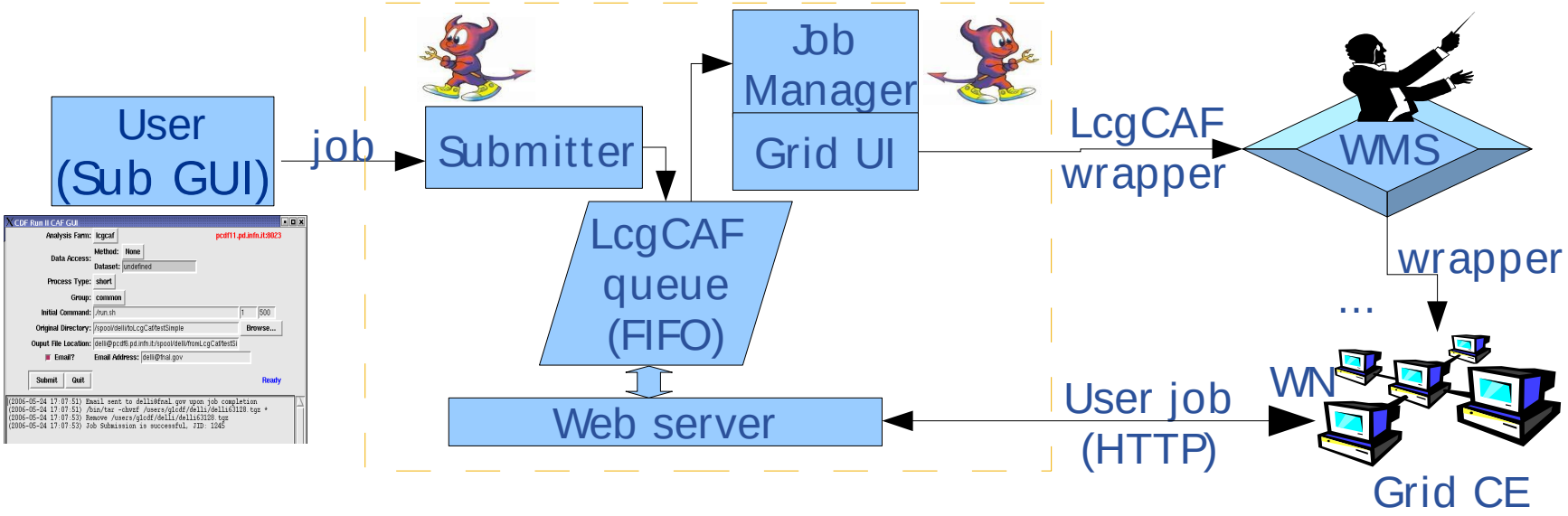
- **CDF computing:**
  - CDF jobs are defined as collections of identical “segments”
  - Each segment is made of
    - A copy of a user-provided tarball
    - The command to be executed (ex. ./run.sh $)

Incremental number of the segment

- **LcgCAF is designed to:**
  - Access Grid resources for Monte Carlo production only using EGEE/LCG middleware
  - Be used (for the user) in the same way as already existing dedicated farms: Cdf Analysis Farms (CAFs)
    - *Submission*
    - *Authentication*
    - *Job Monitor*
    - *Output Storage*

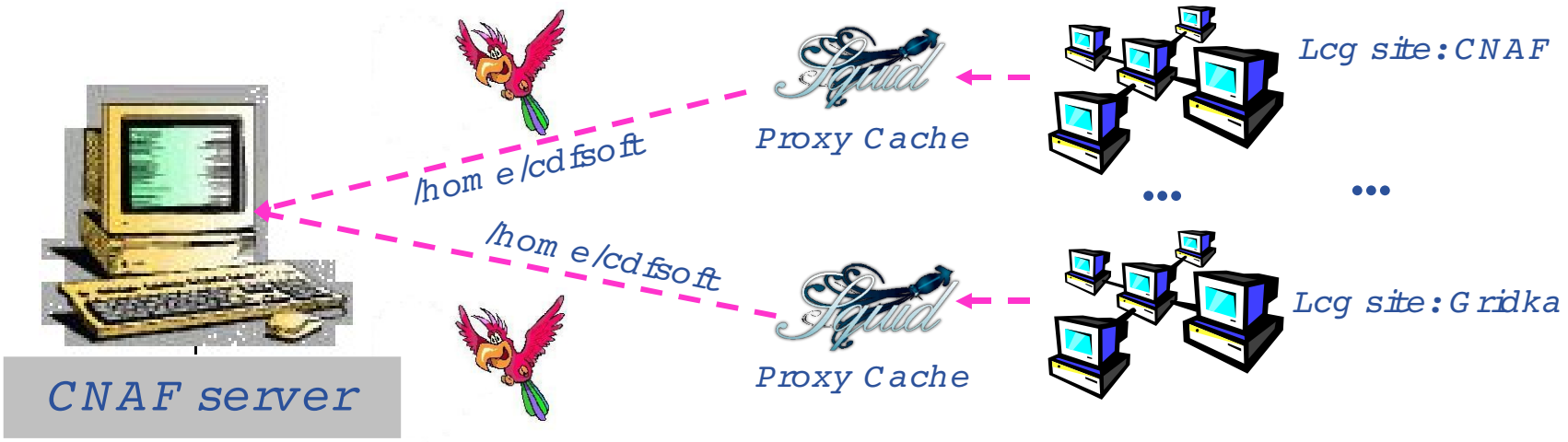


Submission and Execution



- **User submit using standard CDF software**
- **Each segment is treated as single job for the Grid**
- **LcgCAF send a wrapper to the WMS that manage it**
- **On the Worker Node (WN):**
  - Retrieve user job via HTTP (easy cache possible!) and run it,
  - Forks monitoring process,
  - When the job finishes, save the output.

CDF code distribution



- **Each job needs CDF software and run conditions**

To distribute software:

- Parrot is used as virtual file system to get it
  - Hook system calls and retrieve needed files via HTTP

To access FNAL Database (run conditions stored)

- Use Frontier to translate DB queries into HTTP request

- **Use SQUID proxies as cache near bigger sites improve performances!**

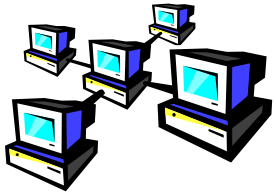
Job Monitoring

LcgCAF combines WMS and custom informations to provide interactive monitor capabilities

LcgCAF head-node

Information System

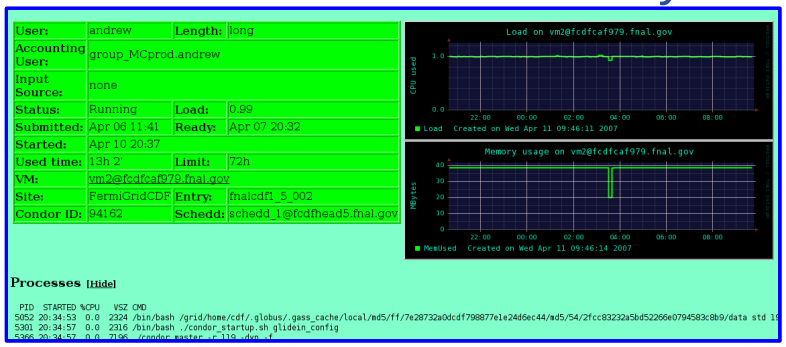
WMS: keep track of job status on the Grid



WN: Job wrapper collects infos (cpu and mem load, dir content,...) and send them to the IS (file-based)

Web monitor

Jobs overview and history



Interactive monitor

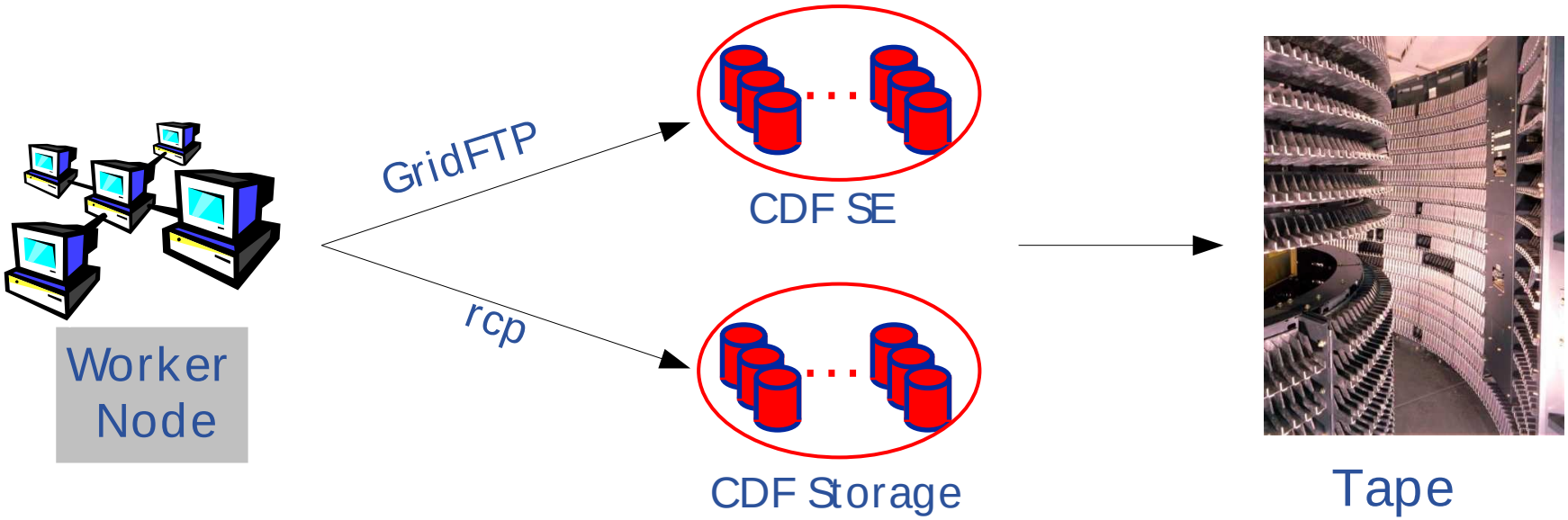
Specific job informations sent to user desktop

```
$ CafMon jobs

Analysis Farm: lcgcaf     Host: pcdf11.pd.infn.it
    Job         Group     From      To    Status

    1208        short       Total:    50
-------------------------------------------------------------
    1208        short         1       11 Pending
    1208        short         7       15 Running
    1208        short        16       50 Success
-------------------------------------------------------------
    1208        short       Success:    42 Pending:     8
    1208        short       Success:   84% Pending:   16%
$
```

Output storage



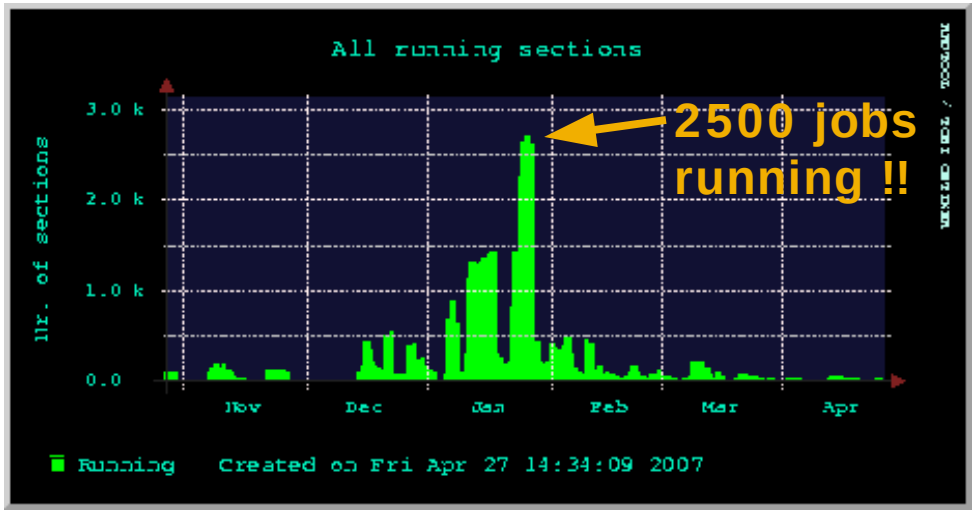
- **User output copied to CDF Storage Elements using**
  - Grid specific tools (GSI authentication using Grid proxy)

  or to CDF storage locations with
  - Rcp-like tools (Kerberos V authentication, CDF default)

- **Files are then transferred (after validation) to tape**

Performances

- **Extensive Monte Carlo production in Jan '07**



- **Getting ready to another soon!**
- **Users are using it also for other CPU intensive analysis jobs**

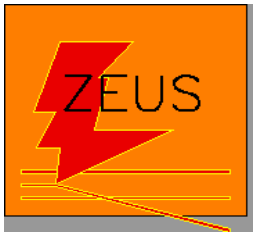
LCG Grid sites used by LcgCAF

INFN-T1	Italy
INFN-Padova	Italy
INFN-Catania	Italy
INFN-Bari	Italy
INFN-Legnaro	Italy
INFN-Roma1	Italy
INFN-Roma2	Italy
INFN-Pisa	Italy
FZK-LCG2	Germany
IEPSAS	Slovakia
IFAE	Spain
PIC	Spain
IN2P3-CC	France
UKI-LT2-UCL-HE	UK
Liverpool	UK

Conclusions

- **CDF has successfully exploited Grid resources using EGEE/ LCG middleware to produce MC simulations**
- **LcgCAF is now in 'production'...**
- **Stability is the key-word for a running experiment**
  - moving to WMS 3.1 → first tests show good performances
- **Adding new resources to LcgCAF**
- **Now users have "FIFO" priorities. We're testing G-PBox to implement policies at site and WMS level**
- **Finally: Running on data is far, but also possible!**

Monte Carlo Mass Production of the H1 and ZEUS Experiments

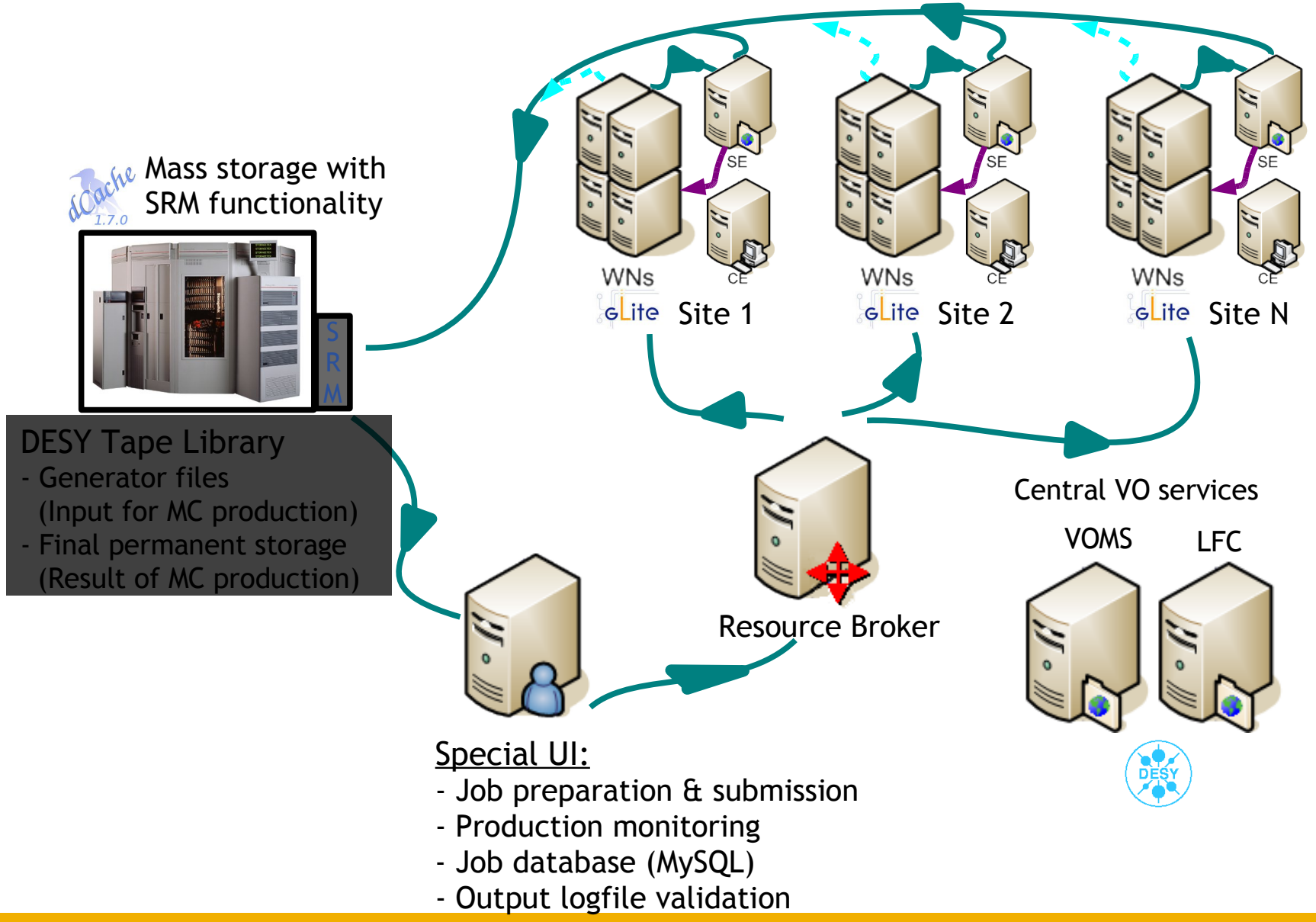


Christoph Wissing (DESY) and
Hartmut Stadie (Universität Hamburg)

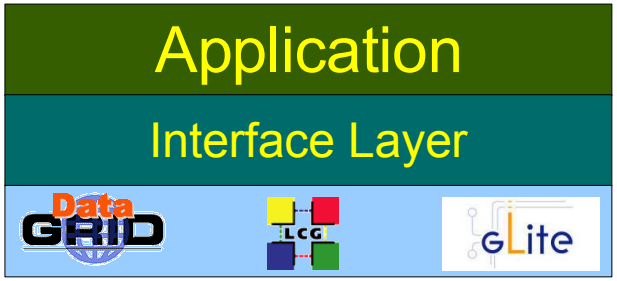
for the development & production teams of H1 and ZEUS:

M. Ernst (BNL, DESY), J. Ferrando (U Oxford), A. Fomenko (LPI),
A. Gellrich (DESY), M. Karbach (U Dortmund), B. Lobodzinski (DESY),
R. Mankel (DESY), T. Namsoo(DESY), T. Preda (NIPNE), M. Vorobiev (ITEP),
K. Wrona (DESY)

Production Chain



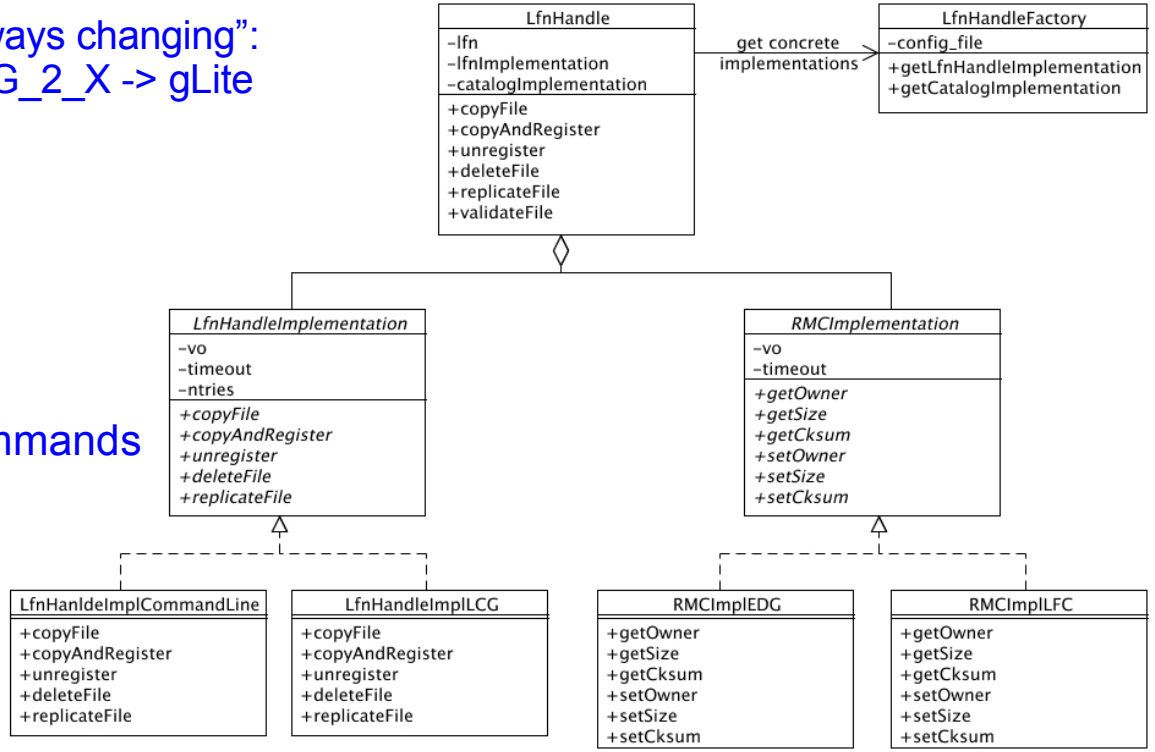
Introduction -



Avoid changes not related to application

Adapt to changes in middleware

"Always changing":
LCG_2_X -> gLite

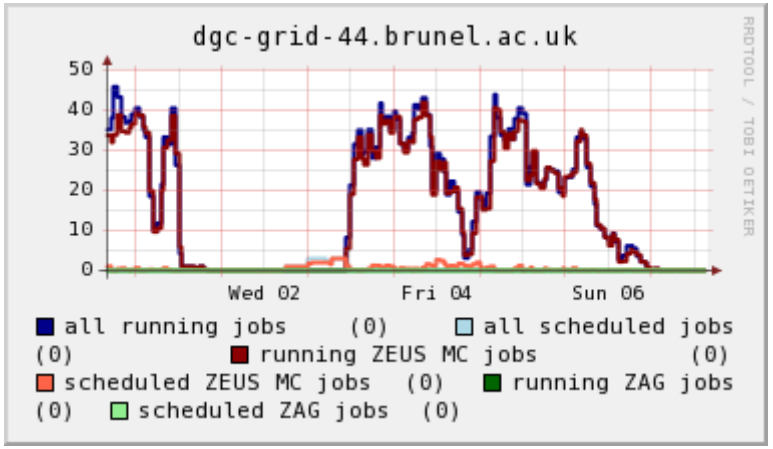


- **Object oriented PERL:**
  - Wrapper around edg-/lcg-commands
  - Enforced timeouts
  - Automatic retries
  - Additional checks
  - Needs working Perl APIs in gLite (32 & 64 Bit)
- **Code basis for both experiments:** **ZEUS Grid-Toolkit**

http://www-zeus.desy.de/~wrona/grid/index.php

Monitoring 1

- **Essential for mass production**
- **Many histograms created automatically**
  - Jobs at sites, error rates etc.
  - Based on RRD tool
- **Details about jobs via web interface**

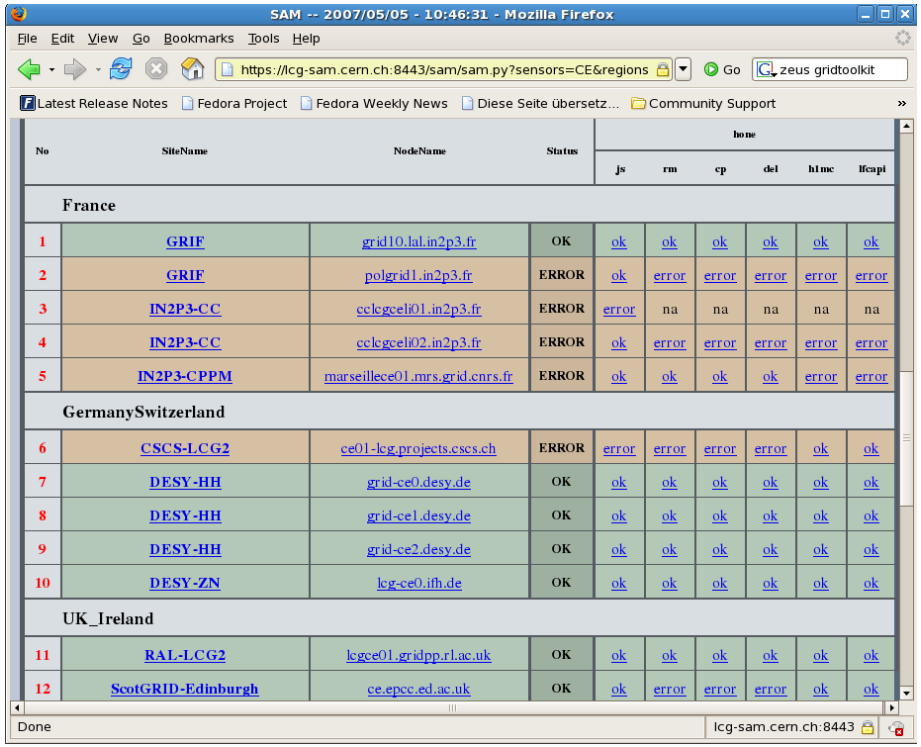


Example from ZEUS RRD plots

H1 interface: Job details

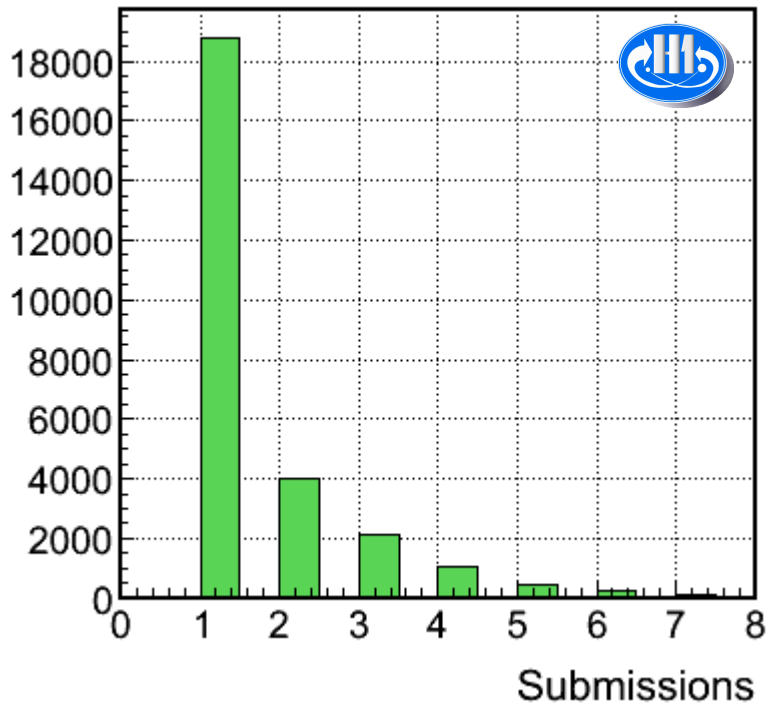
Monitoring 2

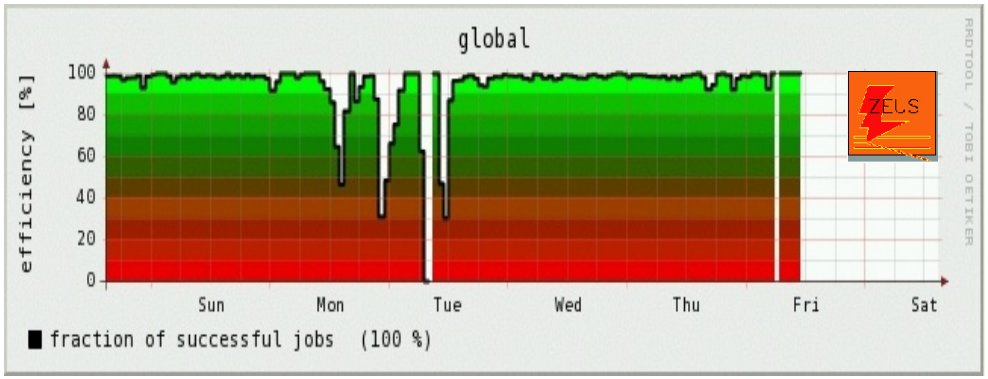
- **Selection of "good sites"**
  - Black/White listes maintained by production teams
  - Input: Own monitoring
- **SAM test by H1**
  - OPS tests
  - H1 specific tests
    - Mini MC job
    - Check of LFC API
- **Very good cooperation with site admins**
  - VO specific issues get fixed
  - "Why don't you run on my site?"



No	SiteName	NodeName	Status	lxone					
				js	rm	cp	del	lrmc	lfcapi
	France								
1	GRIF	grid10.lal.in2p3.fr	OK	ok	ok	ok	ok	ok	ok
2	GRIF	polgrid1.in2p3.fr	ERROR	ok	error	error	error	error	error
3	IN2P3-CC	cclcgceli01.in2p3.fr	ERROR	error	na	na	na	na	na
4	IN2P3-CC	cclcgceli02.in2p3.fr	ERROR	ok	error	error	error	error	error
5	IN2P3-CPPM	marseillece01.mrs.grid.cnrs.fr	ERROR	ok	ok	ok	ok	error	error
	GermanySwitzerland								
6	CSCS-LCG2	ce01-lcg.projects.cscs.ch	ERROR	error	error	error	error	ok	ok
7	DESY-HH	grid-ce0.desy.de	OK	ok	ok	ok	ok	ok	ok
8	DESY-HH	grid-ce1.desy.de	OK	ok	ok	ok	ok	ok	ok
9	DESY-HH	grid-ce2.desy.de	OK	ok	ok	ok	ok	ok	ok
10	DESY-ZN	lcg-ce0.ifh.de	OK	ok	ok	ok	ok	ok	ok
	UK_Ireland								
11	RAL-LCG2	lcgce01.gridpp.rl.ac.uk	OK	ok	ok	ok	ok	ok	ok
12	ScotGRID-Edinburgh	ce.epcc.ed.ac.uk	OK	ok	error	error	error	ok	ok

Production Experiences



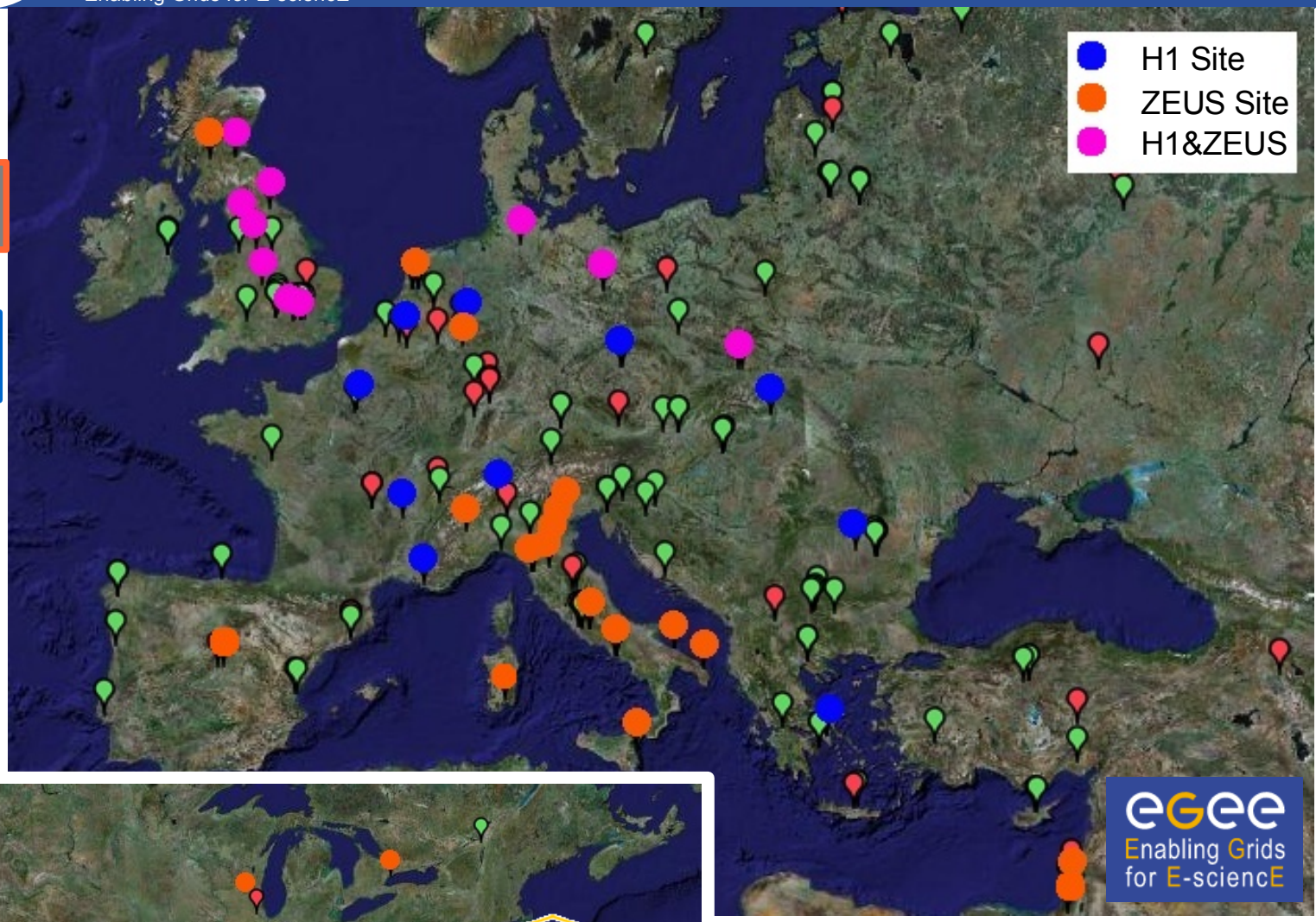


- **Single job efficiency ~80%**
  - Stuck jobs in RB/CE
  - Failure due to data transfer errors
  - Bad site or bad worker node
  - Global failures (RB, catalog)
- **Integrated efficiency 99.9%**
  - Automatic resubmission
  - Almost unattended production
  - Minimal manual interference

H1 & ZEUS Production Sites

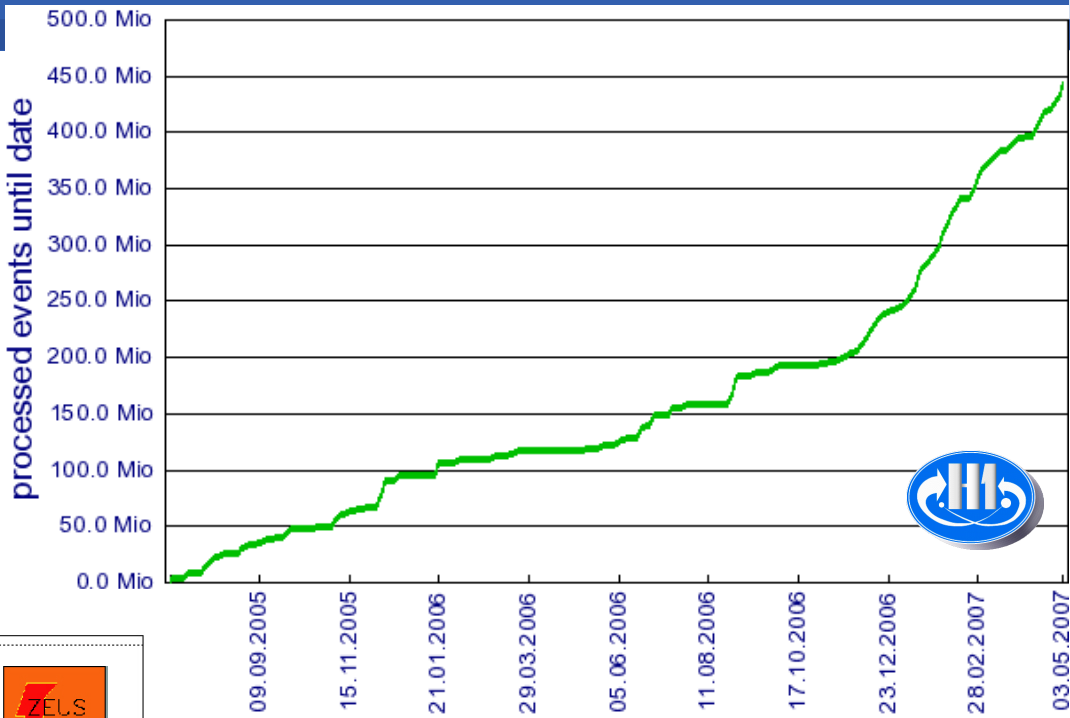
ZEUS: 8000 CPUs

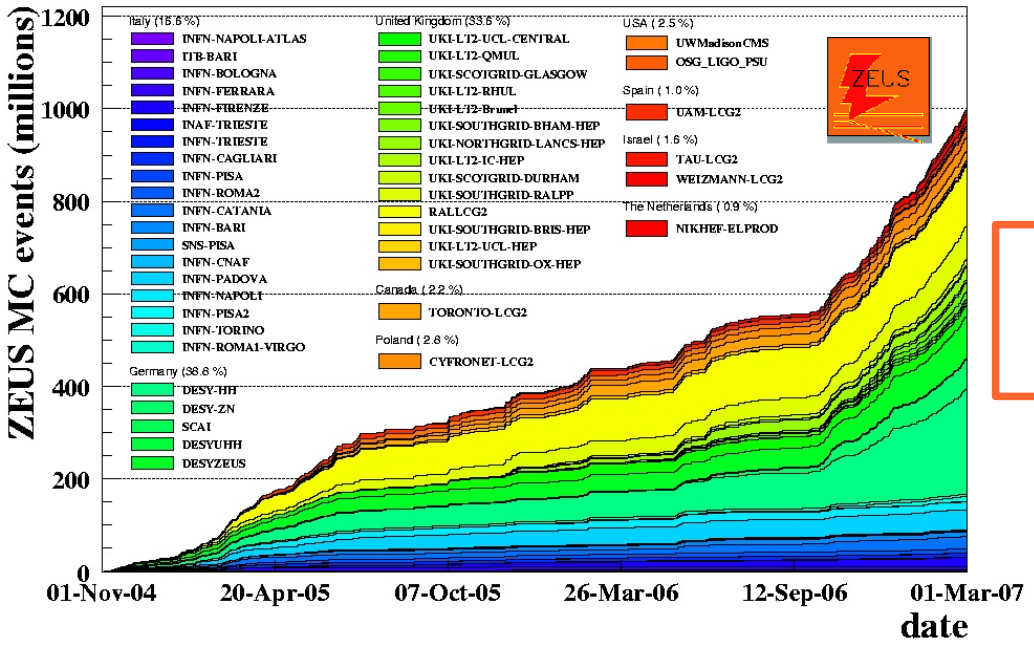
H1: 6000 CPUs



Overall Production

Almost 0.5 Billion Events by H1
~ 25 TB MC data in last 12 month





1 Billion Events by ZEUS
~ 80 TB MC data in last 12 month

Summary: H1 & ZEUS

- **Successful production since more than 2 years**
- **Frame works in object oriented PERL**
  - Work around deficiencies in the current middleware
  - Many monitoring capabilities
  - Almost unattended processing
- **Common effort**
  - Developers & production teams of the experiments
  - DESY team running central VO services & mass storage
  - Ambitious sites admins
- **Analysis on the Grid: ZEUS started on selected sites**
- **More DESY hosted VOs active on the Grid**
  - CALICE (Calorimeter for linear collider) - Test beam data
  - ILC - Mass production by individuals
  - ILDG (Lattice QCD) - Storage of “configurations”

MC Production moves to grid @ COMPASS

V. Duic, G. Sbrizzai, A. Martin
University of Trieste/INFN-Trieste

COMPASS - Computing Model

COMPASS physics data

- High luminosity (reconstr. events) 2×10^{32} cm^{-2} s^{-1} (^{6}LiD target)
- Huge amount of data ~400 TB/y - up to now 1PB = 3^{10} events

- **Event reconstruction** performed @ CERN
  - ***C**OMPASS **C**omputing **F**arm (CCF): 500k SI2k*
- **Simulation & analysis** performed @ collaborating Institutes
  - *Farms scattered around the world ("reduced-scale CCFs")*

Simulation chain computing power/event

Programme	kSPI2k·s	
LEPTO	~ 0.02	(event generation)
COMGEANT	~ 1.5	(prop. thru spectr.)
CORAL	~ 1.7	(ev. reconstruction)
==========		
Complete chain	**~ 3.22**	

COMPASS - Computing Model

COMPASS @ Trieste - computing model

- Main programme: “transversity” (*transverse spin effects in hadron production*)
  - ¼ of COMPASS data = 10^7 reconstructed events (after cuts)
- Requires massive MC simulation and processing (and analysis)
  - main analysis channel (azimuthal asymmetries in hadron production in SIDIS/transversally polarised target): 3-4 × real data
  - other future channel(s) (i.e. Cahn effect): 10 × real data

  (for both, multiple reprocessing + different parameter settings)
- Trieste has set up a computing farm: “**ACID**” (50k SI2k)
  - Trieste **SIMULATION** model (local-farm “centric”)
    - *MC data produced and analysed at home (up to now self-sufficient)*
    - *Outbound (grid) production performed only in case the local resources are insufficient.*
  - Simulation chain is LEPTO + GEANT3 (both or single step)

COMPASS - Simulation framework

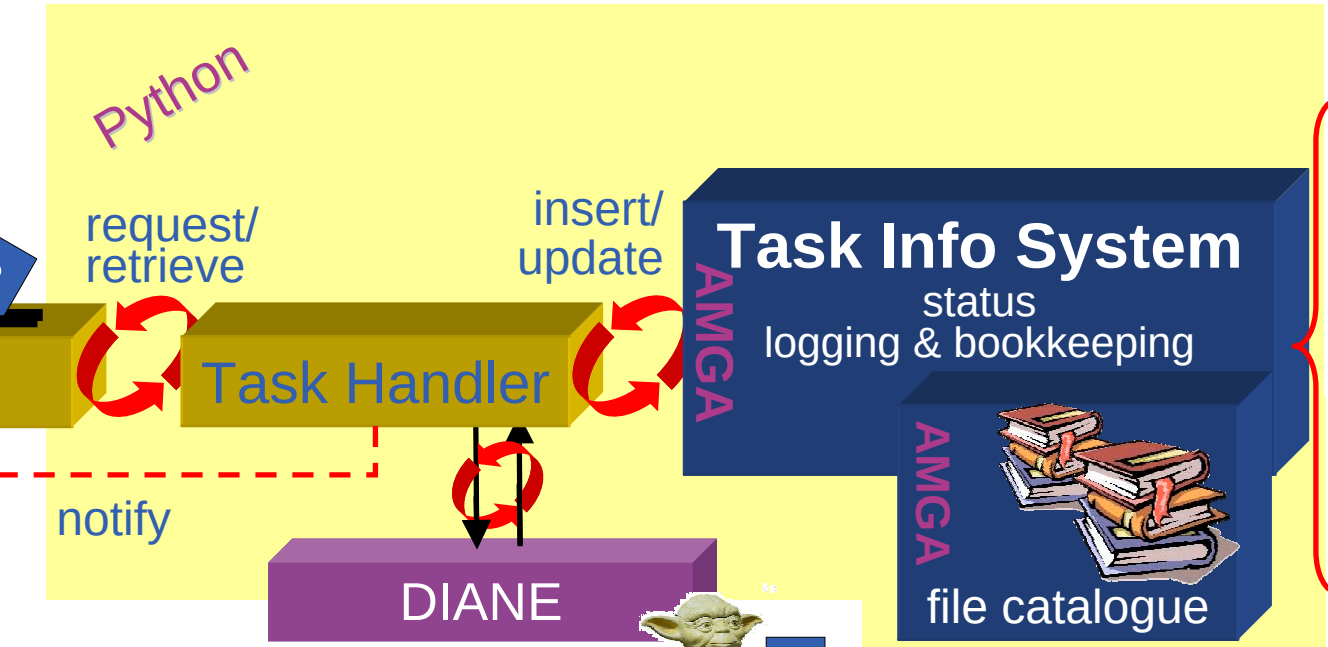
MC production framework - new design

- Hinged on a file catalogue
  - to ease the handling of (a large # of) simulated datasets
    - *data aggregation based on (physics) metadata*
    - *transparent dataset access (FS fragmentation and data-life-cycle relocation is per se a local issue)*
- Integrates with grid technologies
  - to let (world-wide distributed) collaborating Institutes
    - *share the simulated data*
    - *mutually exploit available computing resources inside the Collaboration - as a (sub-)VO*
- Makes use of existing tools: gLite + AMGA + DIANE, glued together within a framework for the MC data production through a set of python scripts

MC production framework - in a nutshell

- Data production and aggregation performed through a system which holds and manages the information about the current production status of each dataset requested by the user
  - Each dataset is exposed to the user as data characterised by a set of physics/MC configuration parameters
  - A dataset translates into a production task. Simulated data aggregation returns dataset corresponding to query (or can be set to trigger a new data simulation if dataset not already produced)
  - A production task is split into subtasks, each wrapped in a tarball containing all the necessary stuff (execs and inputs) to perform simulation and then handed to DIANE for scheduling/submission
  - DIANE's "master" process spawns a series of "worker" agents (which pull subtasks from masters' queue), and hands-back the results upon completion and/or provides the system with the feedback about the production

COMPASS - Simulation framework



MC production framework - tech choices rationale

- Use of AMGA (w/ RDB back-end)
  - offers aggregation based on physics metadata
  - ACL control on tasks / data

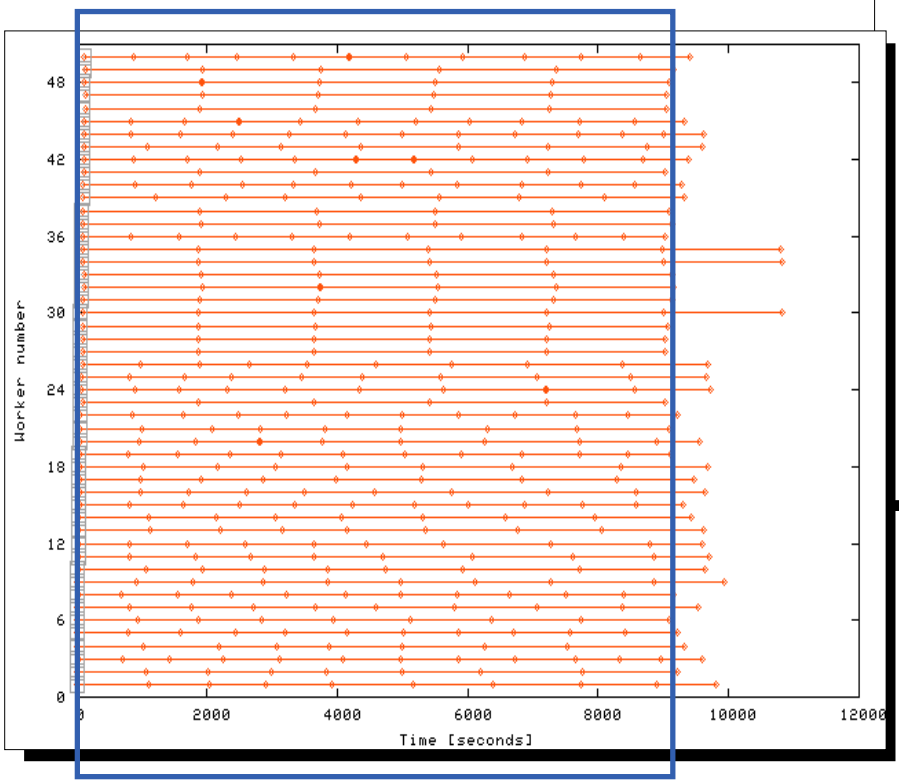
  catalogue is paramount first of all at local level
- Use of DIANE
  - allows exploitation of local (and “prompt”) computing power and mix to it grid-added power whenever needed
    - *special case: pilot jobs to evaluate simulation model/set-up easily checked locally, and ready to be ported to the grid*
  - master/worker model enhances performance in grid submission (each worker is submitted once but servers many tasks)
  - advanced (customisable) task execution fault recovery and task dispatching
  - effecive “natural” load balancing (master “vigilance”: no sink-holes)

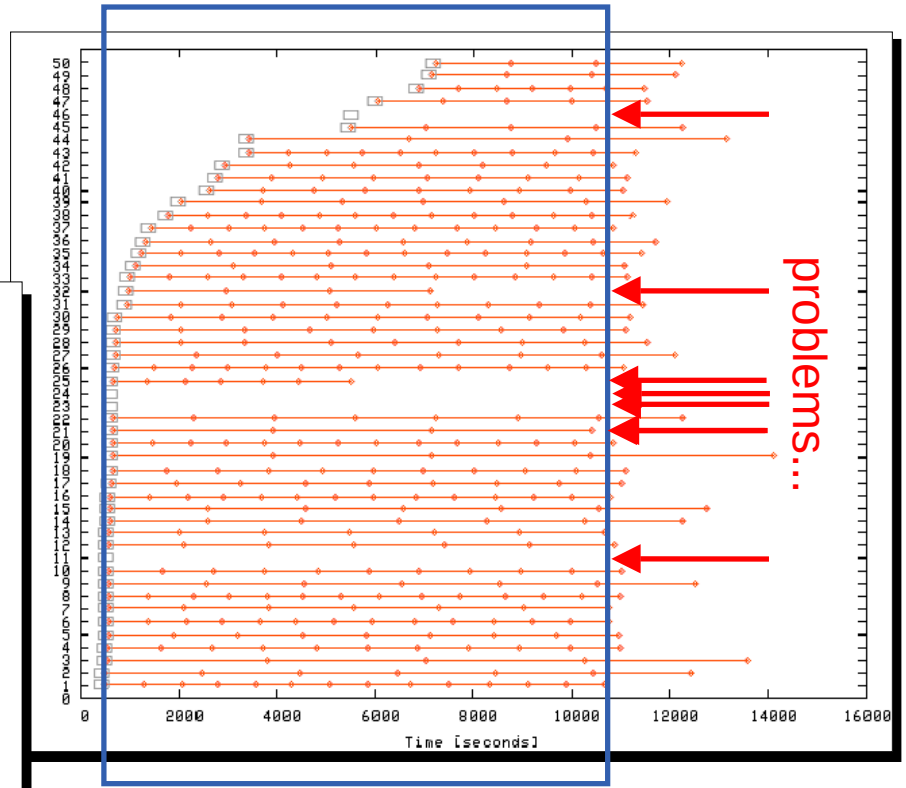
MC production framework - tech choices rationale

– Use of DIANE:

load balancing

…in the local farm





…in the grid

COMPASS - Simulation framework

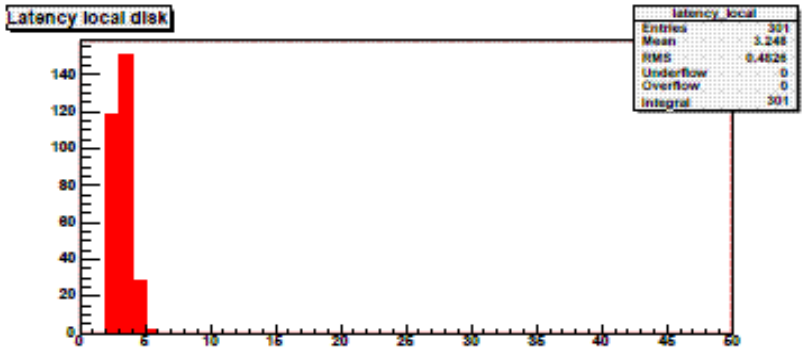
- **MC production framework - development status**
  - Prototype offering the user a CLI with basic functionalities (production jobs managing, dataset cataloguing, and data aggregation)
  - Locally, the framework can only handle data on a DPM SE (no SRM'nt for local tape w/ CASTOR yet)

- **MC production framework - future plans**
  - Debug & tune :)
  - Extend the number and complexity of provided functionalities
  - Add a GUI scientist-friendly front-end interface / web portal
  - Integrate also with COMPASS reconstruction framework CORAL (too complex in a grid env. - unless CEs offer installed CORAL)

End

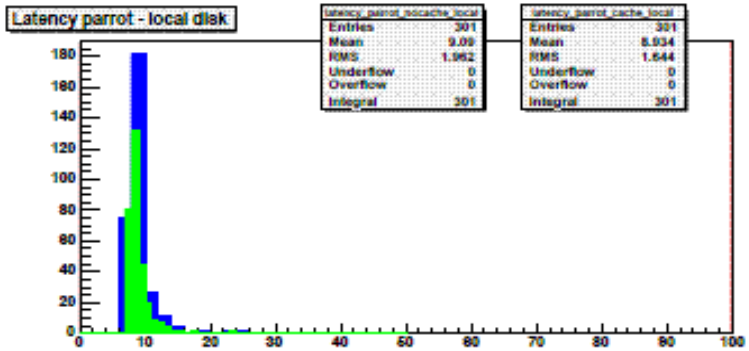
CDF Backup slides

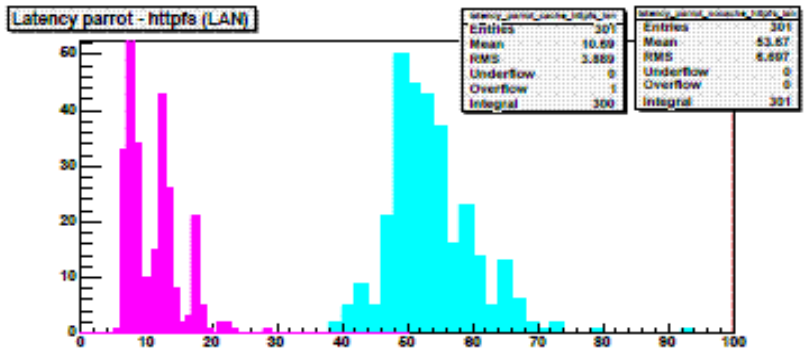
Parrot performances

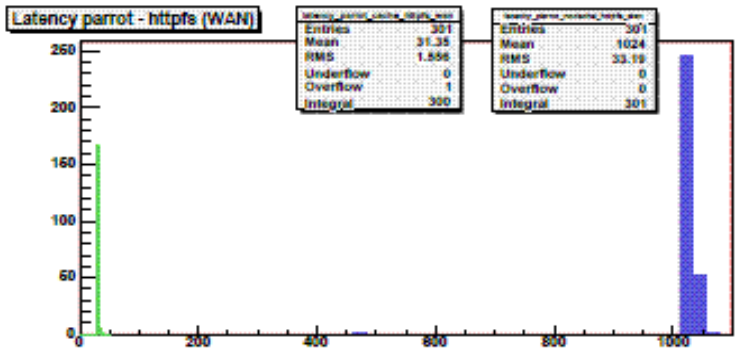


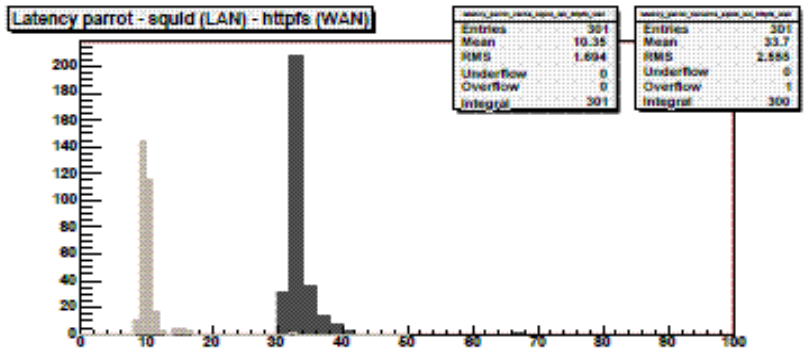
latencyTime = reatTime - (userTime + sysTime) / ncpu

- local
- parrot w/o cache - local
- parrot w/ cache - local
- parrot w/o cache - httpfs (LAN)
- parrot w/ cache - httpfs (LAN)
- parrot w/o cache - httpfs (WAN)
- parrot w/ cache - httpfs (WAN)
- parrot w/o cache - squid (LAN) - httpfs (WAN)
- parrot w/ cache - squid (LAN) - httpfs (WAN)

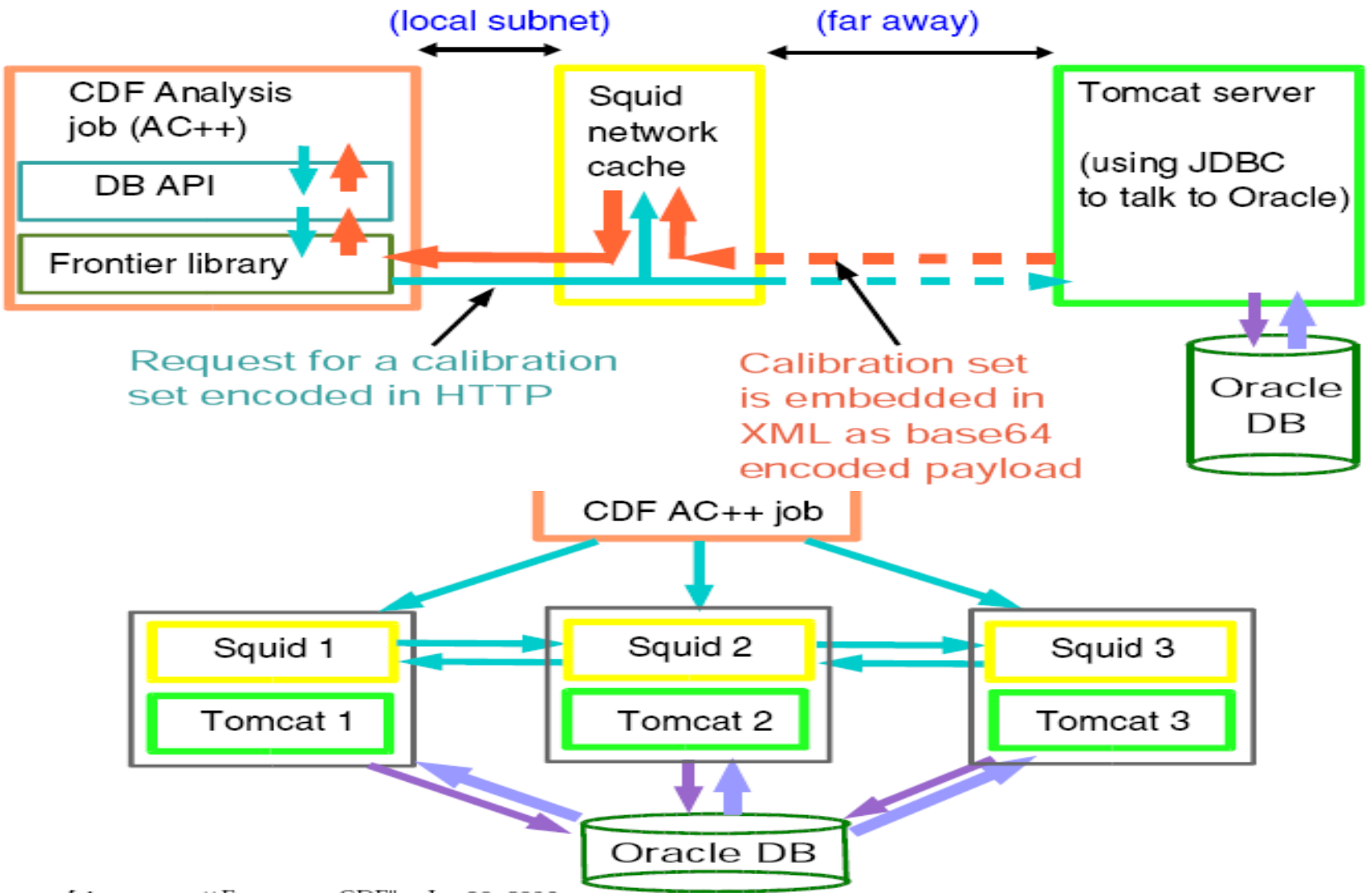






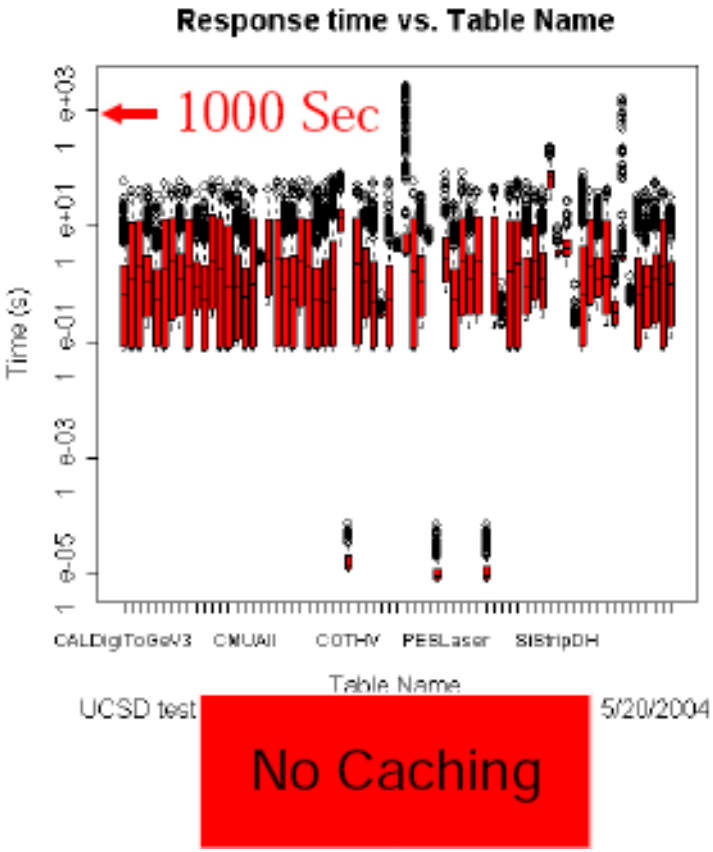


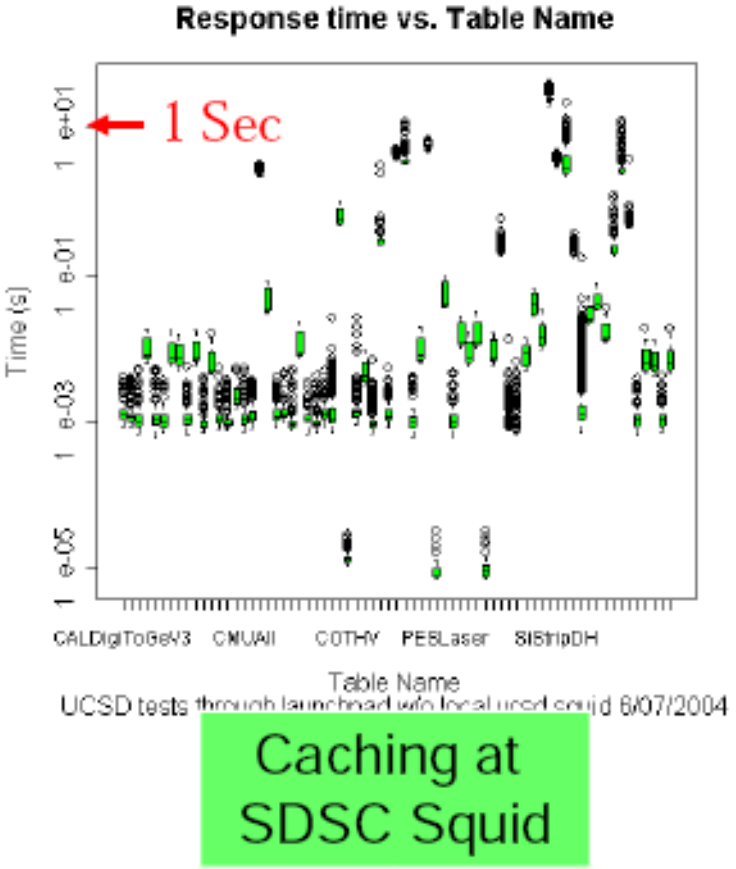
Frontier - I


(local subnet)
(far away)
CDF Analysis job (AC++)
DB API
Frontier library
Squid network cache
Tomcat server
(using JDBC to talk to Oracle)
Oracle DB
Request for a calibration set encoded in HTTP
Calibration set is embedded in XML as base64 encoded payload
CDF AC++ job
Squid 1
Tomcat 1
Squid 2
Tomcat 2
Squid 3
Tomcat 3
Oracle DB

Frontier - II

- **For small objects (ex. Beamline) improvements up to 1000 times!**



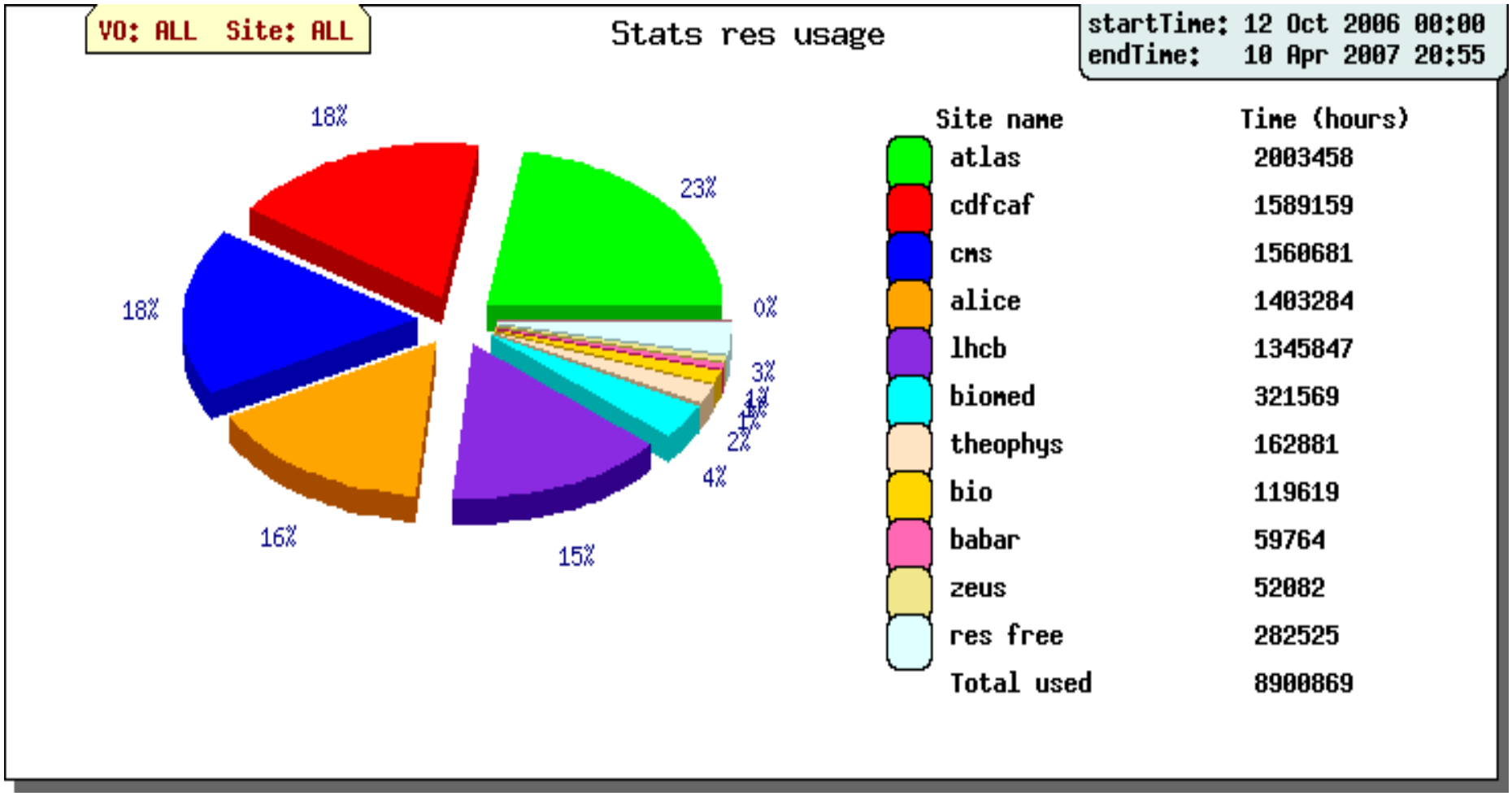


- **For normal CDF jobs improvement up to 60% (18 - > 11 hours)**

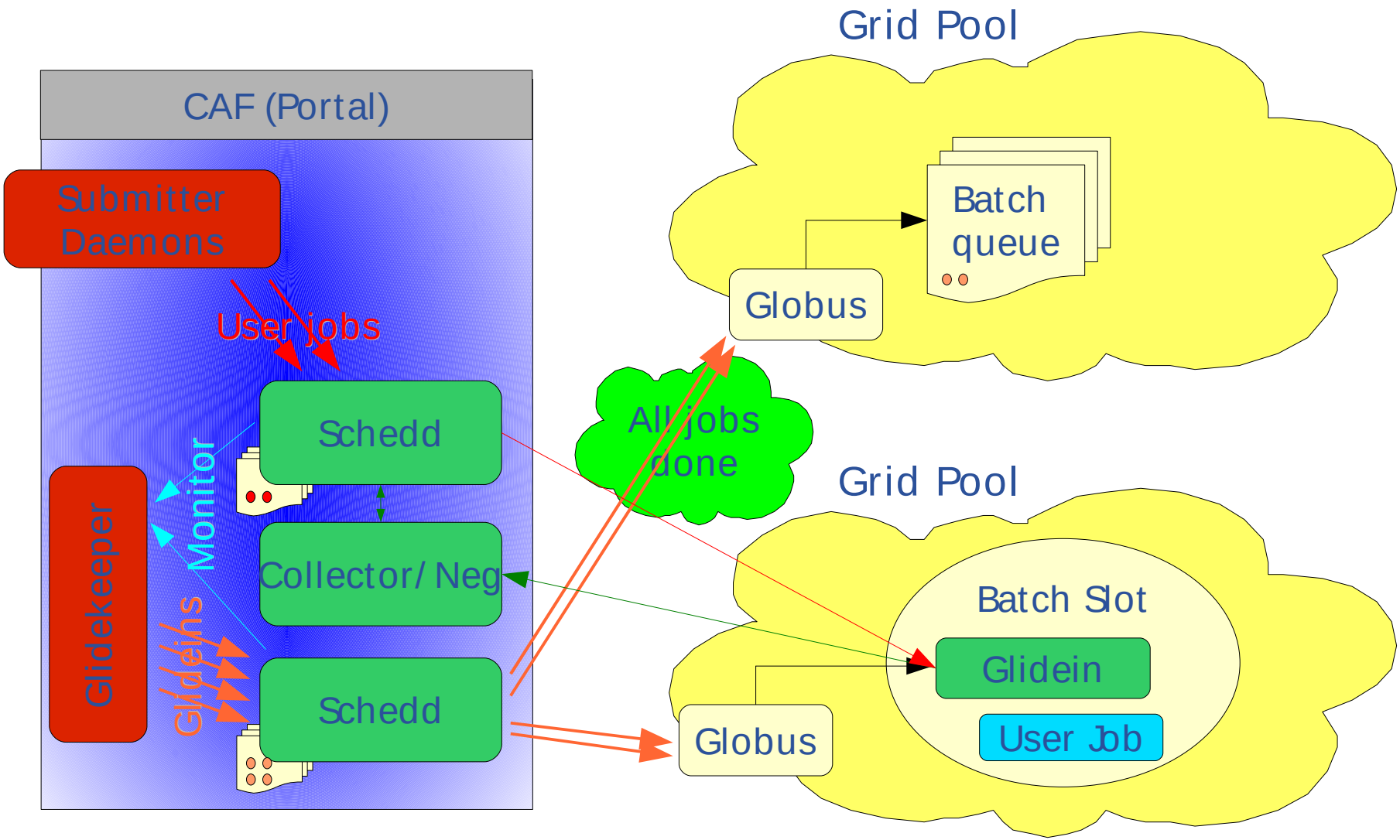
Slide from Petar Maksimovic

CDF usage of LCG resources

- **European sites not included in this plot**



NamCAF and OSG Grid



By I. Sfiligoi, S. Sarkar