Gabriele Compostella
University of Trento and INFN
compostella@tn.infn.it

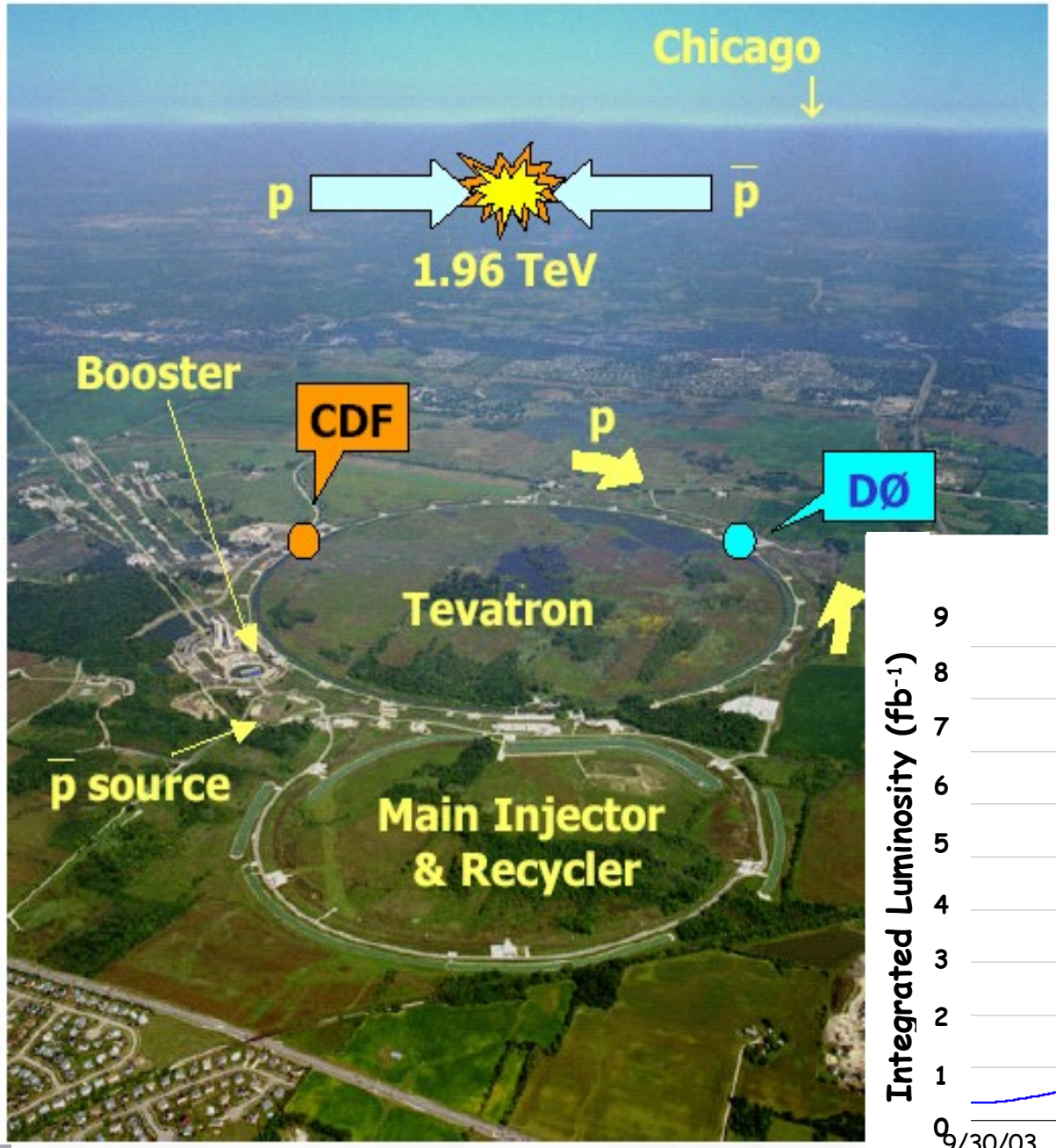
on behalf of CDF Computing Group

CDF Experience using the Grid

Workshop 2007 su Calcolo e Reti dell'INFN
Rimini

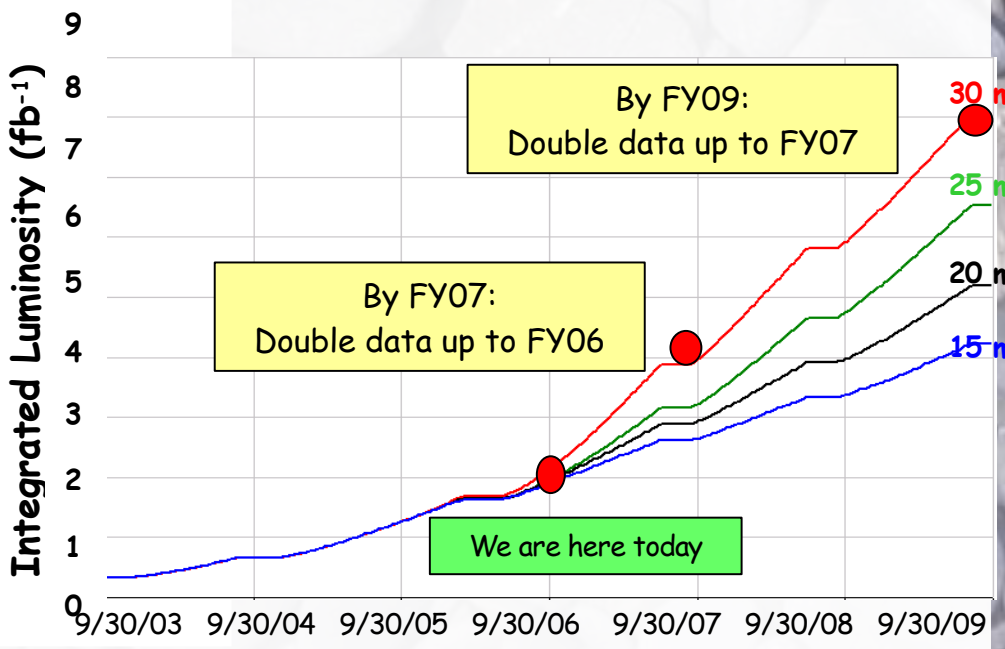
8 Maggio 2007

CDF Experiment Luminosity

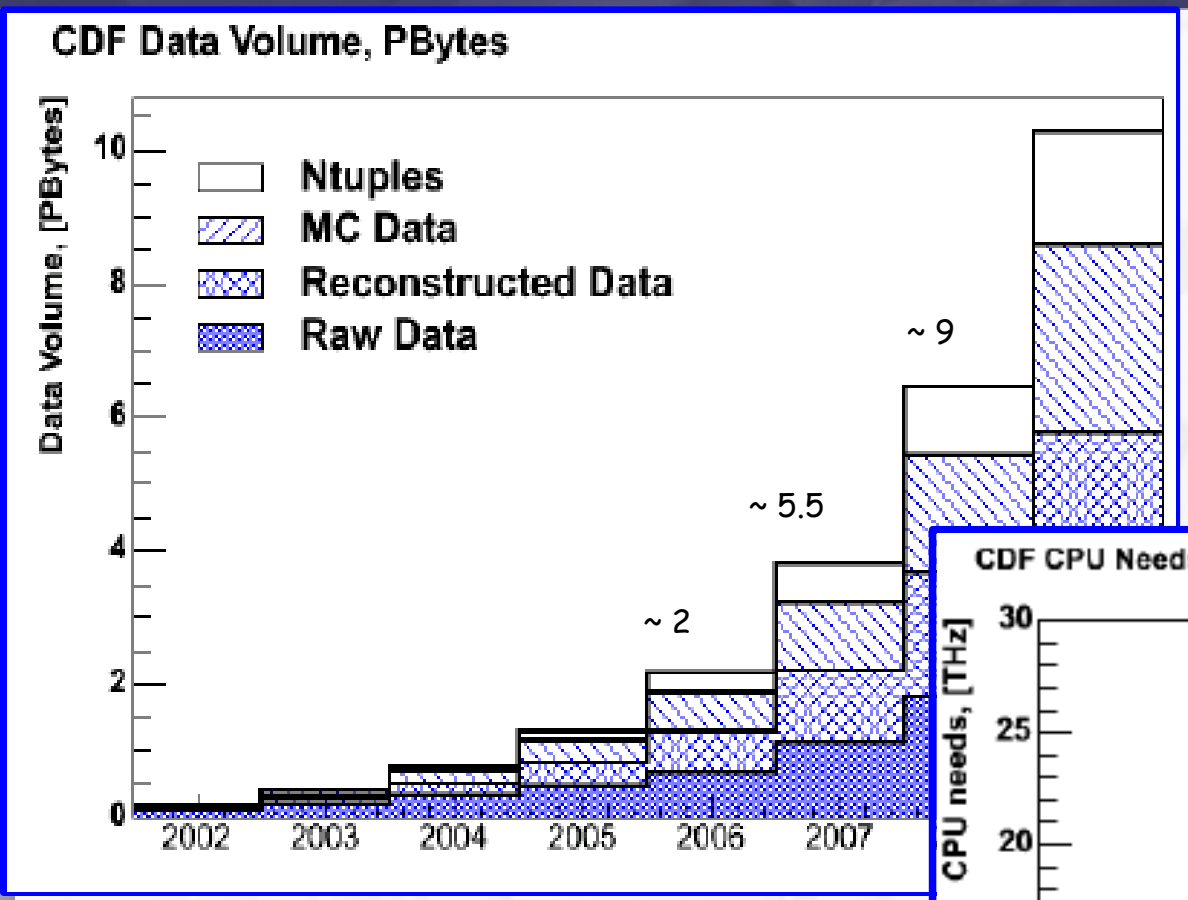


- L_{peak}: $2.3 \times 10^{32}\ s^{-1}\ cm^{-2}$
- 36 bunches
- 396 ns crossing time

Collected Luminosity

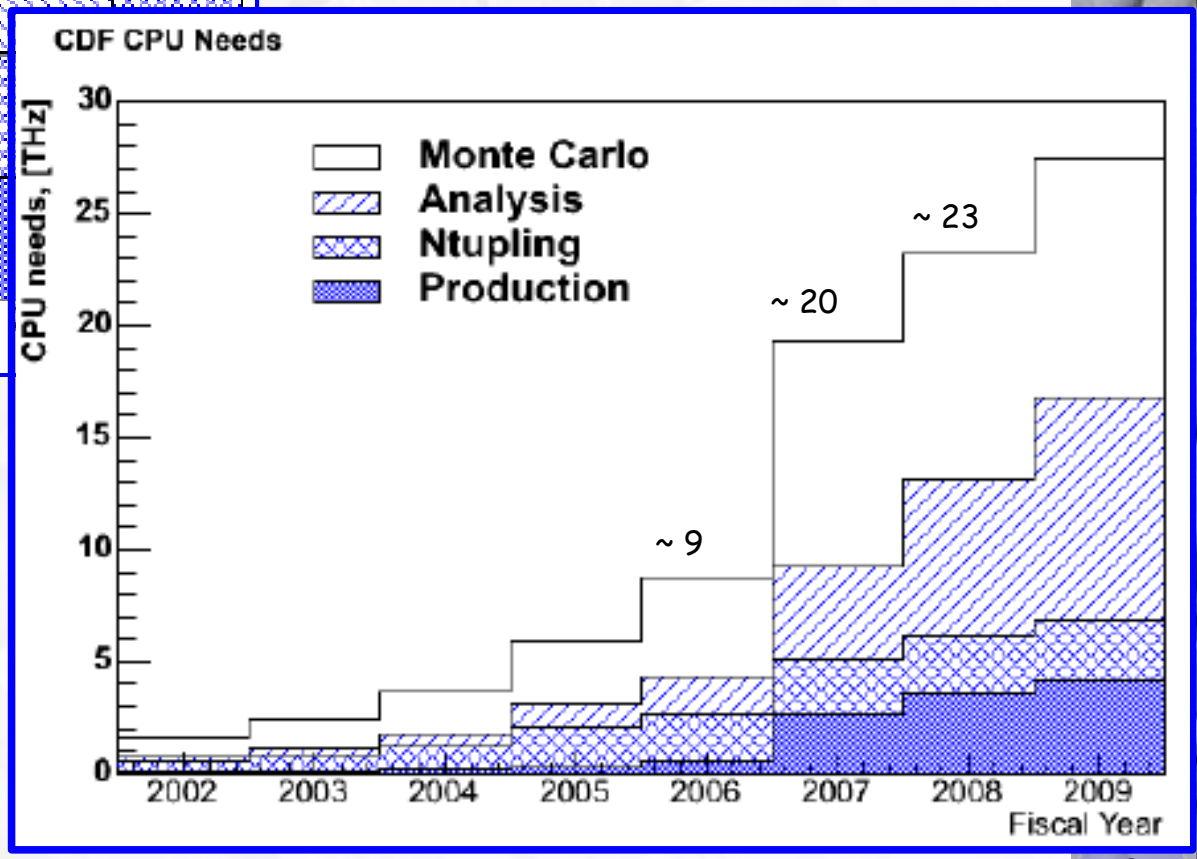


CDF Experiment Data Volumes



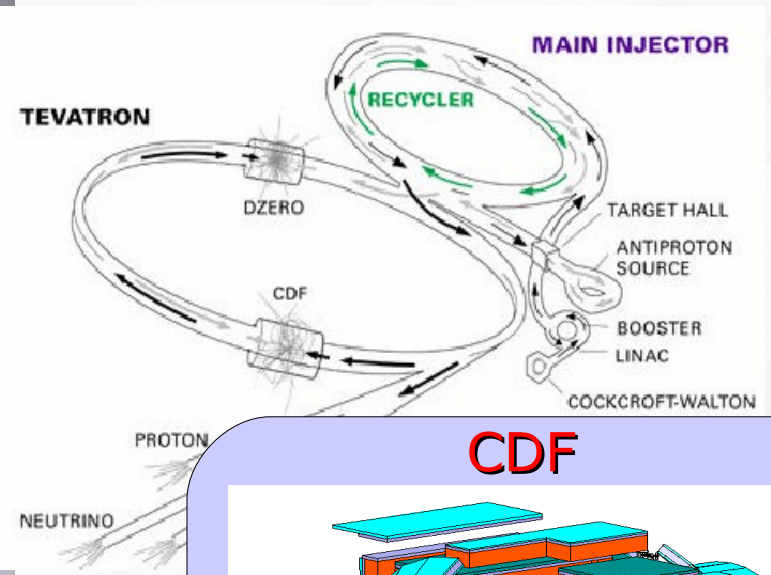
Collected Data

$(2fb^{-1} \approx 3.9 \times 10^{9} events)$

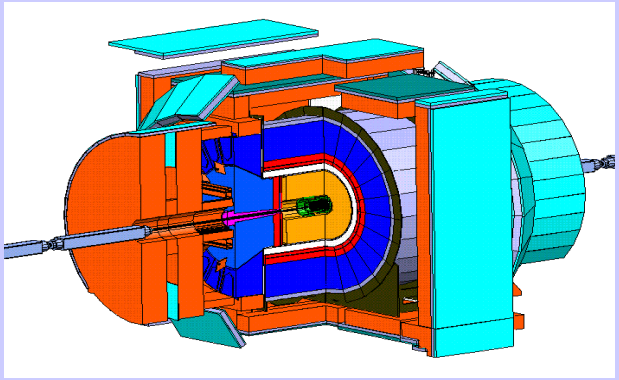


Estimated CPU needs

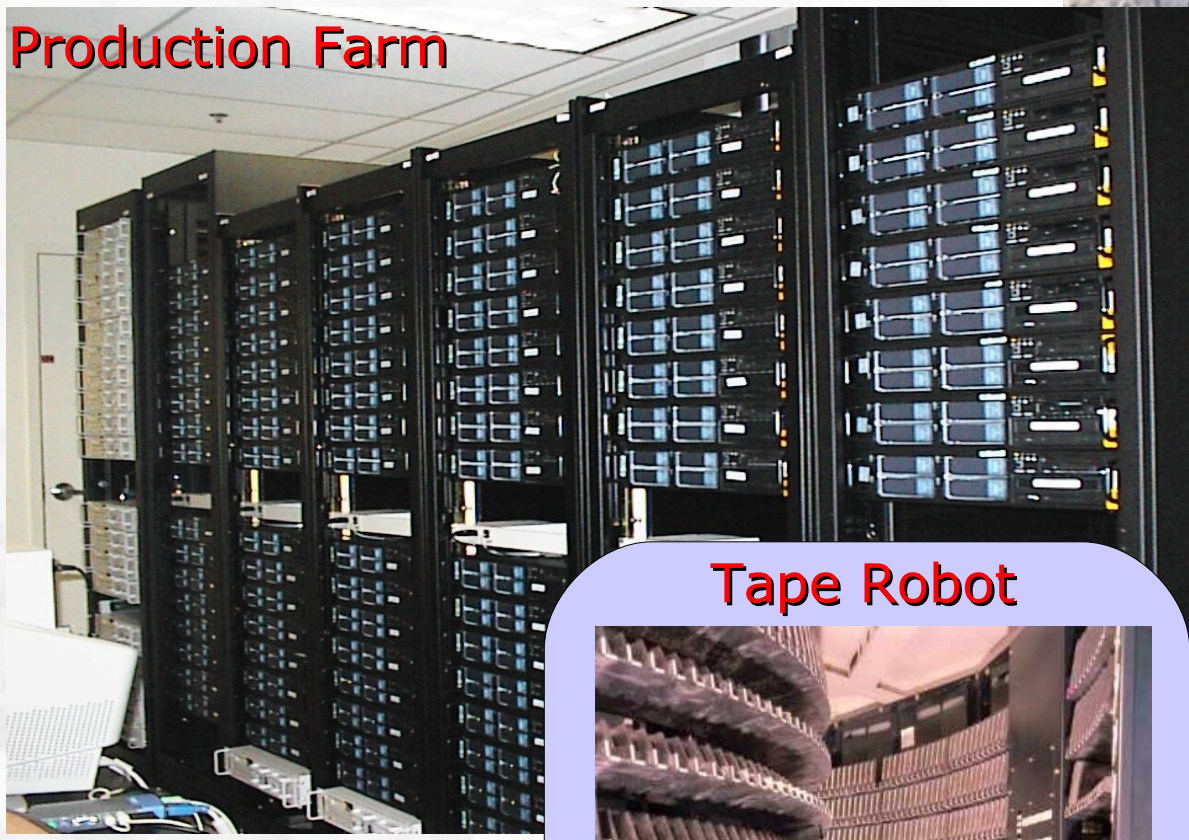
CDF Data Production



CDF



Production Farm



L3 Trigger

Production Data

Tape Robot

SAM

Disk Cache

Raw Data

SAM

- Data Handling model relies on SAM (Sequential data Access via Metadata), a centralized system that manages all official data
- File Catalog for data on tape and on disk:
  - Files based, with metadata attached
  - Files are organized in datasets, users can create their own
- Manage File Transfers: tape-to-cache, cache-to-disk, disk-to-tape
- For jobs accessing data:
  - copy necessary files from the closest location to local cache
  - provide files in the optimal order
  - keep track of processed and unprocessed files
  - Failed sections can be automatically recovered
- Each site serving data needs to have a SAM Station

Users Analysis and MC Production

- CAF Central Analysis Facility: mostly User analysis
  FNAL hosts data -> farm used to analyze them

- Decentralized CAF (dCAF): mostly Monte Carlo production
  Remote sites produce Monte Carlo, some (ex. CNAF) have also data
  Produced files are then uploaded to SAM

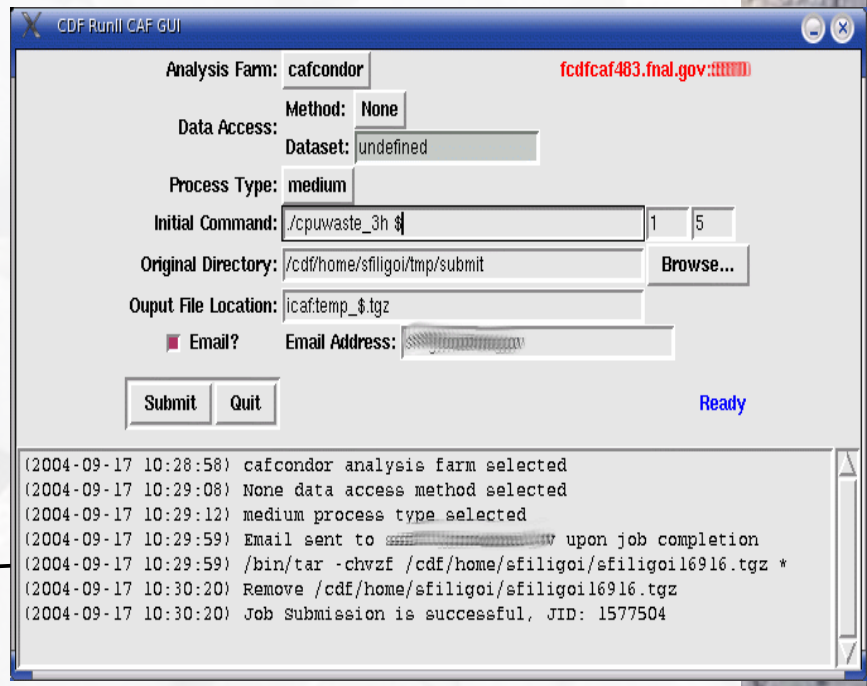
- CAF and dCAF are CDF dedicated farms

CAF Model

Develop, debug and submit from personal PC

CDF user must have access to CAF clients and specific CDF certificate

CAF



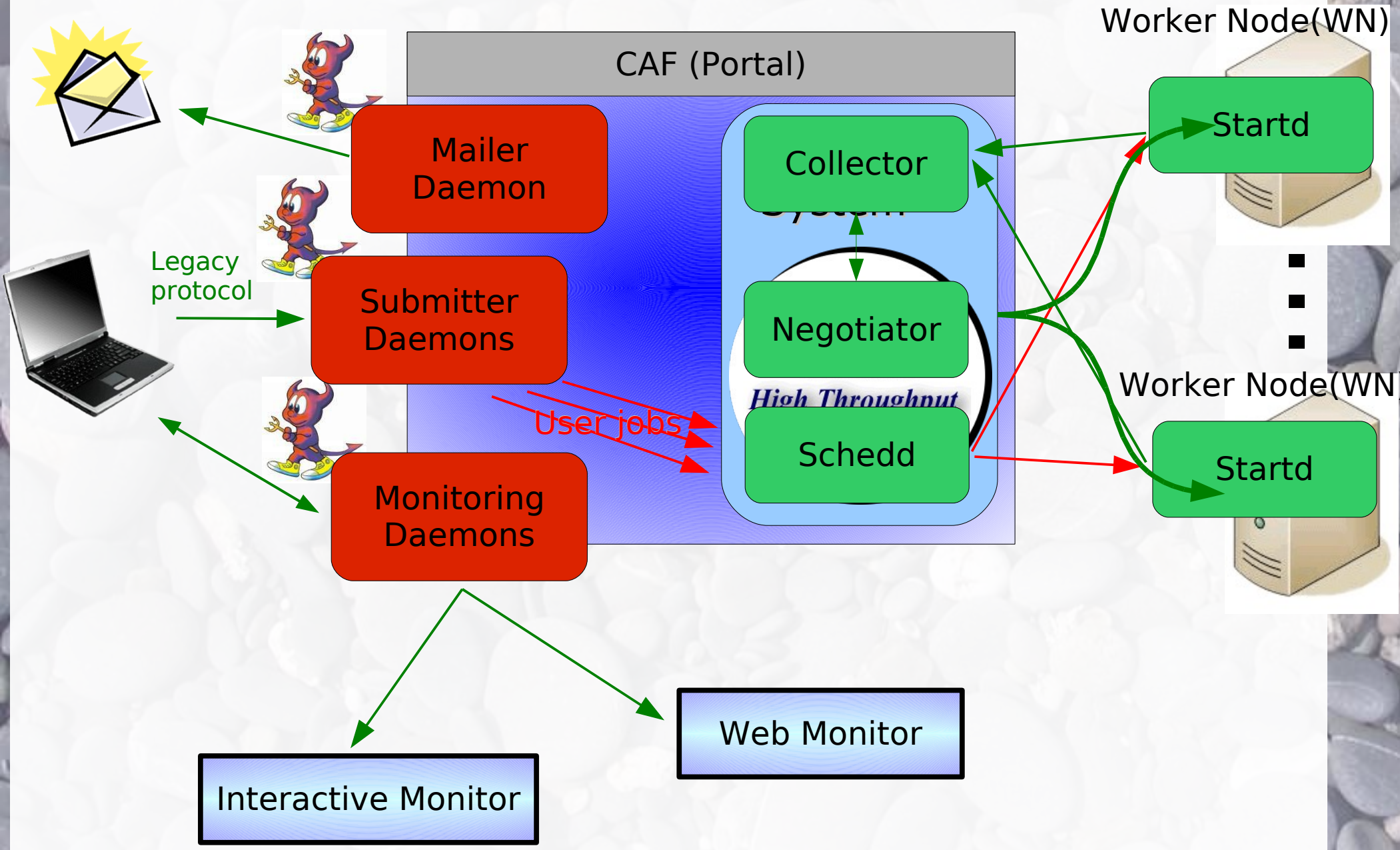
Output to any place

No need to stay connected

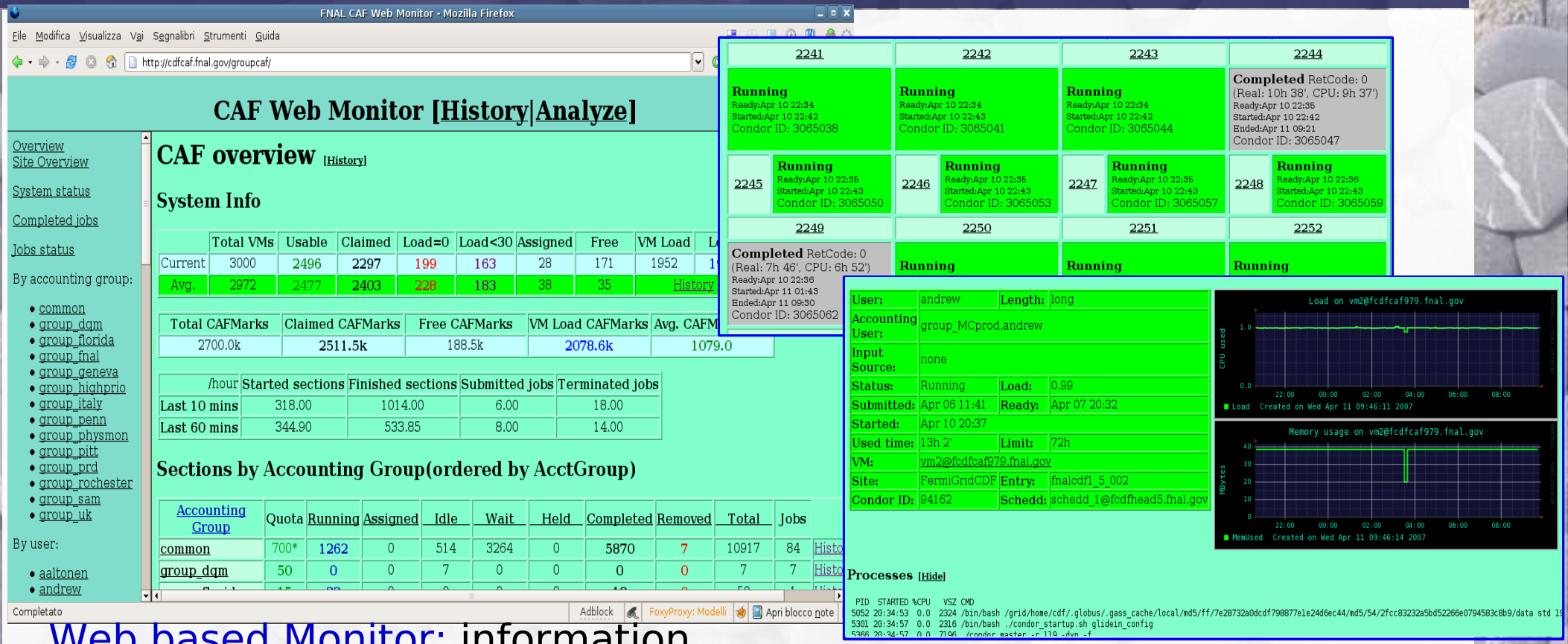
Submit and forget until receiving a mail

Developed by: H. Shih-Chieh, E. Lipeles, M. Neubauer, I. Sfiligoi, F. Wuerthwein

The CAF Headnode



Job Monitoring



Web based Monitor: information
on running jobs for all users and jobs/users history

Interactive job Monitor:
Available commands:

CafMon list CafMon kill
CafMon dir CafMon tail
CafMon ps CafMon top

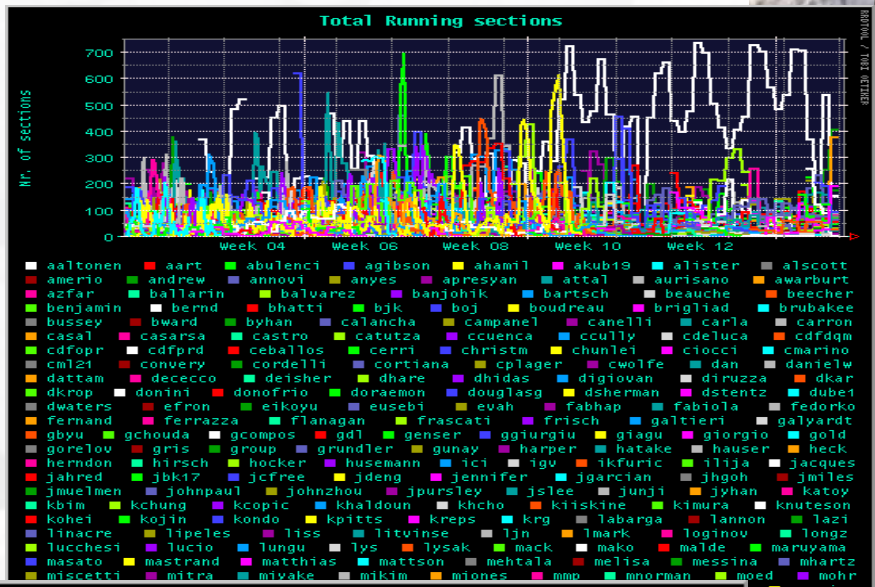
```
$ CafMon jobs

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%
$
```

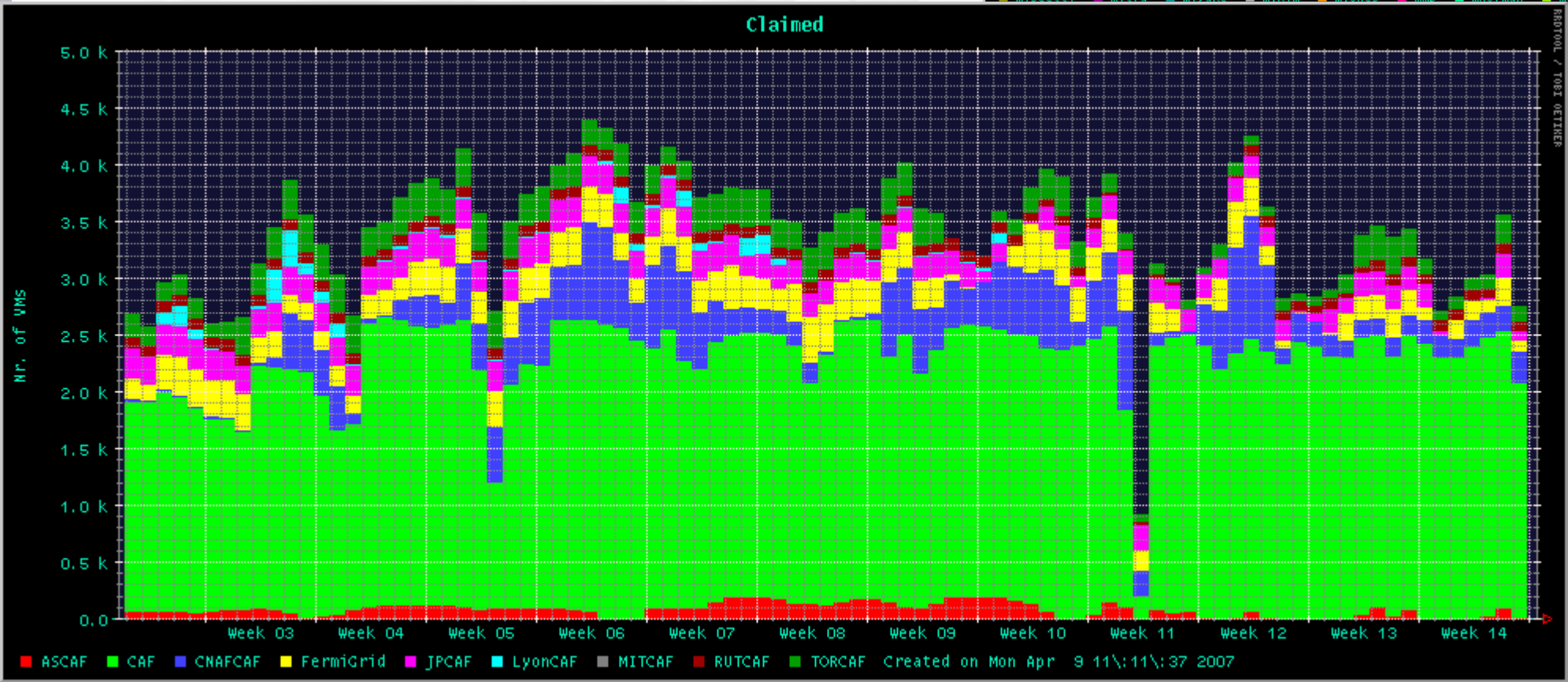
CDF batch computing in numbers

Up to 4000 batch slots (last 3 months)

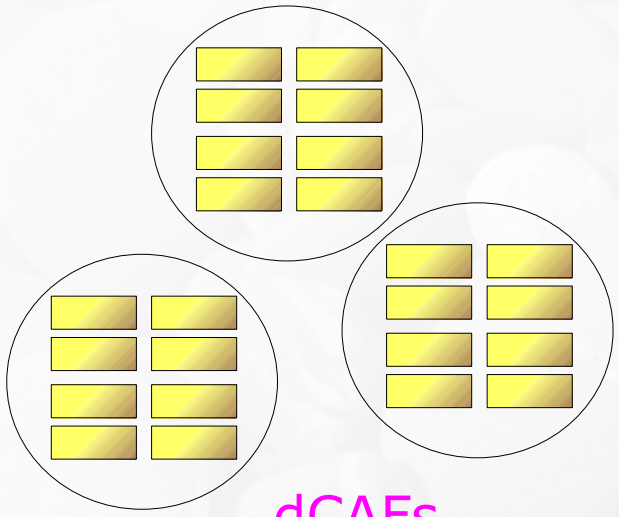
Almost 800 registered users,

Approx. 100 active at any given time

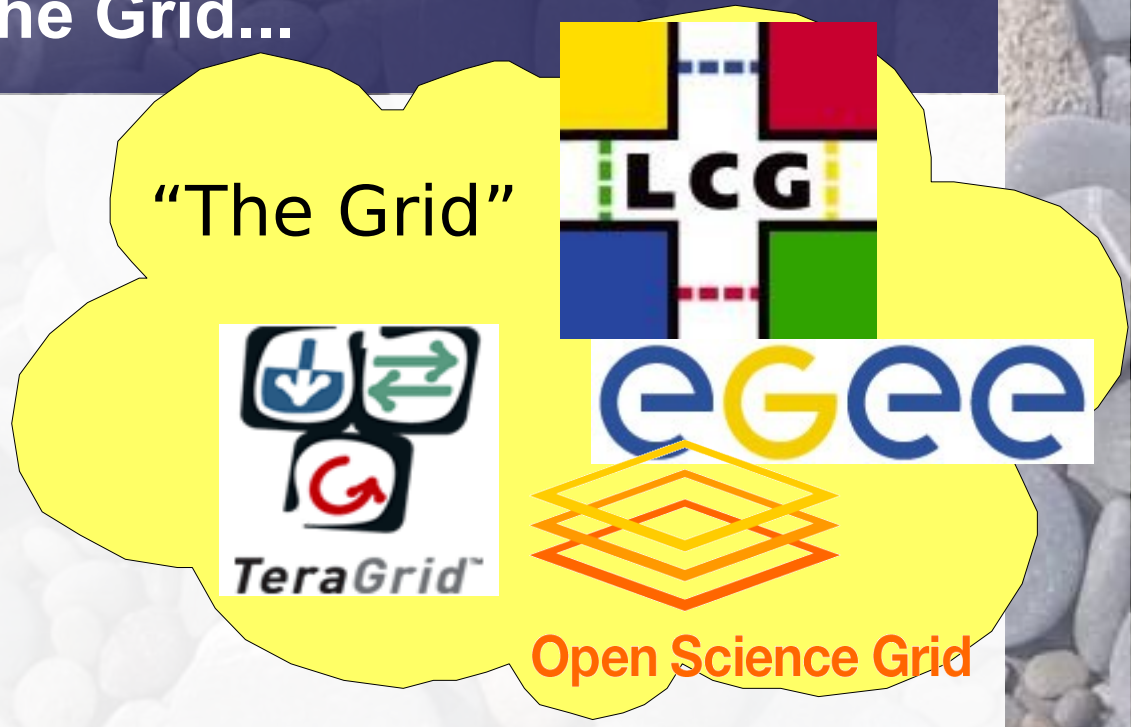




Moving to the Grid...



dCAFs



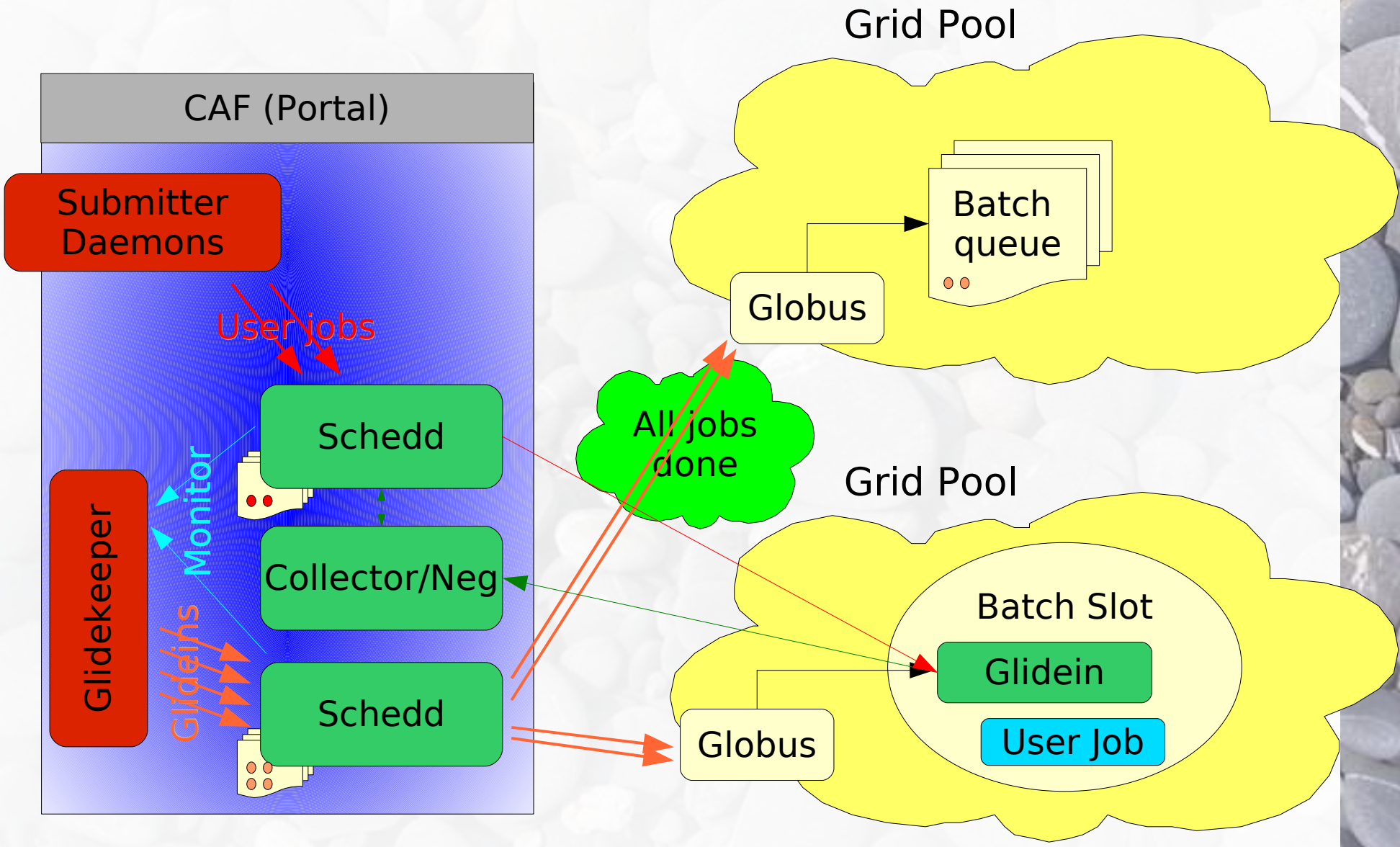
- Need to expand resources, luminosity expect to increase by factor ~4 until CDF stops taking data.
- Resources were in dedicated pools: limited expansion and need to be maintained by CDF personnel
- As a first step, we can move MC jobs to the Grid (no data access...)
- Keep on using CAF (and CNAF) for data analysis.
- CDF developed 2 portals to the GRID:
  - NamCAF for OSG and LcgCAF for LCG

NamCAF

Condor Glidein based CAF

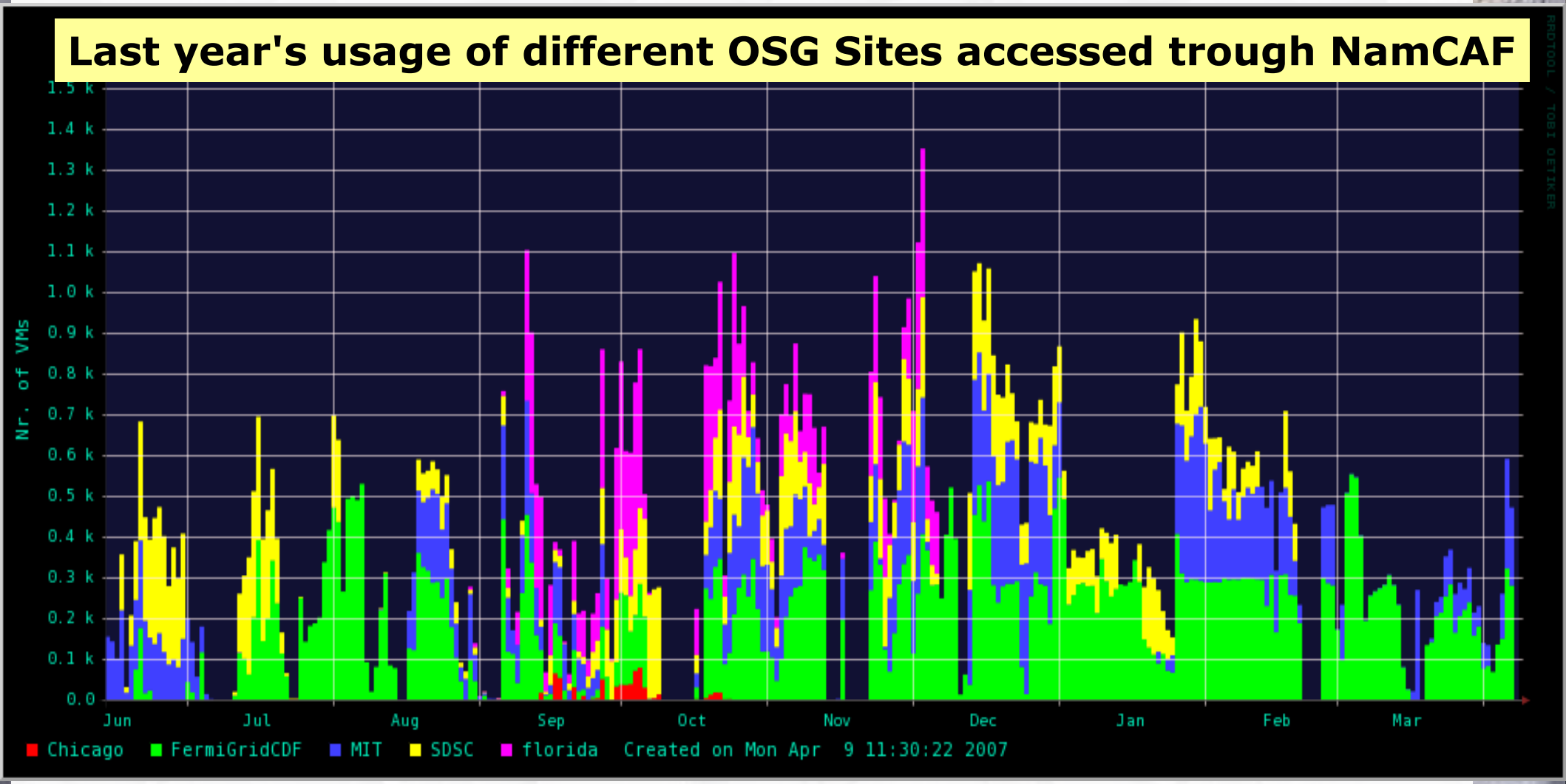
More info at:
http://cdfcaf.fnal.gov
http://cdfcaf.fnal.gov/namcaf

CDF use of Condor glide-ins



By I. Sfiligoi, S. Sarkar

CDF usage of OSG resources per site



- ...Up to 1500 batch slots
- Self contained tarballs
- No specific support on Grid sites

http://cdfcaf.fnal.gov/namcaf/

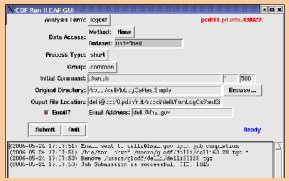
LcgCAF

CDF Submission Portal to LCG

More info at:
http://www.ts.infn.it/cdf-italia/public/offline/lcgcaf.html

LcgCAF: General Architecture

User Desktop



CDF user must have access to CAF clients and Specific CDF certificate

job

cdf-ui ~ cdf-head

Monitor

Mailer

Submitter

job

WMS

job

CafExe

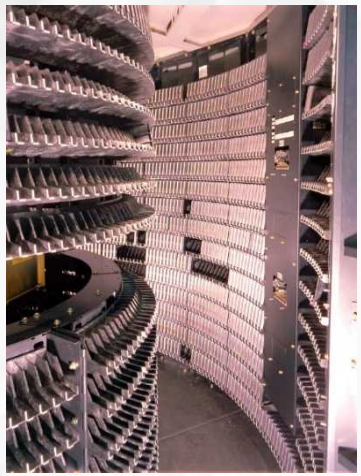
GRID site

GRID site

output

CDF-SE

output



Robotic tape Storage

Developed by: F.Delli Paoli, D.Jeans, D.Lucchesi, S.Sarkar, I.Sfiligoi

Job Submission and Execution

kx509: transforms user's CDF kerberos ticket to valid Grid x509 proxy certificate

Kdispenser service: sends kerberos ticket to worker node, needed to rcp output to CDF data servers

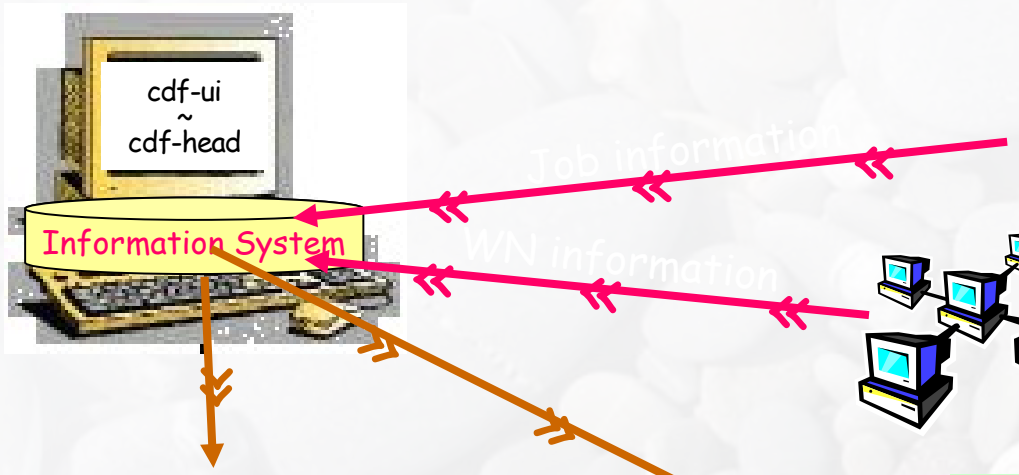
Submitter, completely rewritten for lcg tools:

- internally enqueue user job, then send to WMS
- create job wrapper to run the job on GRID environment
- make available user tarball

Job wrapper:

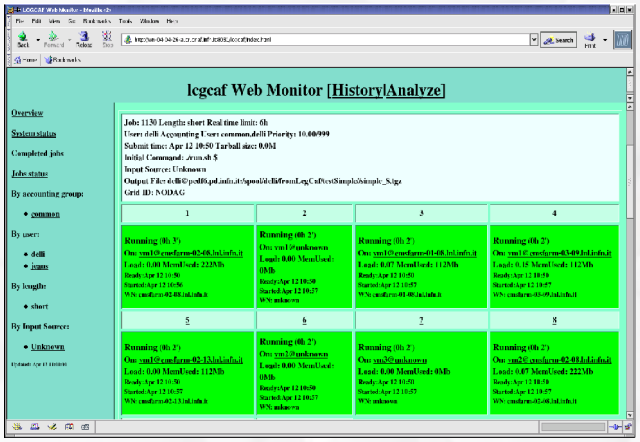
- discover site and choose correct configuration/proxy
- copy user tarball from http server to WN
- run user job and fork monitoring processes
- copy whatever is left at the end of user code to user specified location

Job Monitor



WMS: keeps track of job status on the GRID

WN: job wrapper collects information like load dir, ps, top..



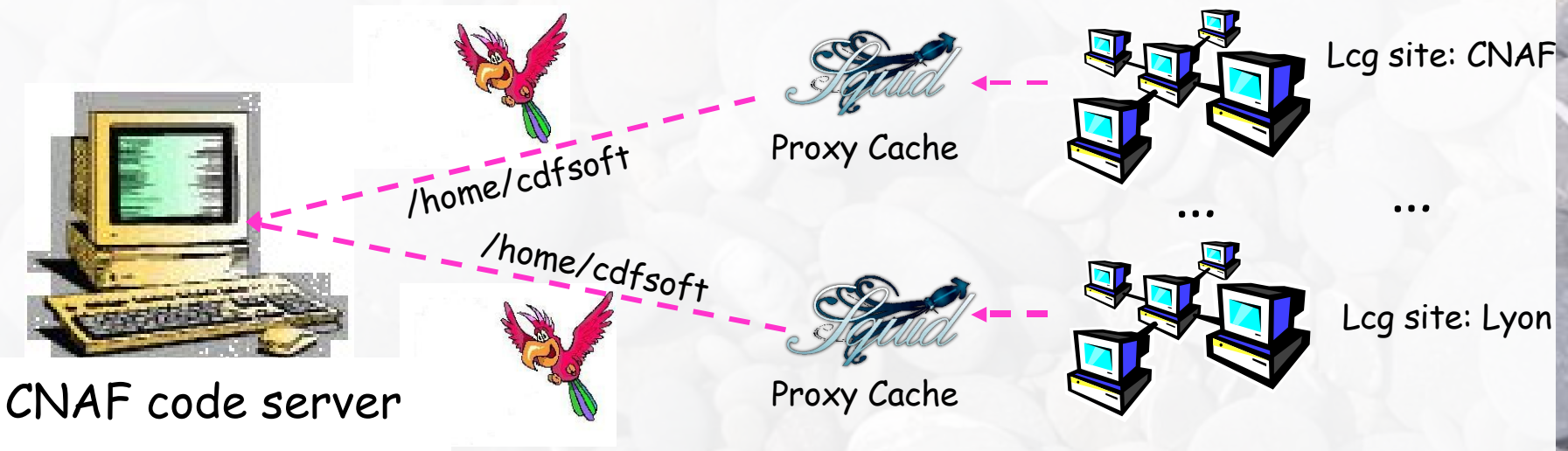

```
$ CafMon jobs

Analysis Farm: lcgcaf     Host: pcdfl1.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%
$
```

Web based Monitor: information on running jobs for all users and jobs/users history are read from LcgCAF Information System

Interactive job Monitor: running jobs information are read from LcgCAF Information System and sent to user desktop. Available commands:

CafMon list CafMon kill
CafMon dir CafMon tail
CafMon ps CafMon top

CDF Code Distribution



Parrot is used as virtual file system to distribute CDF code:

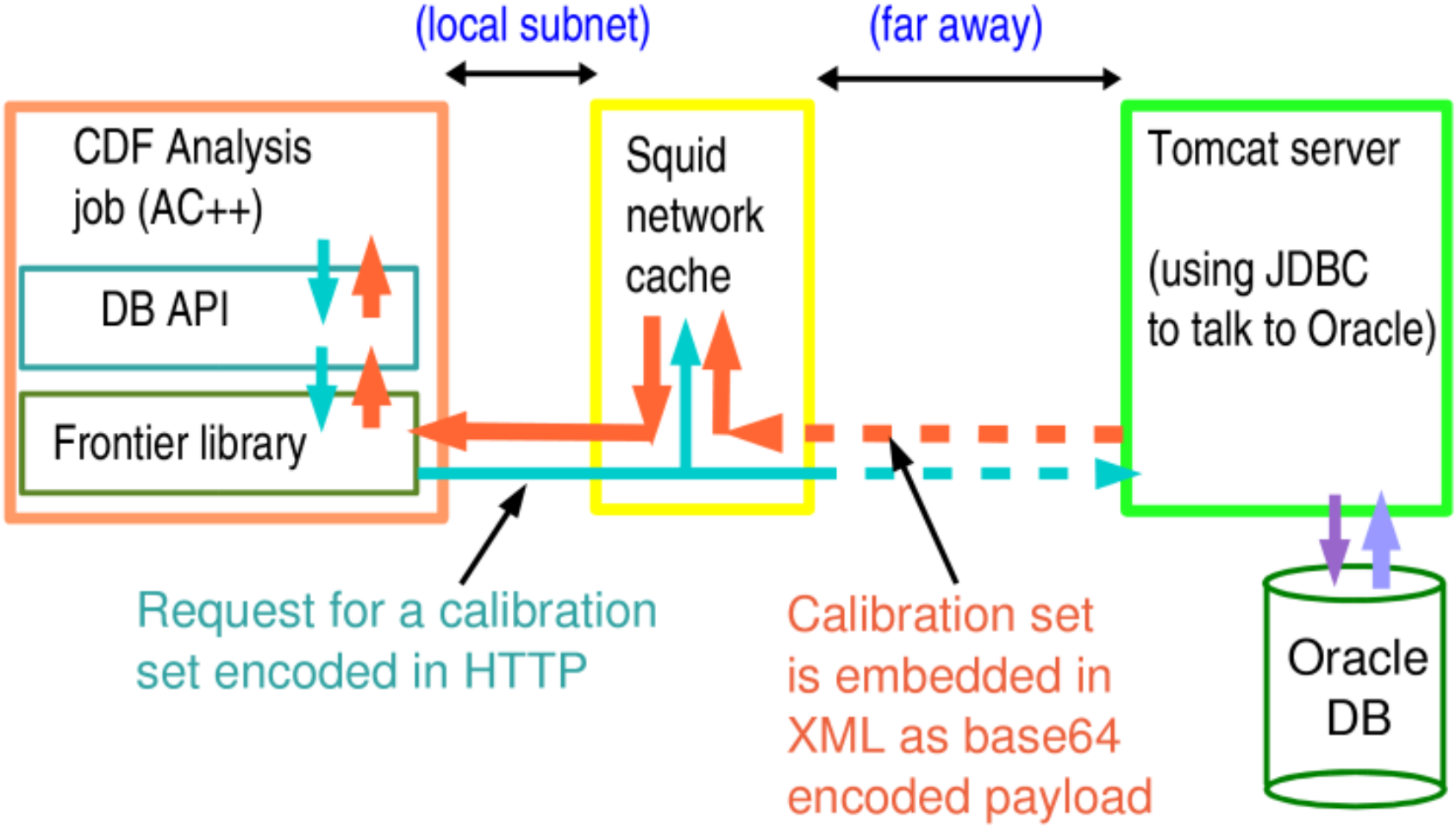
It's just a wrapper: when a user job needs a file/library not available to the worker node, parrot puts the job on hold and downloads the required file from our code server

- No specific software requirement on site
- Only the portions of CDF code that are really needed by the job are transferred to the worker node
- No need to choose in advance what to transfer

To have good performances with Parrot, a local Squid web proxy cache is also required

Frontier

Run Condition DB distribution for data Analysis and MC production

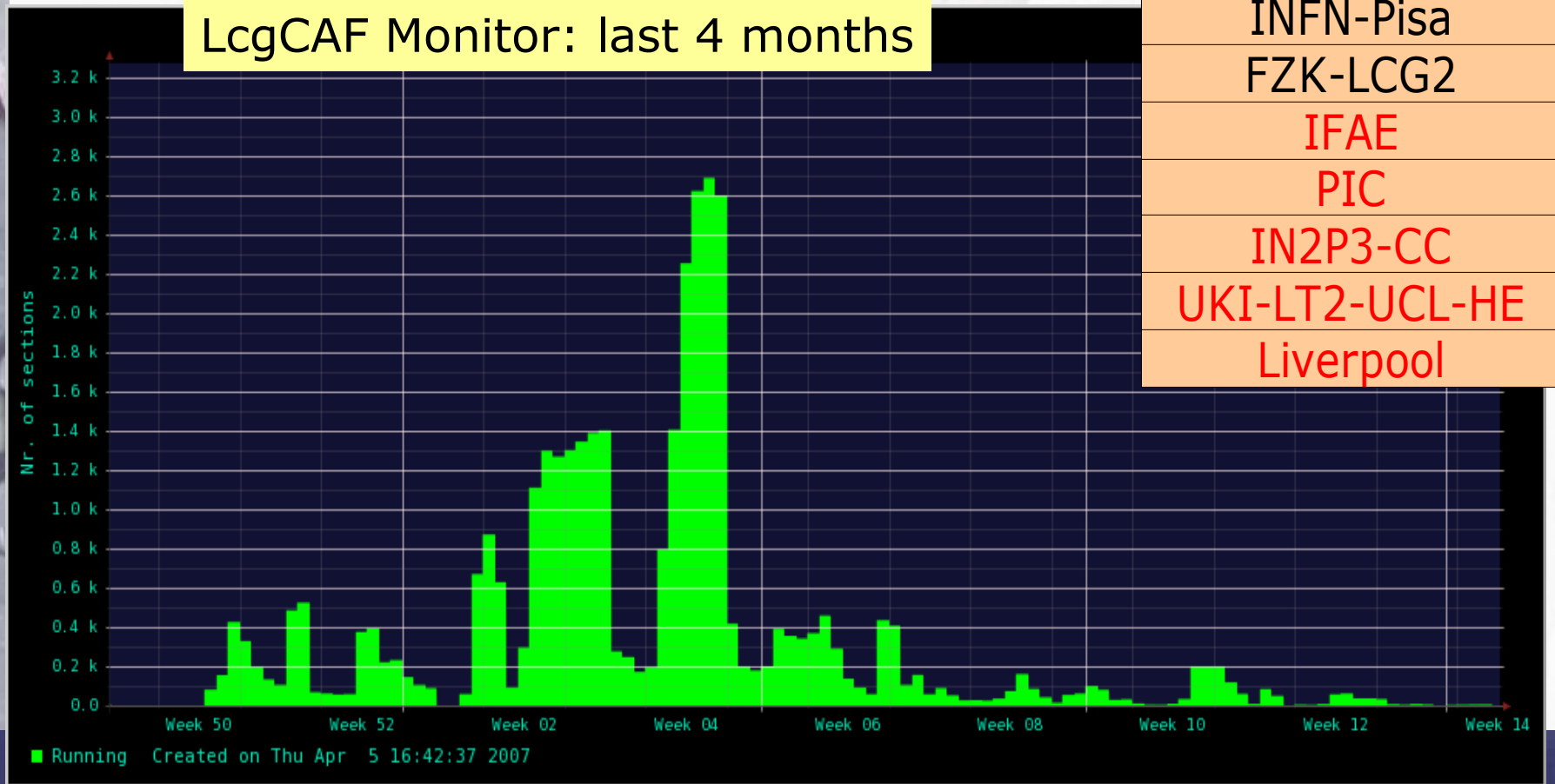


- Prevent central DB overloads
- Improvements up to 60% in certain type of jobs

LcgCAF Usage

- LcgCAF opened to all users and officially supported by CDF since November 2006
- European CDF portal to LCG resources
- Currently an average of 10 users running, peaks of more than 2500 running sections

LcgCAF Monitor: last 4 months



Sites accessed trough LcgCAF

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

LcgCAF performances and issues

Performances:

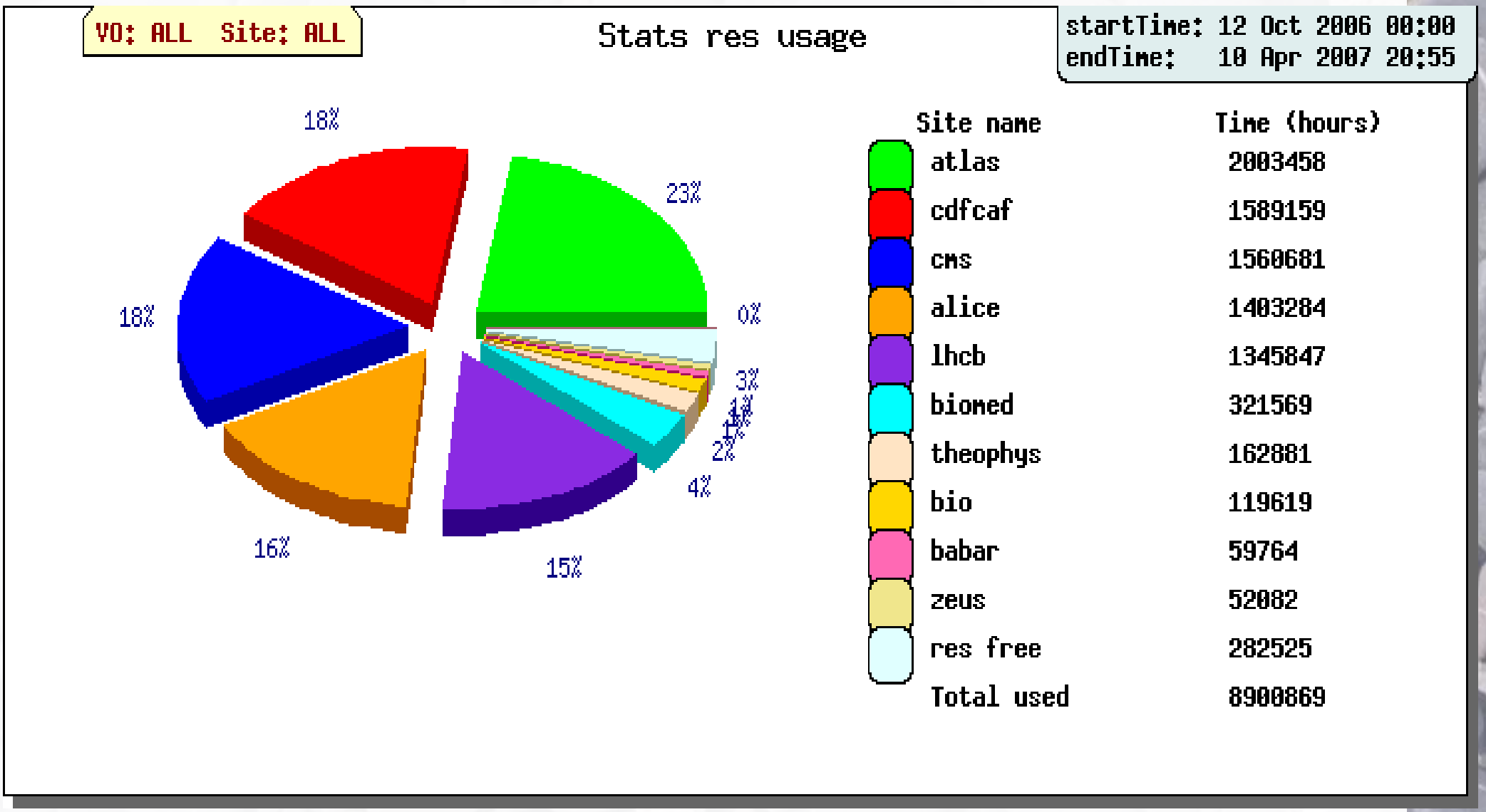
efficiency ~95% using wms 3.0 (discarding temporary sites inefficiencies)

Issues:

- wms 3.0 bugs and stability problems
- problems with matchmaking, logging and bookkeeping
- misconfigured sites, ex. clock problem: if date and time are not correct, kerberos authentication doesn't work, so it's impossible to move job output to finale destination

Now testing WMS 3.1

CDF Usage of LCG resources



(European sites not included in this plot, only italian sites)

Conclusions

...GRID can serve also a running experiment!

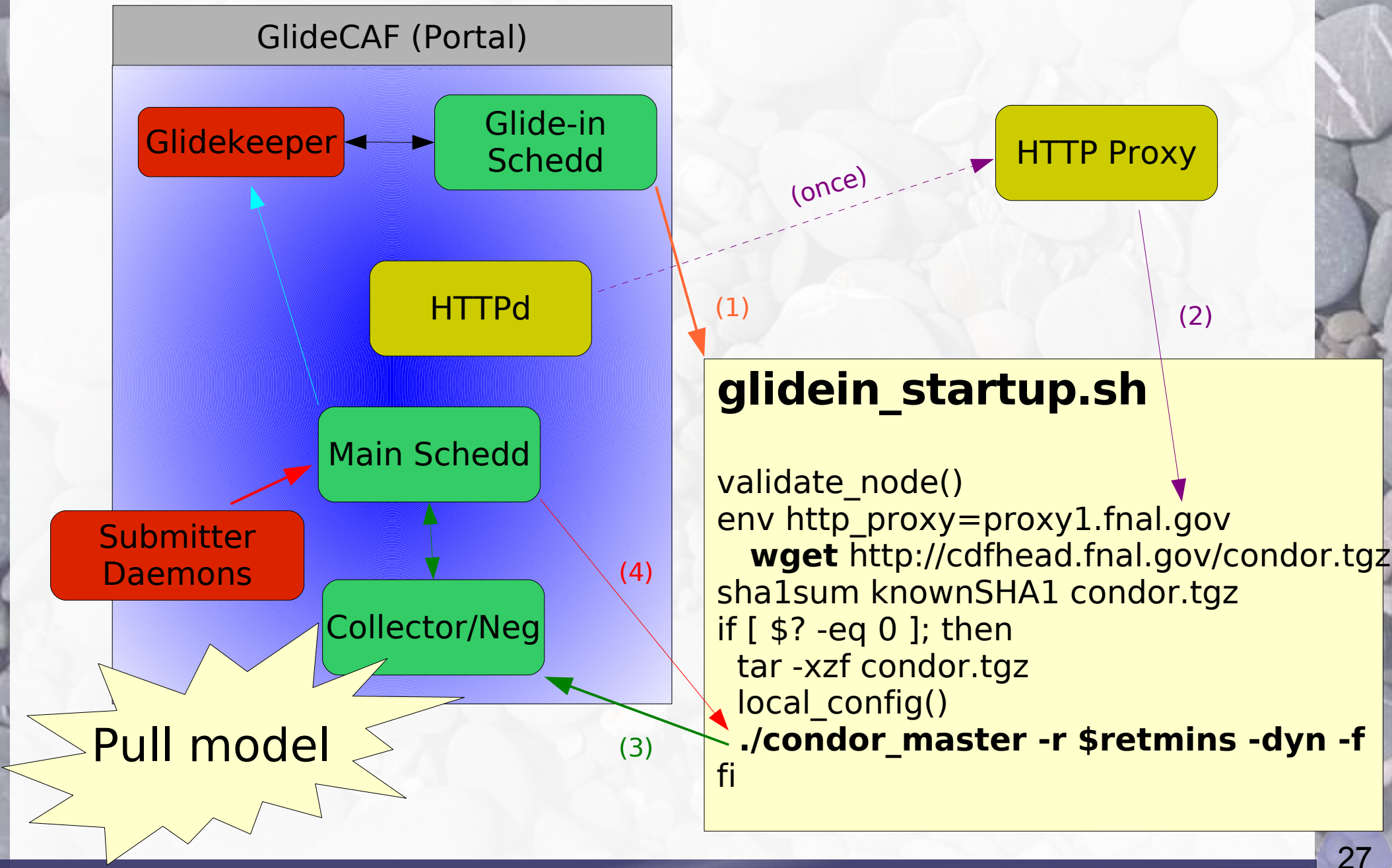
- CDF proved to have an adaptable, expandable and succesfull computing model
- Great effort has been put on exploiting GRID:
  - Used standard and common tools (http + squid)
  - No specific software requirement on sites
  - Resources are used oportunistically
  - Nothing changes from the user point of view
- MC production is completely moving towards distributed systems

Future Developments

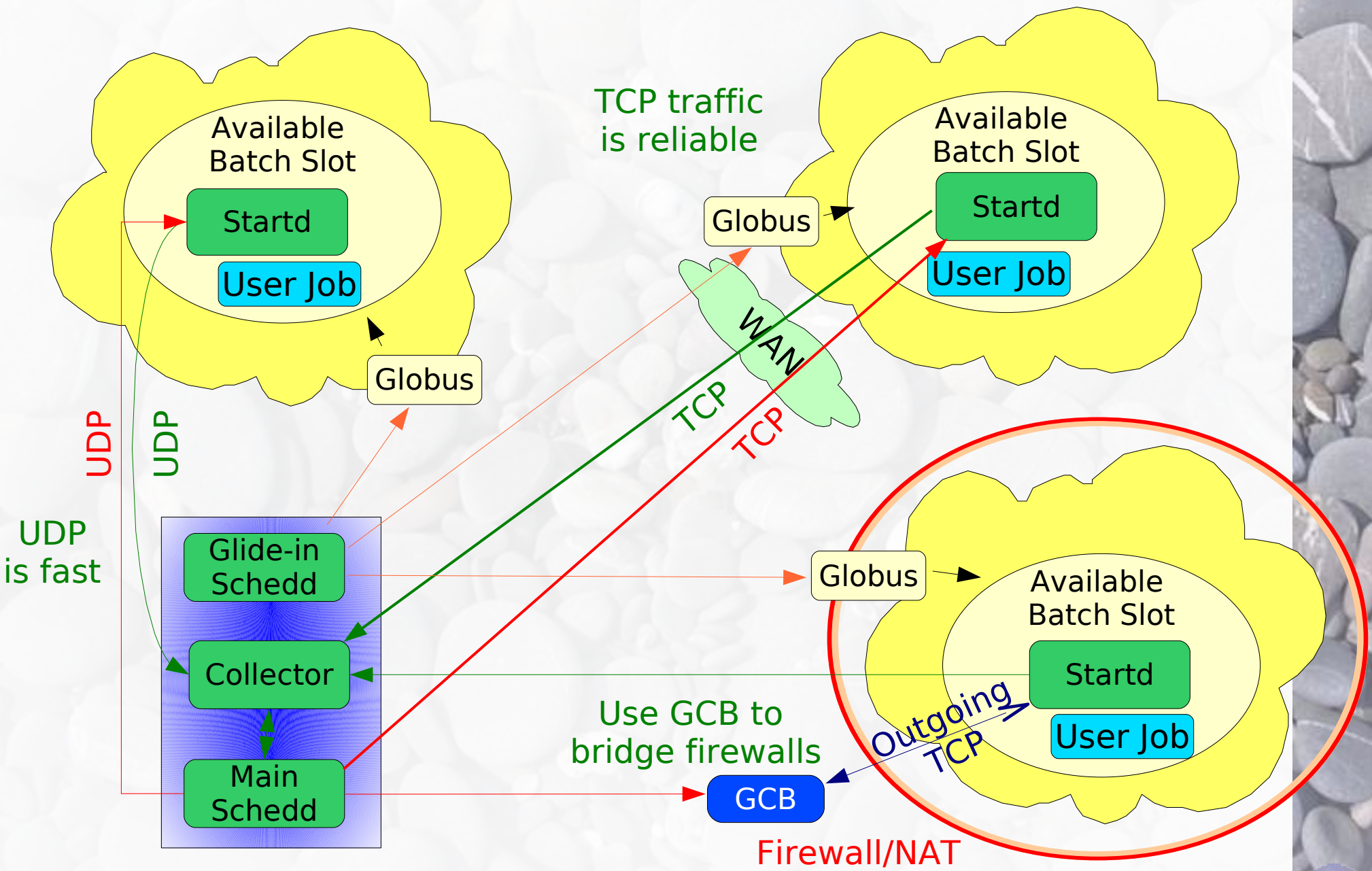
- Plan to add more resources into NamCAF and LcgCAF
- Plan to implement a mechanism of policy management for LcgCAF on top of GRID policy tools:
  - now testing GPBox for policies enforcement at CE level
  - waiting for DGAS: we need good accounting to be able set priorities dinamically
- Developed SAM-SRM2 interface to allow:
  - output file declaration into catalogue directly from LcgCAF
  - automatic transfer of output files from local CDF storage to FNAL tape
  - waiting for a SRM2 compliant storage element
- Explore the possibility of running on data moving jobs towards sites that can serve the desired files

Backup

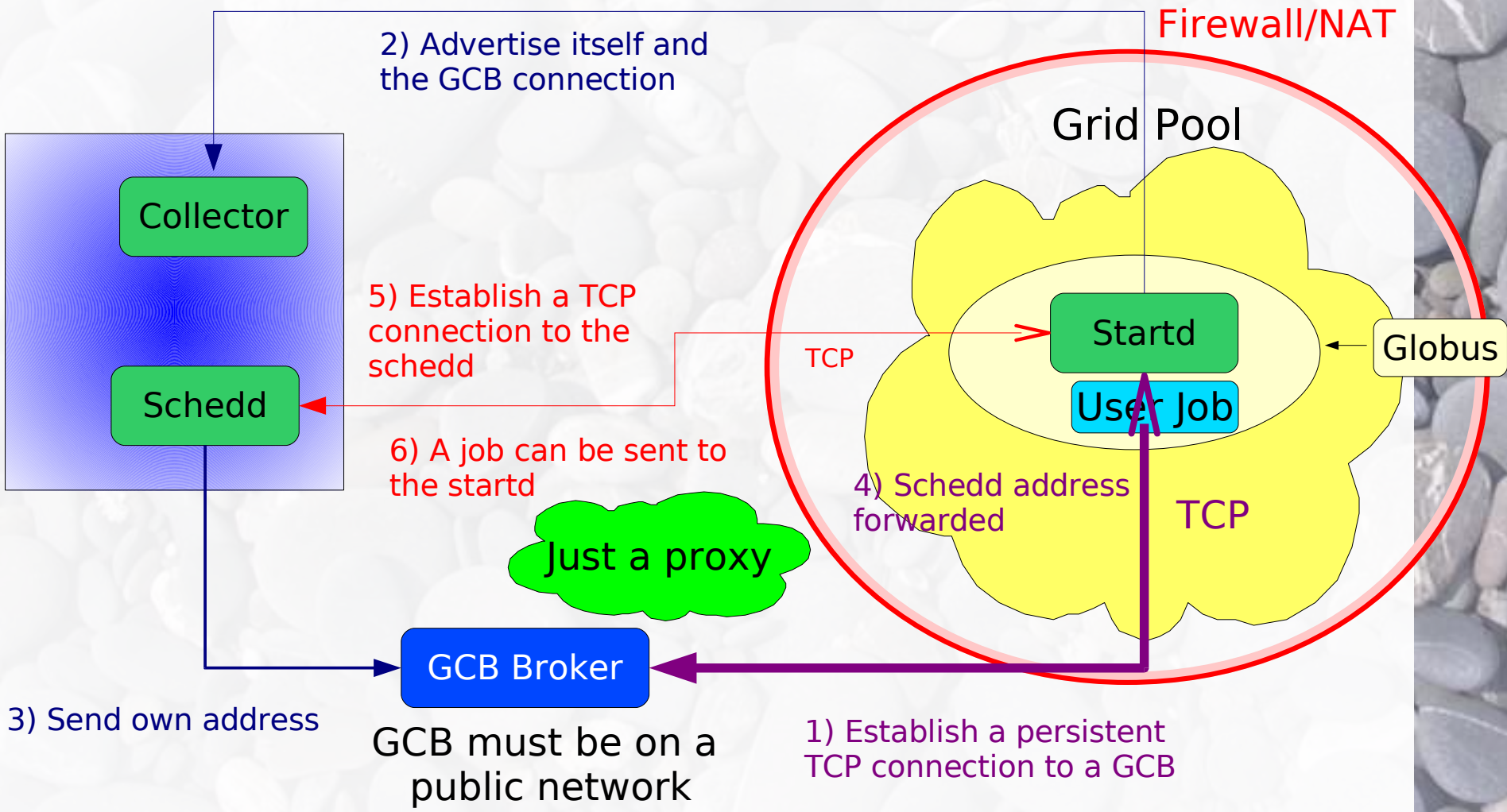
GlideCAF: Condor Binaries Transfer



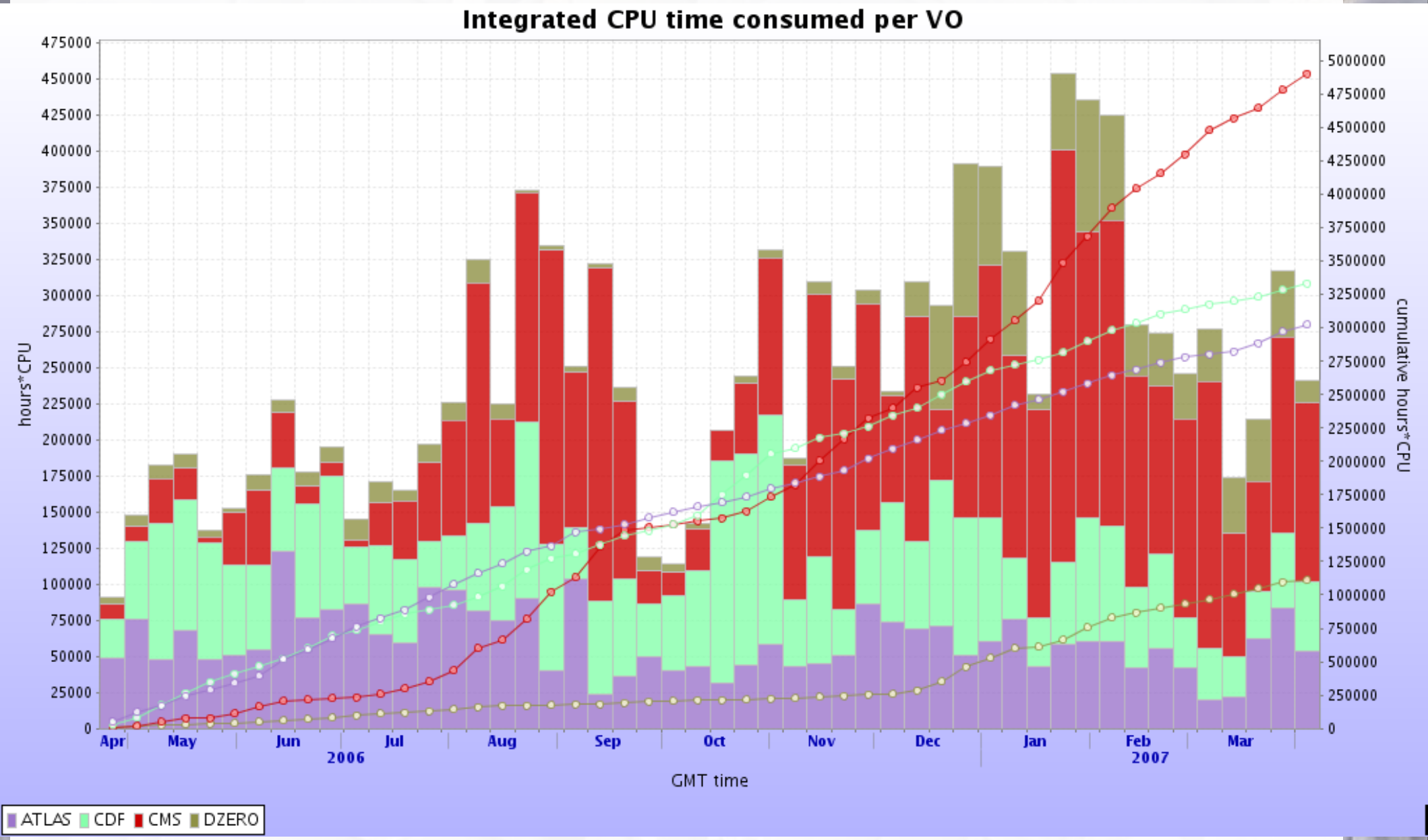
NamCAF: Working with remote Grid sites



NamCAF: How GCB works



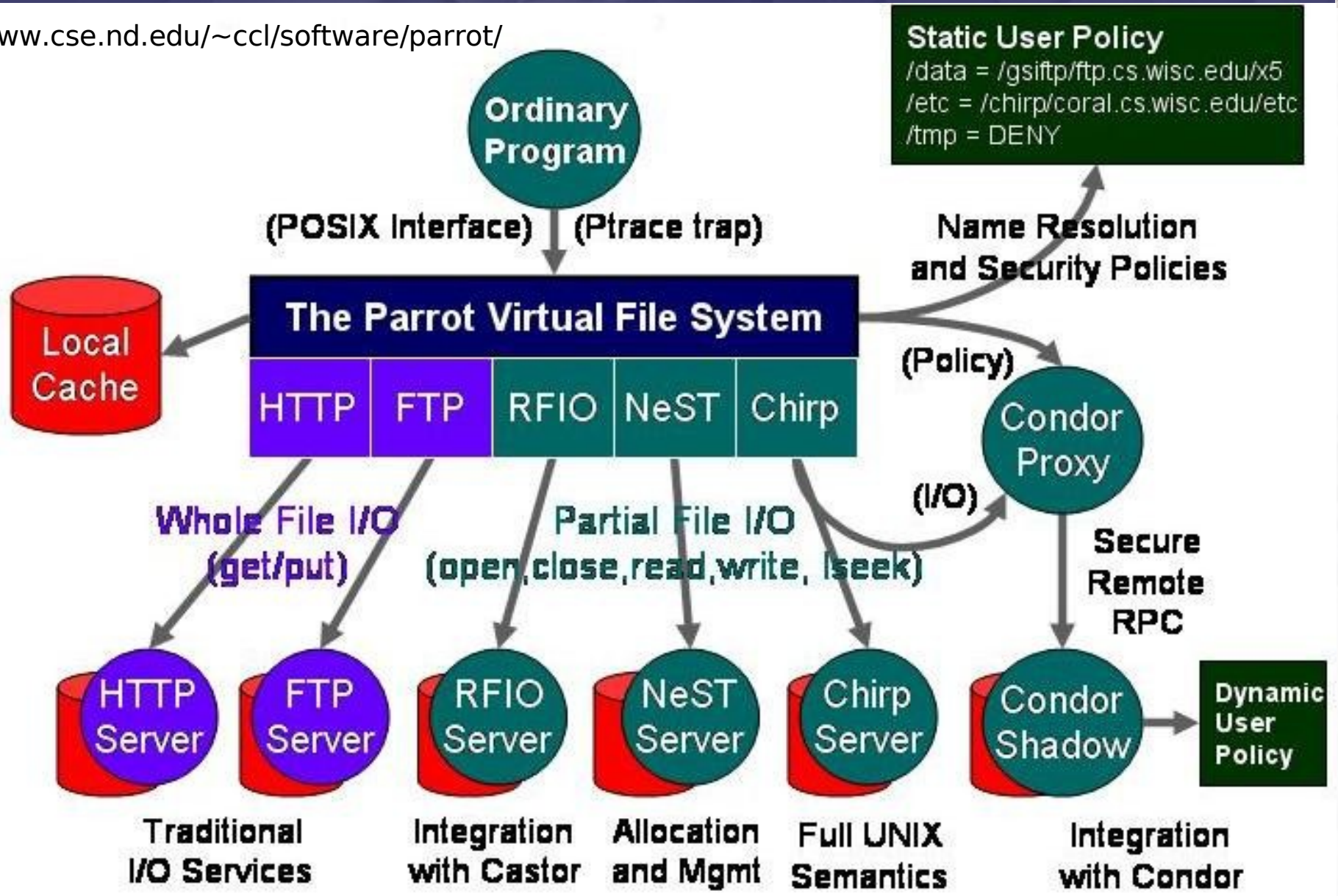
CDF usage of OSG resources



http://grid02.uits.indiana.edu:8080/show?page=index.html

Overview of Parrot

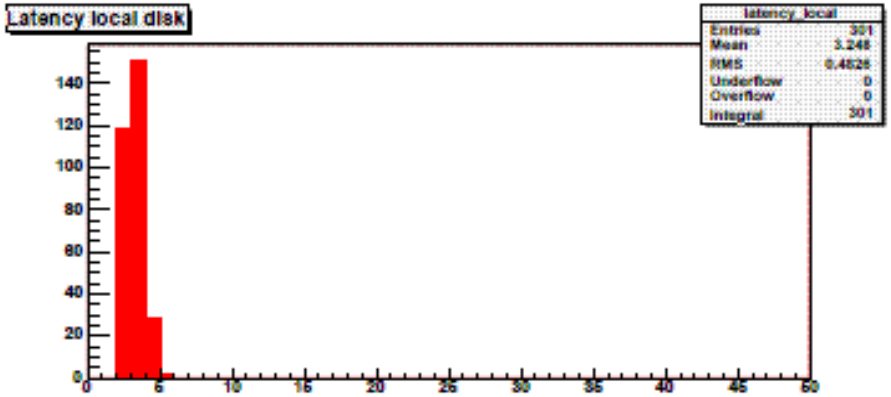
http://www.cse.nd.edu/~ccl/software/parrot/



Invoking Parrot on the WN is as easy as:

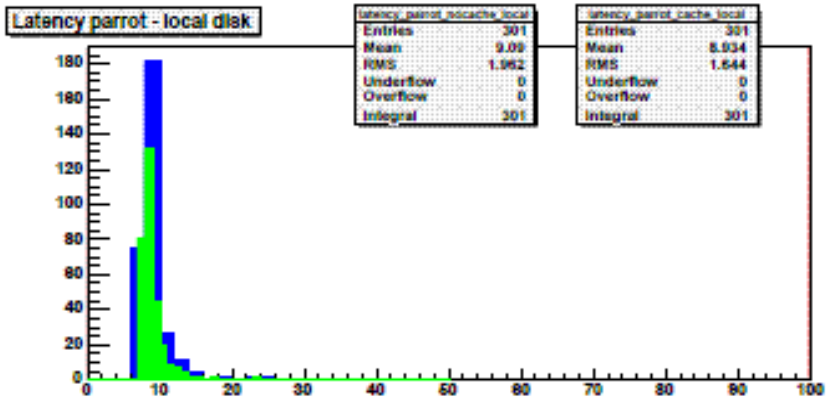
./parrot -M parrotMountPoints -p <squid_proxy> ./userJob.sh

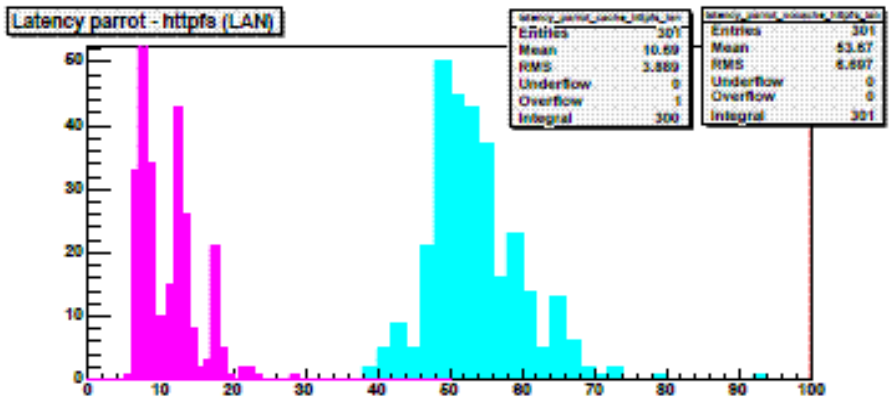
Parrot Performances with Http in a CDF-like job

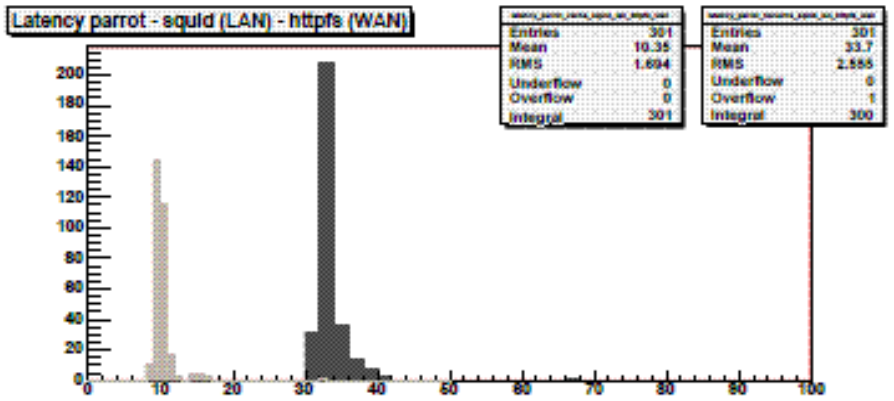


latencyTime = reatTime - (userTime + sysTime) / ncpu

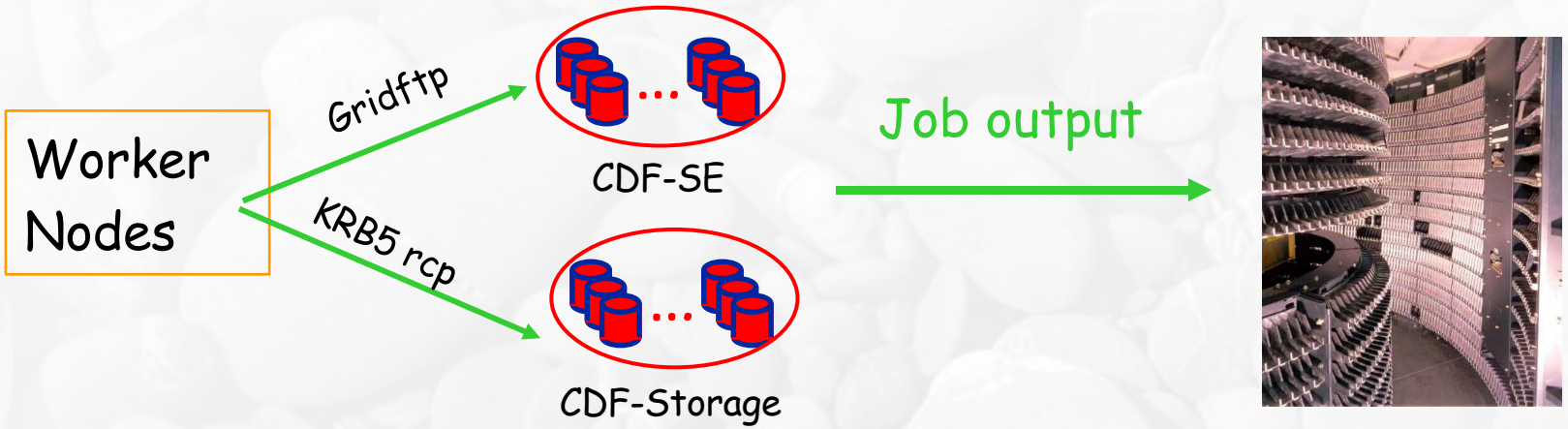
- local
- parrot w/o cache - local
- parrot w/ cache - local
- parrot w/o cache - httpfs (LAN)
- parrot w/ cache - httpfs (LAN)
- parrot w/o cache - httpfs (WAN)
- parrot w/ cache - httpfs (WAN)
- parrot w/o cache - squid (LAN) - httpfs (WAN)
- parrot w/ cache - squid (LAN) - httpfs (WAN)







LcgCAF Output Storage



- User job output is transferred to CDF Storage Element (SE) via gridftp or to a CDF-Storage location via rcp.
- From CDF-SE or CDF-storage output copied using gridftp to FNAL fileservers.
- Automatic procedure copies files from fileservers to tape and declares them to the CDF catalogue.
- Planning to use a SRM-SE interface for big sites.

LcgCAF Performances

CDF B Monte Carlo production
by Power Users in July 2006: ~5000 jobs in ~1 month

Samples	Events on tape	ε	ε with 1 Recovery
B_s->$D_s\pi$ D_s->$\phi\pi$, D_s-> K*K, D_s->K_sK	~6,2 · 10^6	~94%	~100%

Possible Failures:

- GRID: site miss-configuration, temporary authentication problems, WMS overload, LB "loose" informations
- LcgCAF: Parrot/Squid cache stacked (Parrot cache coherency problems are under investigation)

Automatic Retry mechanism:

- GRID failures: WMS retry shallow retry
- LcgCAF failures: "home-made" retry (log files parsing) – only for certain type of jobs