Interactive Data Analysis with PROOF

Bleeding Edge Physics with Bleeding Edge Computing

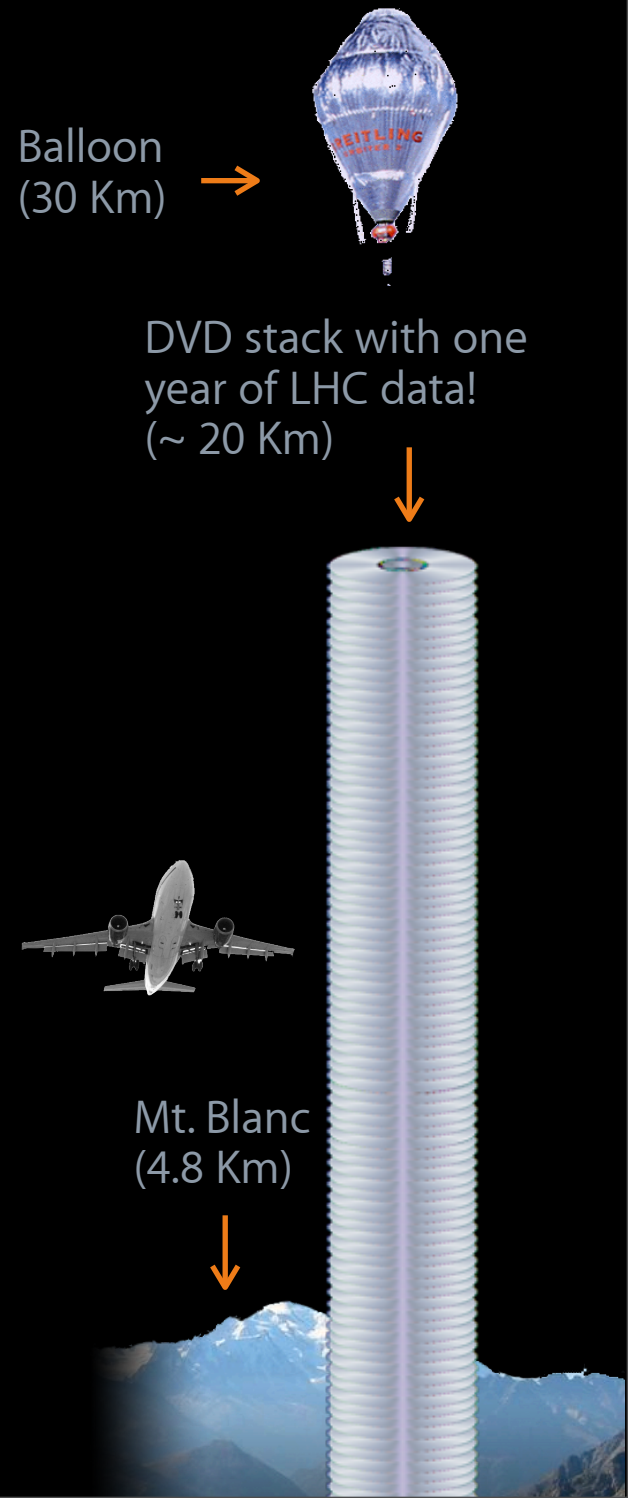
Fons Rademakers
CERN

LHC Data Challenge

- The LHC generates:
  - 40 million collisions per second
- Combined the 4 experiments record:
  - After filtering, 100 interesting collision per second
  - From 1 to 12 MB per collision ⇒ from 0.1 to 1.2 GB/s
  - 10^{10} collisions registered every year
  - ~ 10 PetaBytes (10^{15} B) per year
  - LHC data correspond to 20 millions DVD's per year!
  - Computing power equivalent to 100.000 of today's PC
  - Space equivalent to 400.000 large PC disks

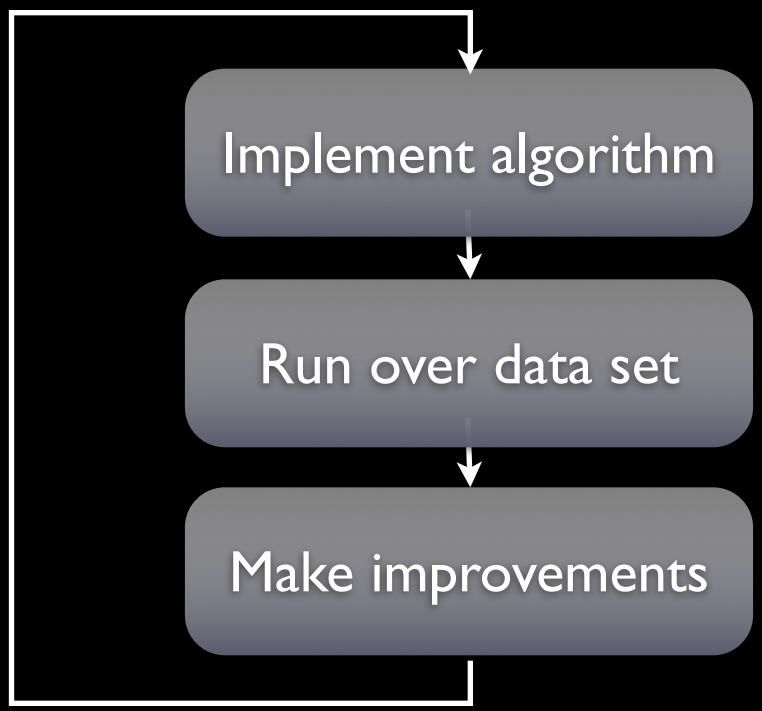
LHC Data Challenge

- The LHC generates:
  - 40 million collisions per second
- Combined the 4 experiments record:
  - After filtering, 100 interesting collision per second
  - From 1 to 12 MB per collision ⇒ from 0.1 to 1.2 GB/s
  - 10^{10} collisions registered every year
  - ~ 10 PetaBytes (10^{15} B) per year
  - LHC data correspond to 20 millions DVD's per year!
  - Computing power equivalent to 100.000 of today's PC
  - Space equivalent to 400.000 large PC disks



HEP Data Analysis

- Typical HEP analysis needs a continuous algorithm refinement cycle



HEP Data Analysis

- Ranging from I/O bound to CPU bound
- Need many disks to get the needed I/O rate
- Need many CPUs for processing
- Need a lot of memory to cache as much as possible

Some ALICE Numbers

- 1.5 PB of raw data per year
- 360 TB of ESD+AOD per year (20% of raw)
- One pass using 400 disks at 15 MB/s will take 16 hours

Some ALICE Numbers

- 1.5 PB of raw data per year
- 360 TB of ESD+AOD per year (20% of raw)
- One pass using 400 disks at 15 MB/s will take 16 hours

Using parallelism is the only way to analyze this amount of data in a reasonable amount of time

Some History

- In the beginning the was PIAF (early 90's)
  - Parallel Interactive Analysis Facility
  - Parallel version of PAW's nt/plot and nt/loop commands
  - Client - Master - Worker architecture where the master pushed work to the workers

- PIAF ran on an 8 node HP 755 125MHz PA-RISC cluster
  - 128 MB RAM per node
  - 16.5 GB RAID per node (4.5MB/s)
  - FDDI (6.5MB/s) network
  - NFS shared file system

Some More History

- A small but loyal group of users from the LEP experiments
  - Usage statistics of March 1994:
    - 1.2 TB of data processed
    - 60 different users
    - Connecting from 11 different countries
    - Average speedup of 3.2 per query
- On the top of its success CN Div decided it did not want to continue with PIAF
- Rene and I left CN and started work on ROOT

Early PROOF

- First prototype developed in 1997
  - Corrected the main weakness of PIAF by using a PULL instead of PUSH architecture for better scalability
- Showed potential, but there was not a huge interest
  - ROOT's core was still being developed
  - Experiments using ROOT did not yet have a data problem
- Changed early 2000 when the RHIC experiments started collecting significant amounts of data
- Development taken up by the MIT Phobos group

Production Usage in Phobos

PROOF in PHOBOS

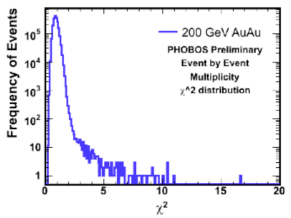
Maarten Ballintijn / MIT
maartenb@mit.edu

May 24, 2006 – Application Area Meeting

Components of the Facility

- 25 Interactive Nodes
- 425 Compute nodes w/ distributed disk
  - 100 TB disk space
  - Mix of 100Mb and 1Gbit Ethernet
- HPSS tape robot / Mass Storage System
- Centralized disk space
  - NFS (0.9 TB) – home directories, software
  - Panasas (3.8 TB) – data, proof work directories

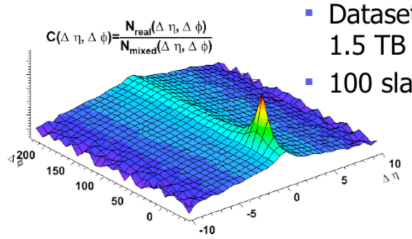
Rare high multiplicity event search



- Burak Alver
- Dataset: 11k files, 4.5 TB
- 150 slaves, ~1 hour

Two Particle Correlations @ 200GeV

Two particle correlation function of minbias dAu 200Gev



- Wei Li, Constantin Loizides
- Dataset: 4.5k files, 1.5 TB
- 100 slaves, 75 min

PROOF for LHC

- ALICE sees PROOF as strategic tool for prompt analysis on their Central Analysis Facility

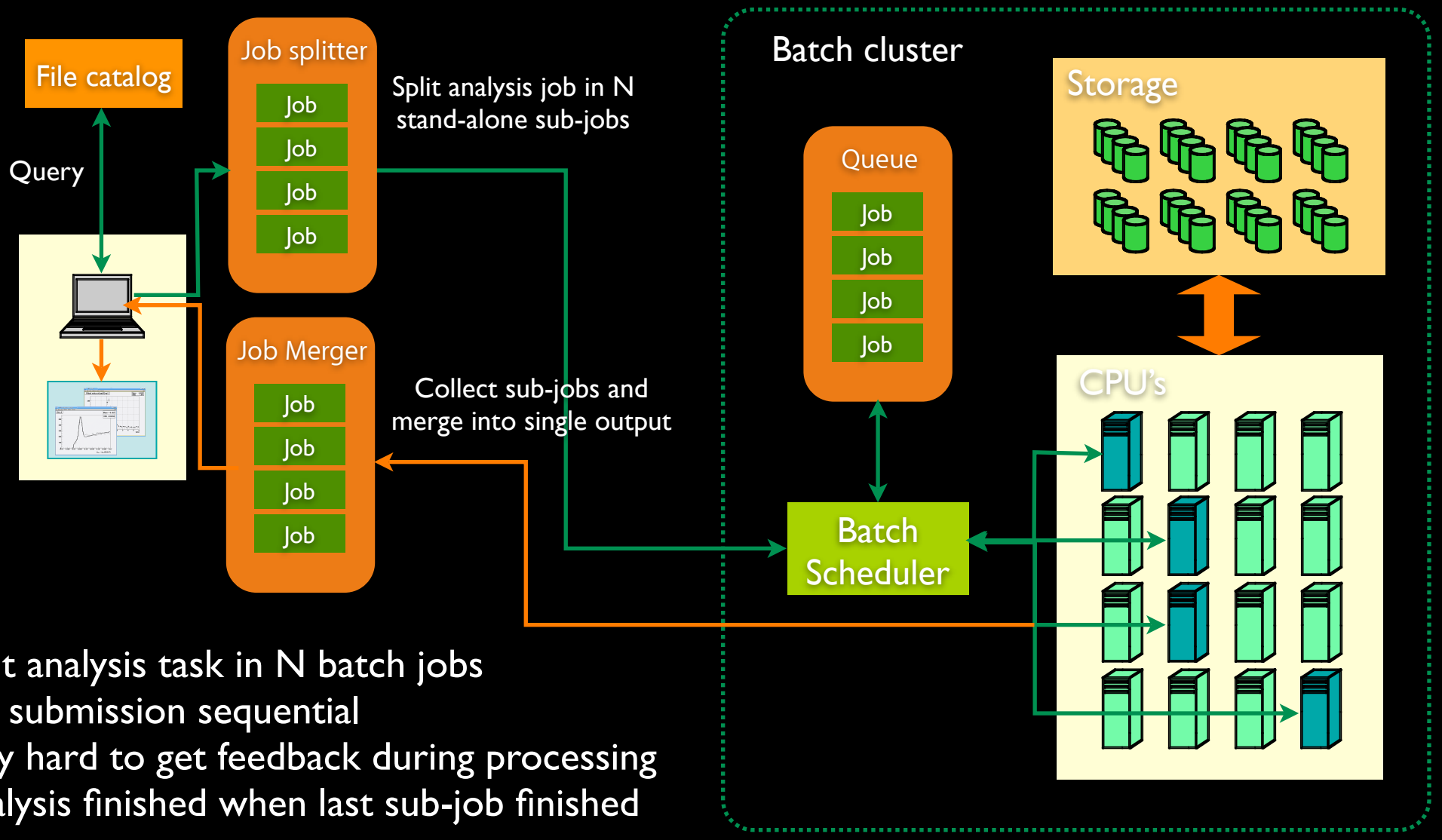
- Development taken back to CERN
  - Change of core technologies - tight integration with xrootd
  - Support for Interactive Batch
  - Need for scheduling
  - Better data set management
  - Etc, etc.

- Other LHC experiments, having their final data in bare ROOT files, are getting interested

Where Can PROOF Be Used

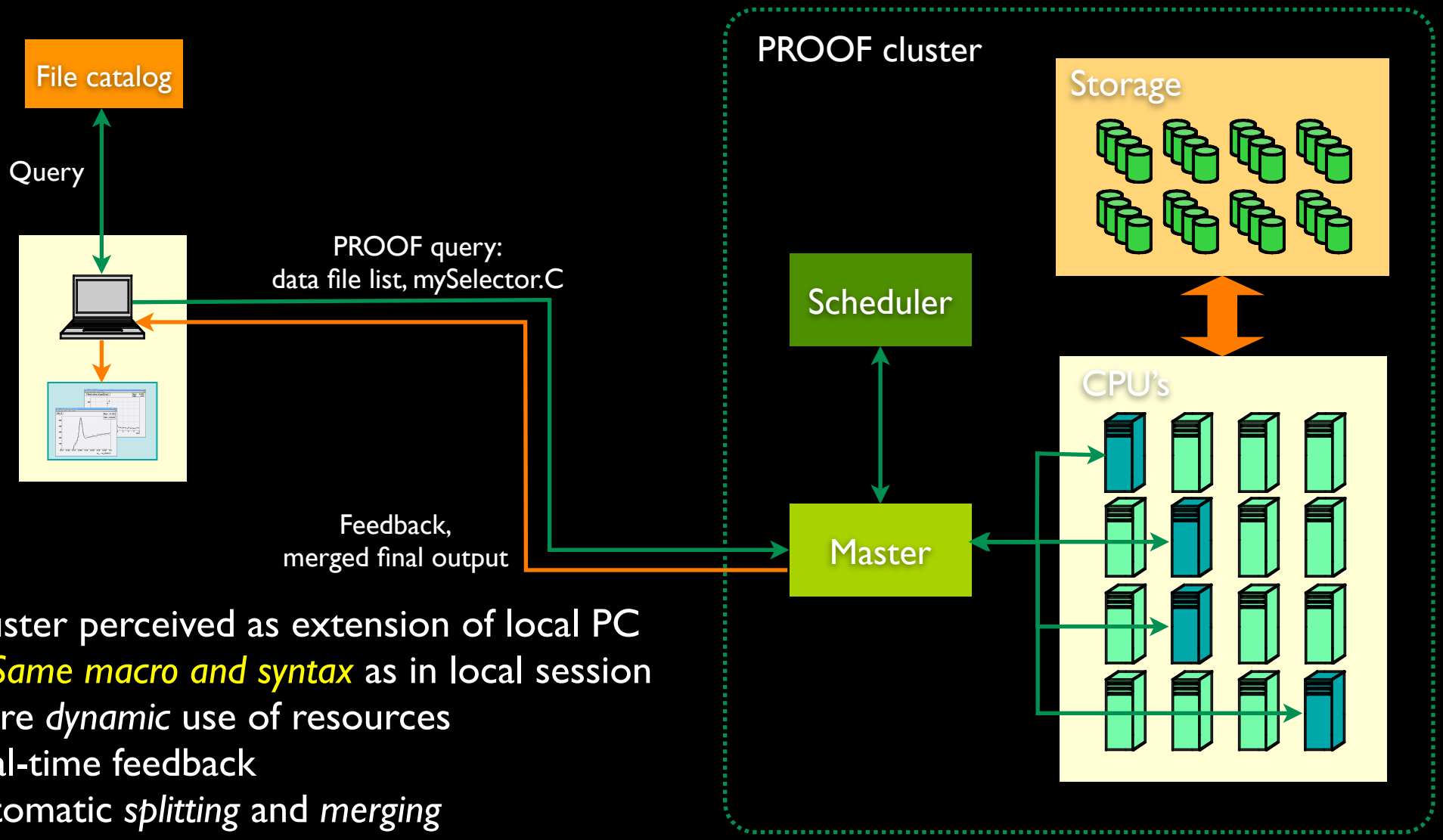
- CERN Analysis Facility (CAF)
- Departmental workgroups (Tier-2's)
- Multi-core, multi-disk desktops (Tier-3/4's)

The Traditional Batch Approach



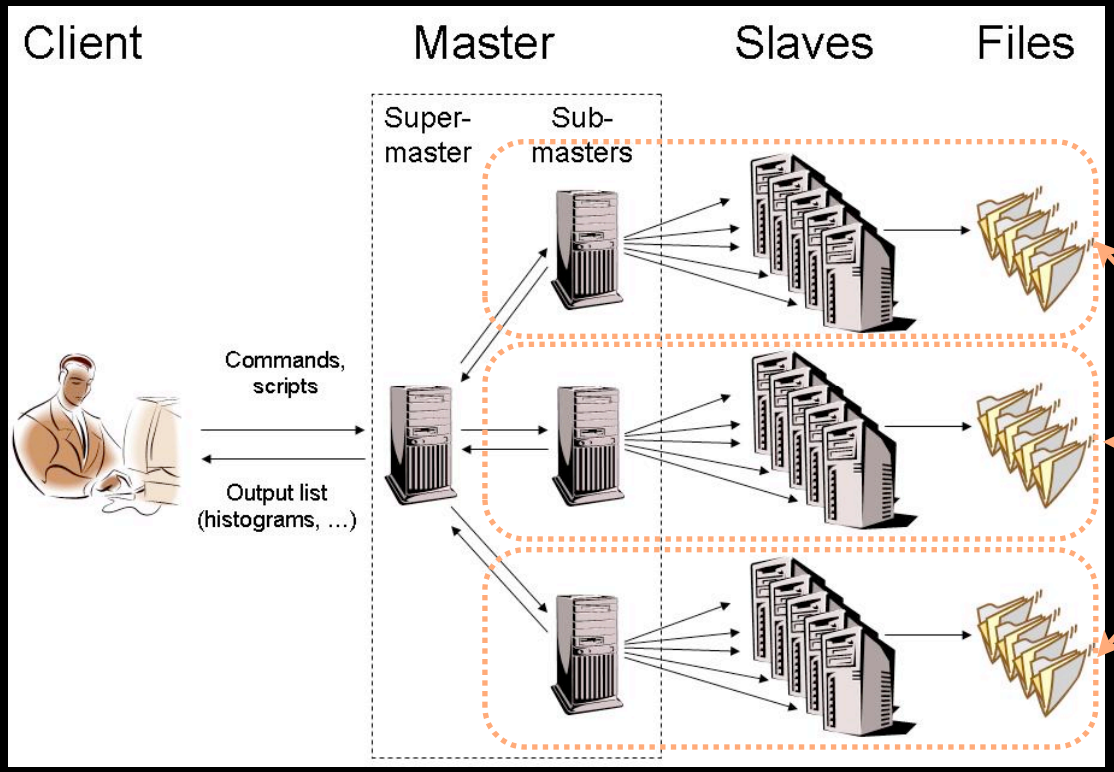
- Split analysis task in N batch jobs
- Job submission sequential
- Very hard to get feedback during processing
- Analysis finished when last sub-job finished

The PROOF Approach



- Cluster perceived as extension of local PC
  - *Same macro and syntax* as in local session
- More *dynamic* use of resources
- Real-time feedback
- Automatic *splitting* and *merging*

Multi-Tier Architecture



Adapts to wide area *virtual* clusters

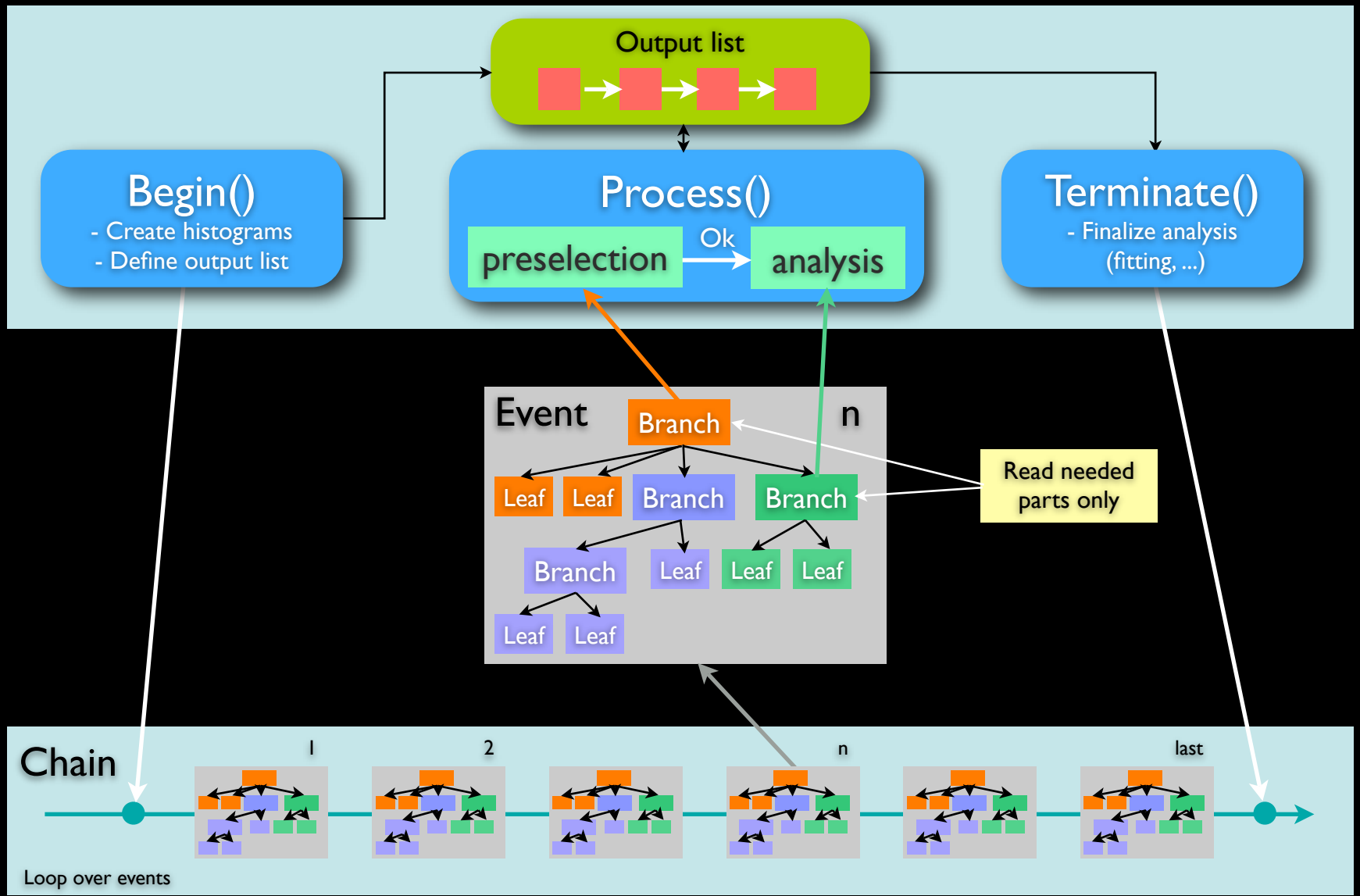
Geographically separated domains, heterogeneous machines

Network performance

Less important → VERY important

Optimize for data locality or high bandwidth data server access

The ROOT Data Model
Trees & Selectors



TSelector - User Code

```
// Abbreviated version
class TSelector : public TObject {
protected:
    TList *fInput;
    TList *fOutput;
public
    void   Notify(TTree*);
    void   Begin(TTree*);
    void   SlaveBegin(TTree *);
    Bool_t Process(int entry);
    void   SlaveTerminate();
    void   Terminate();
};
```

TSelector::Process()

```
...
...
// select event
b_nlhk->GetEntry(entry);          if (nlhk[ik] <= 0.1)     return kFALSE;
b_nlhpi->GetEntry(entry);         if (nlhpi[ipi] <= 0.1)   return kFALSE;
b_ipis->GetEntry(entry); ipis--; if (nlhpi[ipis] <= 0.1) return kFALSE;
b_njets->GetEntry(entry);         if (njets < 1)           return kFALSE;

// selection made, now analyze event
b_dm_d->GetEntry(entry);          //read branch holding dm_d
b_rpd0_t->GetEntry(entry);        //read branch holding rpd0_t
b_ptd0_d->GetEntry(entry);        //read branch holding ptd0_d

//fill some histograms
hdmd->Fill(dm_d);
h2->Fill(dm_d,rpd0_t/0.029979*1.8646/ptd0_d);
...
...
```

The Packetizer

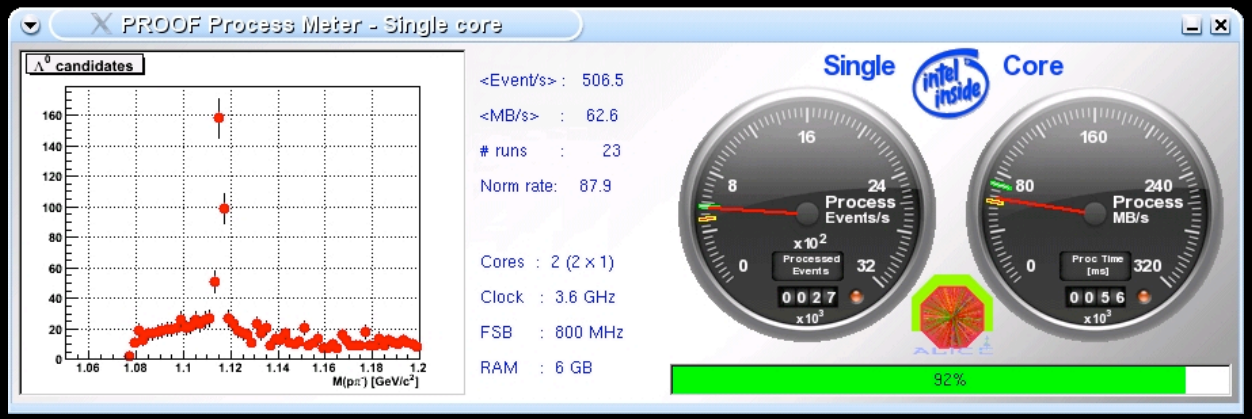
- The packetizer is the heart of the system
- It runs on the master and hands out work to the workers
- Different packetizers allow for different data access policies
  - All data on disk, allow network access
  - All data on disk, no network access
  - Data on mass storage, go file-by-file
  - Data on Grid, distribute per Storage Element
- Makes sure all workers end at the same time

Pull architecture

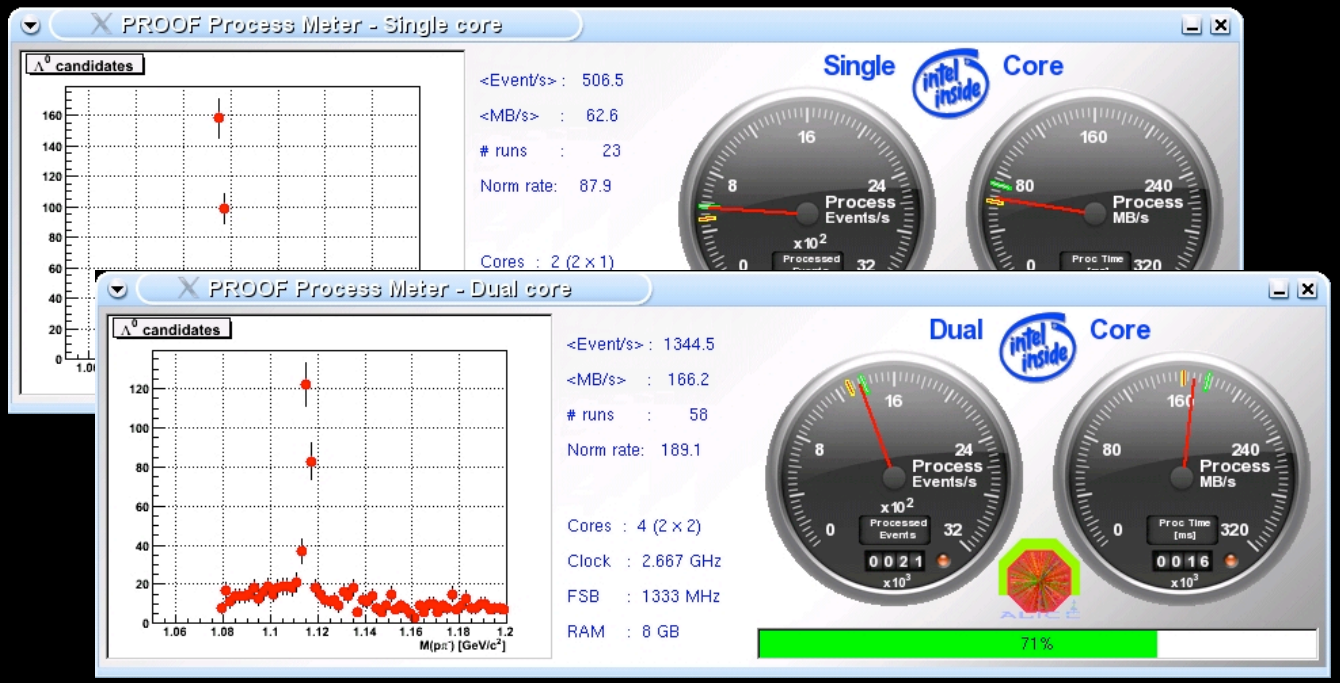
workers ask for work, no complex worker state in the master

PROOF Scalability on Multi-Core Machines

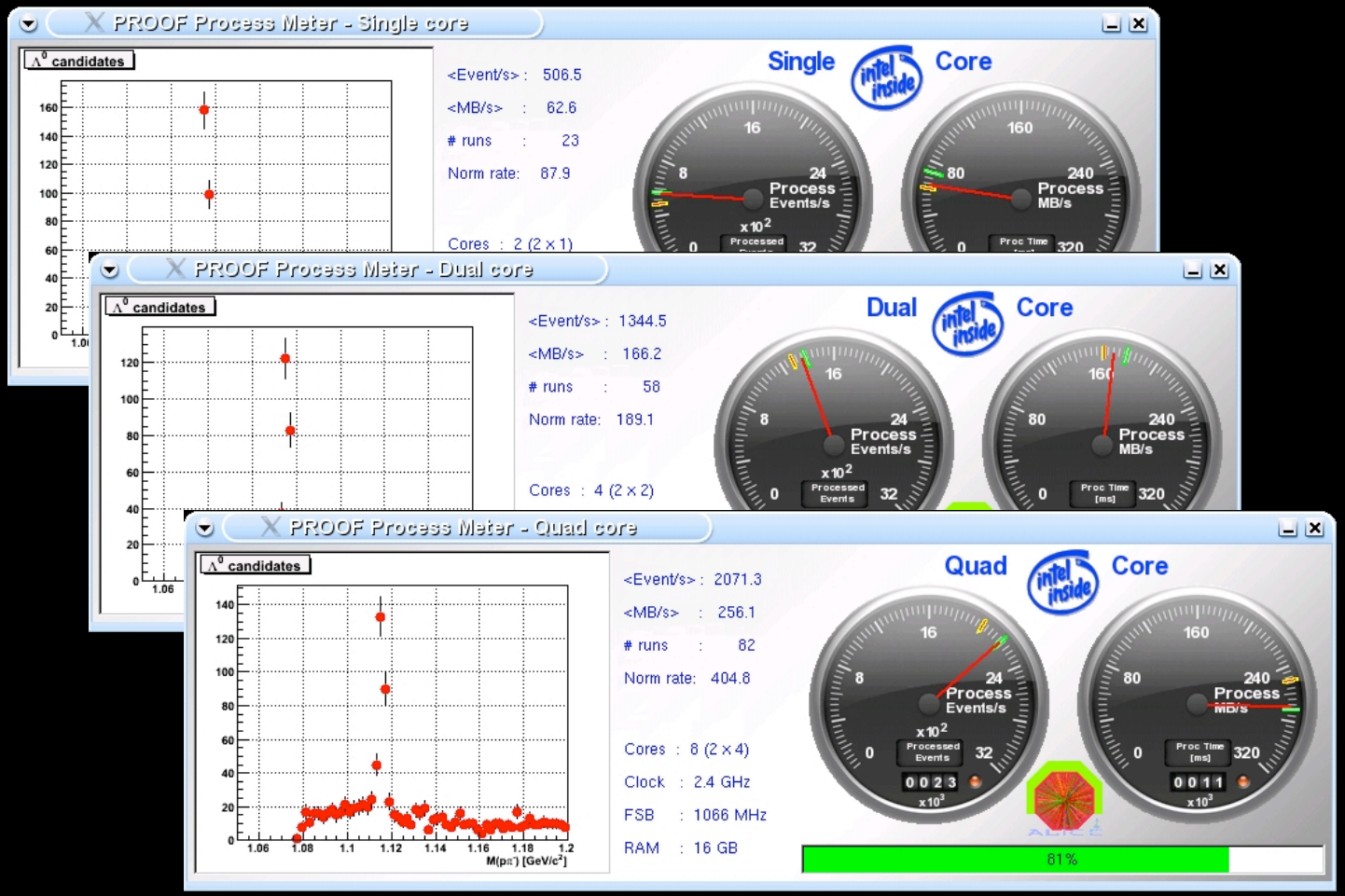
PROOF Scalability on Multi-Core Machines



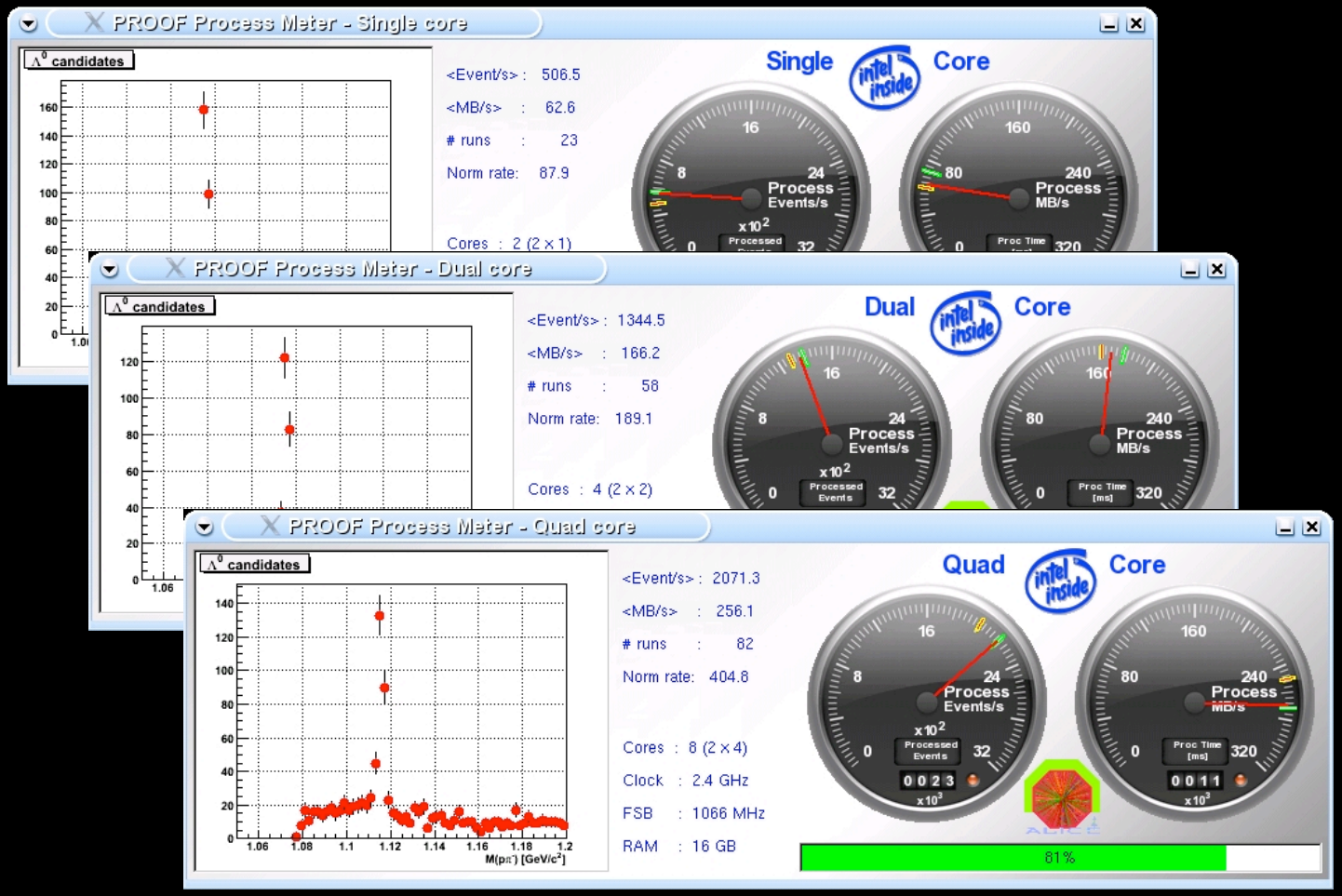
PROOF Scalability on Multi-Core Machines



PROOF Scalability on Multi-Core Machines



PROOF Scalability on Multi-Core Machines



Current version of Mac OS X fully 8 core capable. Running my MacPro since 4 months with dual Quad Core CPU's.

Interactive Batch

- Allow submission of long running queries
- Allow client/master disconnect, reconnect
- Allow interaction and feedback at any time during the processing

Analysis Scenario

AQ1: 1s query produces a local histogram
AQ2: a 10m query submitted to PROOF1
AQ3 - AQ7: short queries
AQ8: a 10h query submitted to PROOF2

Monday at 10:15
ROOT session on my laptop

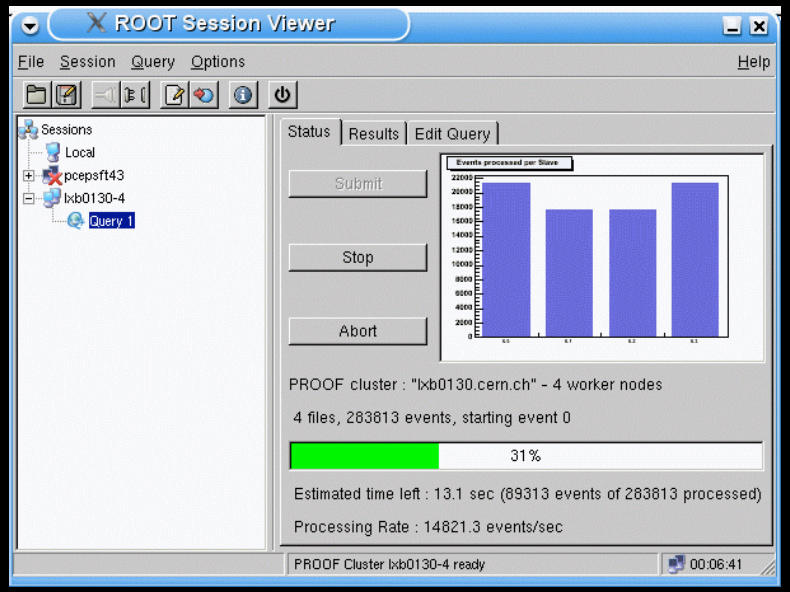
BQ1: browse results of AQ2
BQ2: browse intermediate results of AQ8
AQ3 - AQ6: submit 4 10m queries to PROOF1

Monday at 16:25
ROOT session on my desktop

CQ1: browse results of AQ8, BQ3 - BQ6

Wednesday at 8:40
Browse from any web browser

Session Viewer GUI

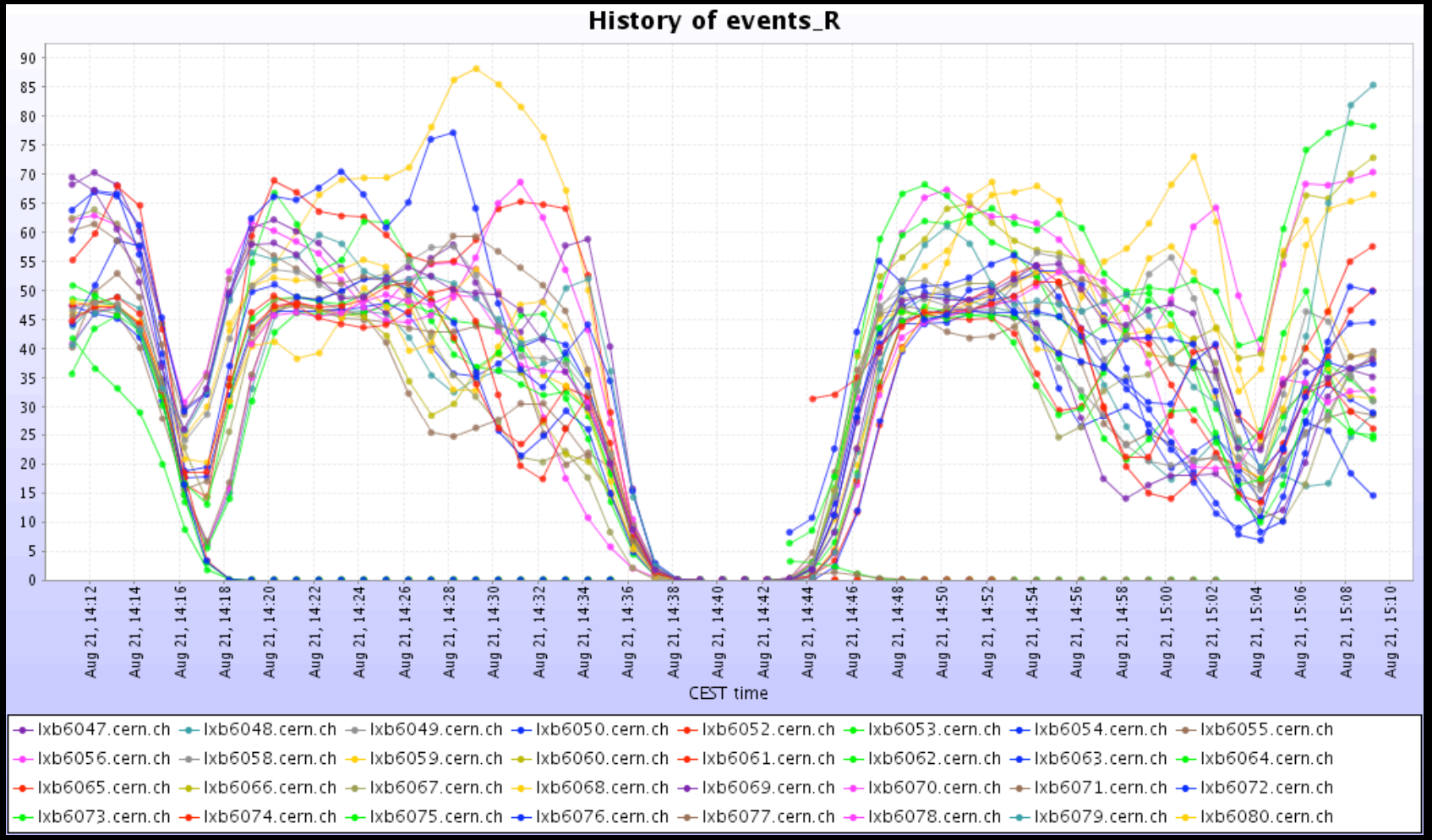


- Open/close sessions
- Define a chain
- Submit a query, execute a command
- Query editor
- Online monitoring of feedback histograms
- Browse folders with query results
- Retrieve, archive and delete query results

Monitoring

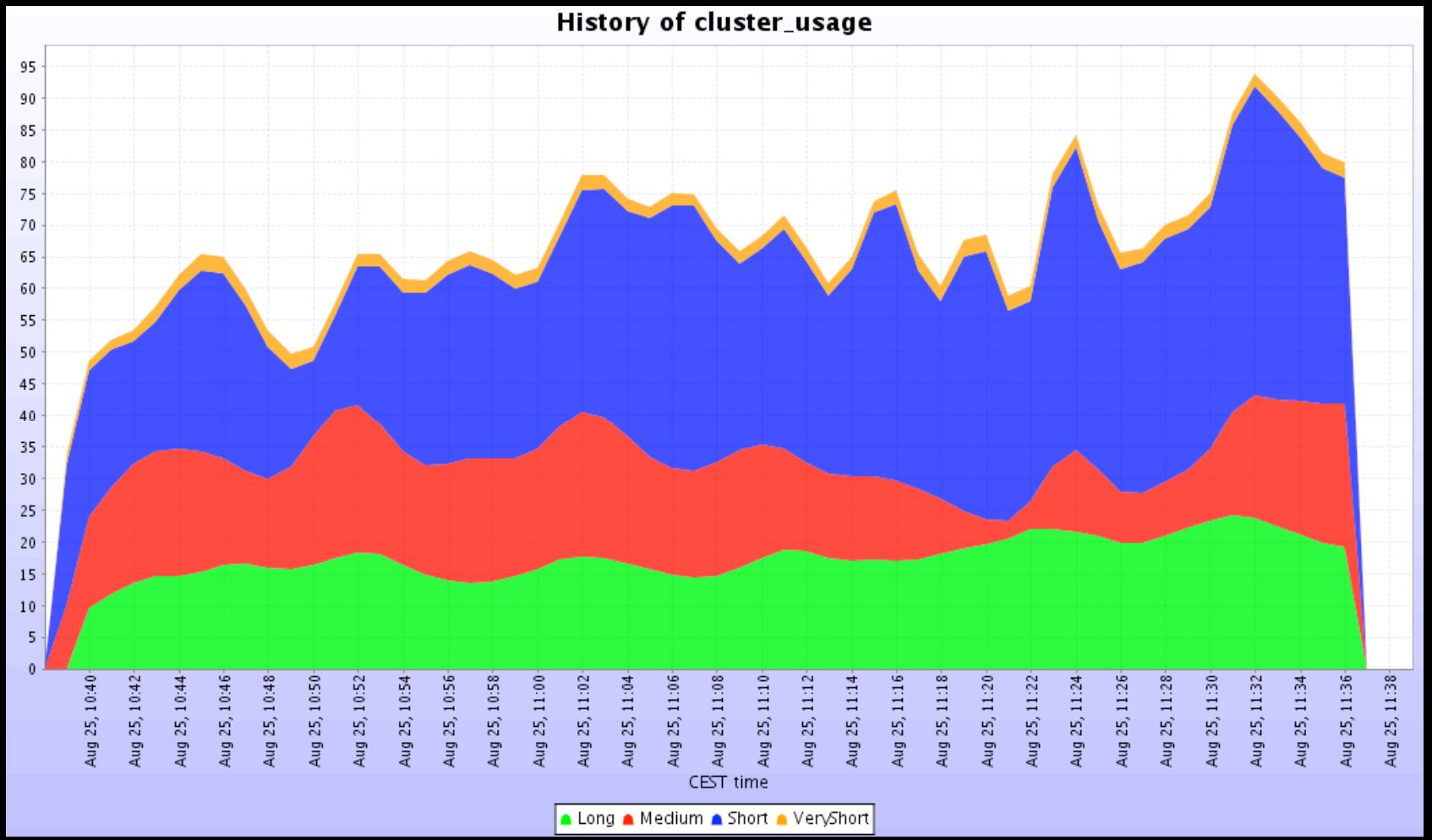
- MonALISA based monitoring
  - Each host reports to MonALISA
  - Each worker reports to MonALISA
- Internal monitoring
  - File access rate, packet generation time and latency, processing time, etc.
  - Produces a tree for further analysis

Query Monitoring



The same for: CPU usage, cluster usage, memory, event rate, local/remote MB/s and files/s

Cluster Efficiency



What's Next?

- Lot of cool developments ahead
  - Fully dynamic sessions under control of a scheduler
    - Collaboration with Condor/Wisconsin team
  - Optimized version for multi-core machines
  - Better error recovery strategy
  - Use sub-masters on large clusters to merge the output lists in parallel
  - Easy selection of feedback histograms
  - Easy installation and configuration
  - ...

The Main People Behind PROOF

- Prototype
  - Fons Rademakers (CERN)
- First generation
  - Maarten Ballintijn (MIT)
- Current version
  - Gerri Ganis (CERN)
  - Jan Iwaszkiewicz (doct student)

- And of course many thanks to all who have worked for some time on the project

Conclusions

- The LHC will generate data on a scale not seen anywhere before
- LHC experiments will critically depend on parallel solutions to analyze their enormous amounts of data
- Grids will very likely not provide the needed stability and reliability we need for repeatable high statistics end-user analysis
- A lot of interesting things still in the pipeline
- Let’s have a good workshop