BaBar Analysis Model and ROOT

Peter Elmer
Princeton University
29 September, 2005

Other presentations

- ROOT2004
  - Eric Charles – Primary data storage with ROOT
- ROOT2002
  - Peter Elmer – BaBar and ROOT data storage
- Related presentations at this workshop
  - Andy Hanushevsky – xrootd
  - Wouter Verkerke (Atlas) - RooFit

History

- This is now old news, but by way of introduction:
  - BaBar used Objectivity for event storage through 2003
  - Significant issues with scaling, data access and distribution, etc.
  - Early on a separate "micro" format (for analysis) called Kanga was created:
    - A single TTree per file containing "micro" data written by a dedicated "converter" reading from Objectivity
    - This Kanga was in a format which you could open in the ROOT framework, but wasn't structured particularly well for use for interactive analysis in ROOT directly
    - Solved most data access/distribution issues for analysis

More History

- In 2002, BaBar took the decision to abandon Objectivity altogether as the primary event storage in favor of ROOT I/O (deployed 2003-2004)
- Develop a single "new Kanga" that would be produced directly by Reconstruction/Simulation and used by analysis
- Address the issue that with both Objectivity and the original Kanga, people would run very large "ntuple productions" to transform the data to something they could use directly (TTrees for ROOT, Hbook)

“Dual use” data format

- Among other goals we decided to develop the format such that it was “dual use”:
  - Existing BaBar reco/sim/analysis Framework tools work as before (with code basically unchanged), but also enable use of new Kanga “micro” directly within ROOT Framework:
    - Write TTrees
    - Allow BaBar Framework applications to write customized “user” data (i.e. more or less the “ntuple” they would have written subsequently anyway)
    - Build a workflow as part of the data reduction which allows user customized data to be produced as part of the production processes

So what about the "dual use" model?

- Was it universally adopted? - No
- Did physics get done with it? -Yes
- Has it been universally adopted?
  - It was as fast as "bare ROOT" and was not (IMO) and more difficult to use then normal customized TTrees
  - Existing analysis group infrastructure for "ntuples"
  - Some initial difficulties with documentation, etc.
  - People love to "roll their own"
  - It wasn't the main problem for people doing analysis, at least in BaBar (see later slide)

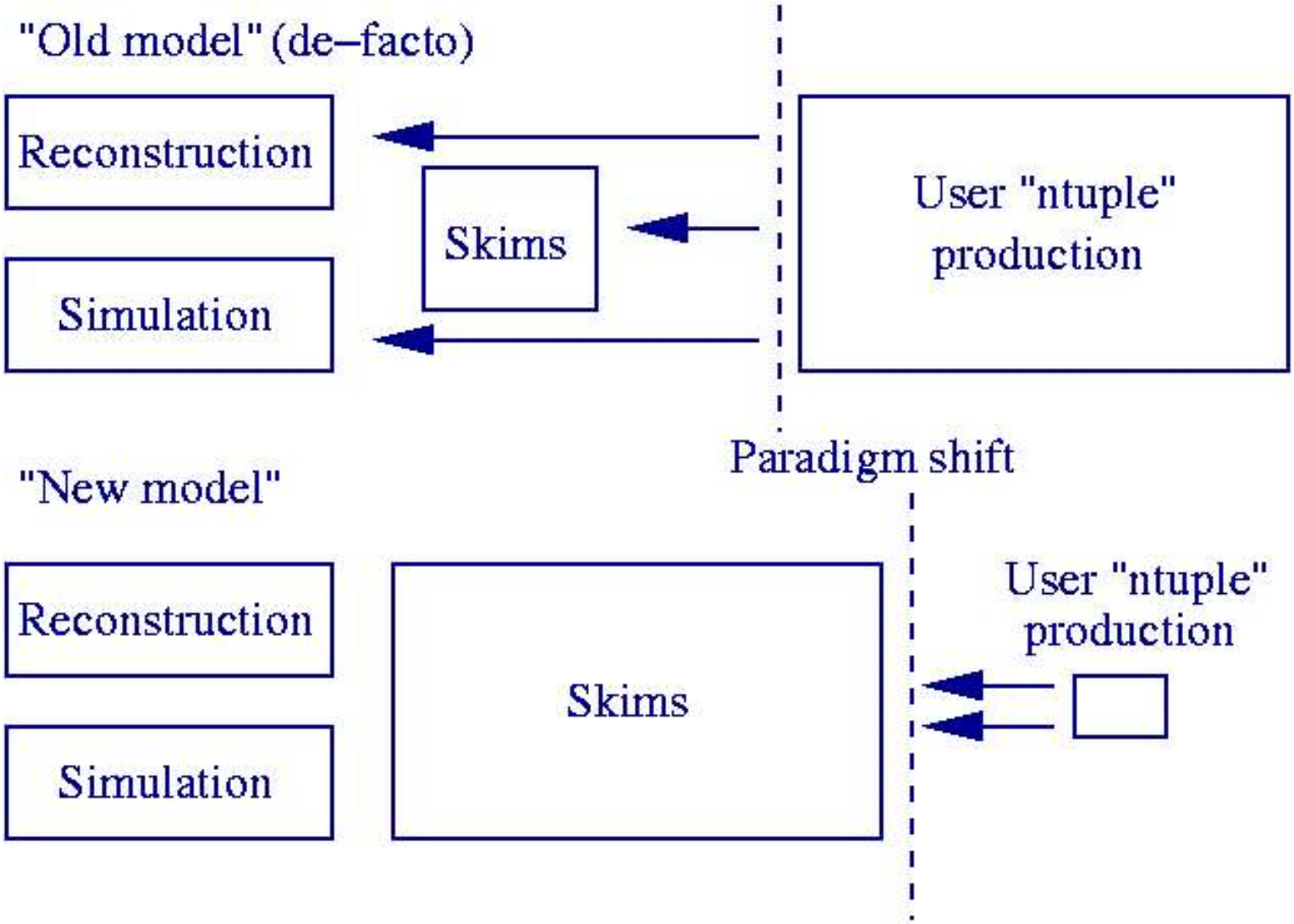
So what about the "dual use" model?

- So was it worth it?
  - Yes, unequivocally
  - Simple tools were users were easy to create
  - Debugging was simple using ROOT Framework
  - Less obscure persistency meant that it was less alien to many users
  - It gave people more options for getting their analysis work done, and some set of enthusiastic users got physics done with it

Main issues

- The main (coupled) issues impeding analysis were:
  - Data access, distribution and management
  - An effective model for data reduction

Data reduction and paradigm shift



Conclusions

- During 2003-2004 BaBar has moved forward to an event model enabling both its Framework access as well access within the ROOT Framework
- While the latter hasn't been universally adopted, it has succesfully been used for physics results and has allowed for a new, important of working for the physics user
- IMO, a “dual use” model will however never fully eliminate “user ntuple” productions and the key to managing that is data reduction.

Data Access

Modern HEP experiments are faced with two large problems:

The need to analyze very large data samples

Petabytes of data, millions of files

The need to use distributed computing resources

10-100 sites involved, lots of people

Thousands of commodity clients and servers

What is xrootd?

xrootd is a data access system

It provides performant, fault-tolerant and scalable access to data sitting on remote machines

The actual data location is transparent to the user

Functionalities for both “small” and “large” sites

Emphasis on ease of installation and operation

http://xrootd.slac.stanford.edu

Features

Fault tolerance:

Built in to the protocol

Clients wait/retry if server goes away, then look for another server

Can add or remove servers dynamically

High Performance:

Connection multiplexing

Async I/O, read ahead

Compact, efficient protocol

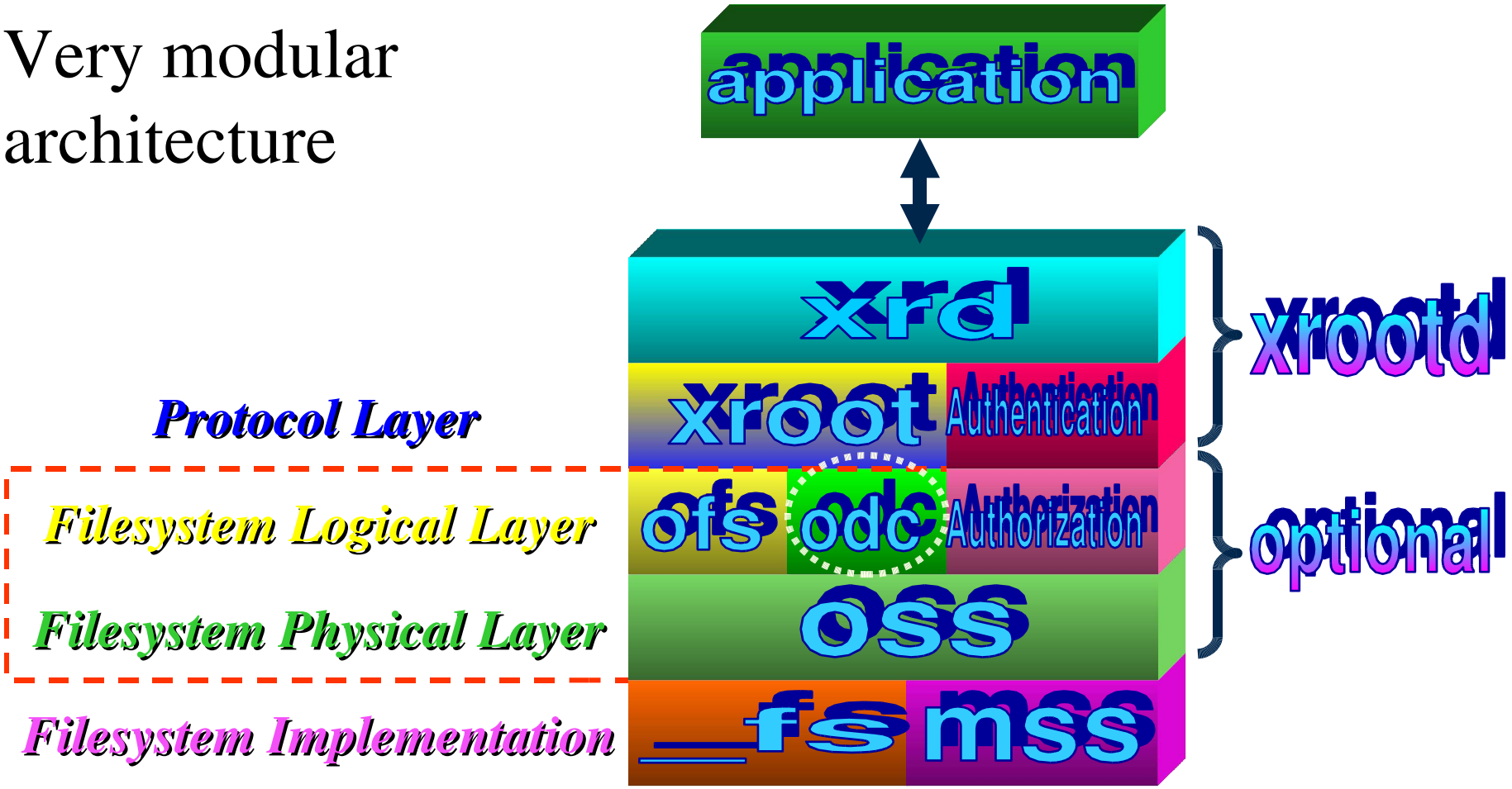
Heavily multi-threaded

Load-adaptive I/O buffer management

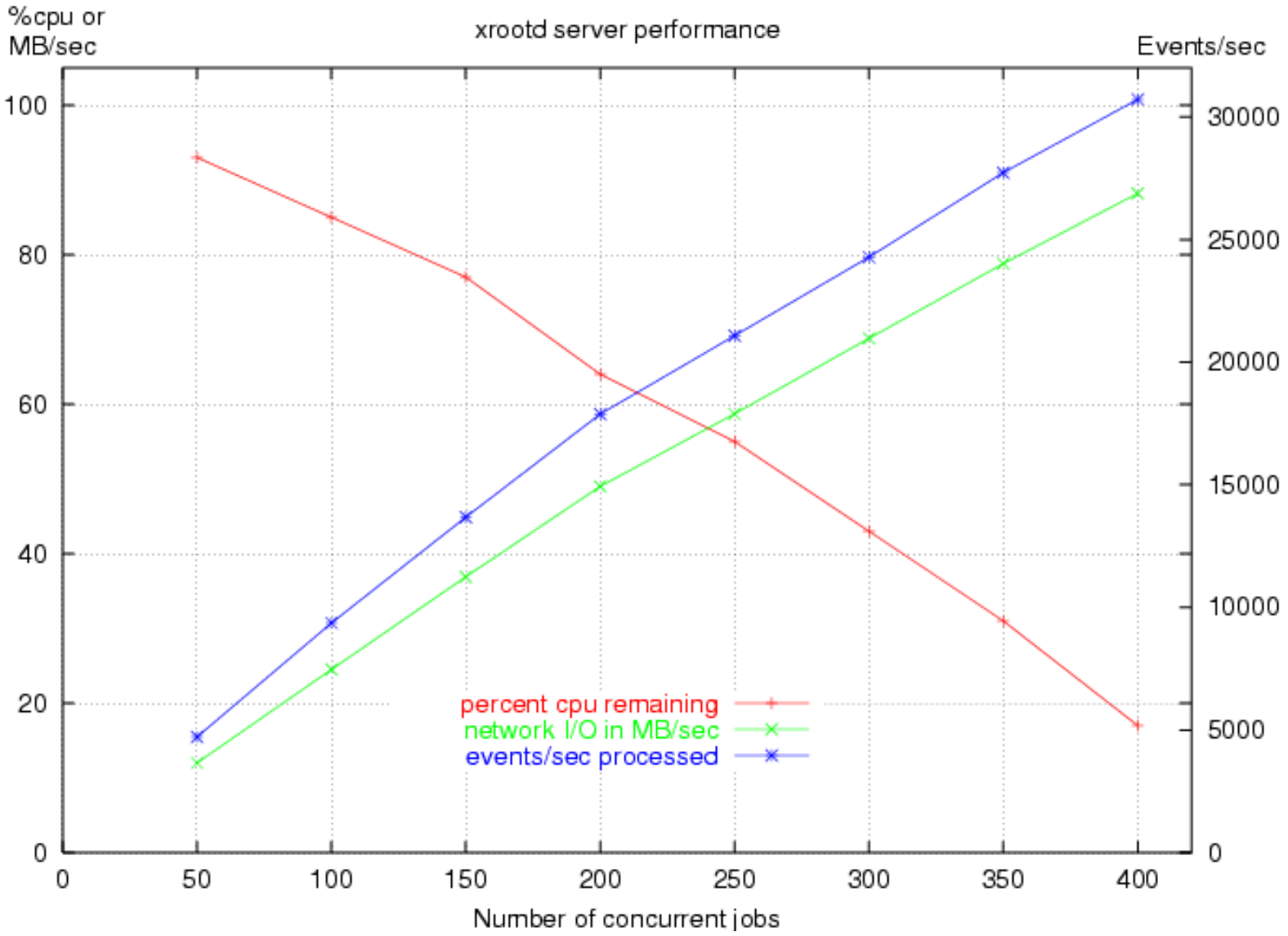
Multiple parallel requests per client

Architecture

Very modular architecture



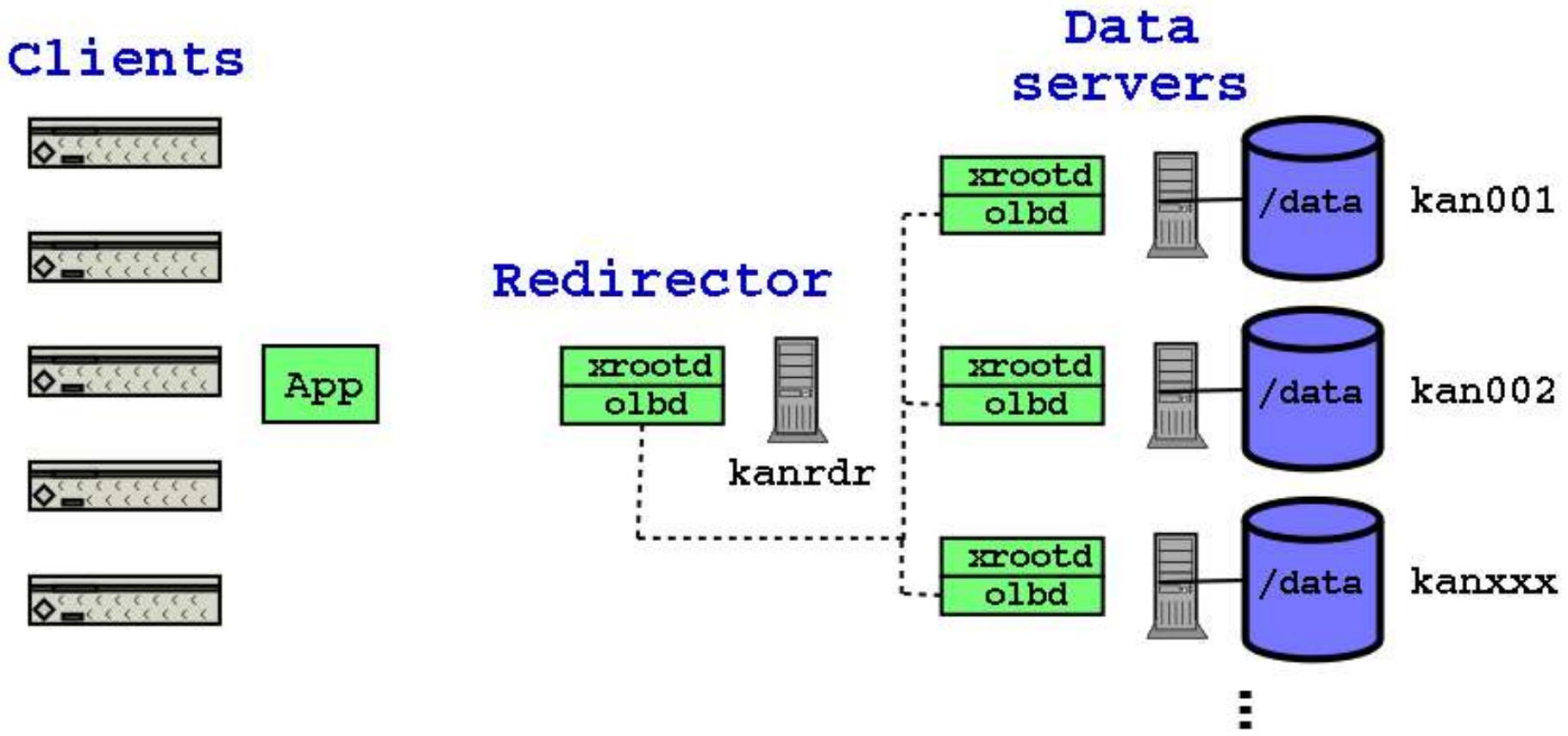
Scalability (example)



Load balanced system

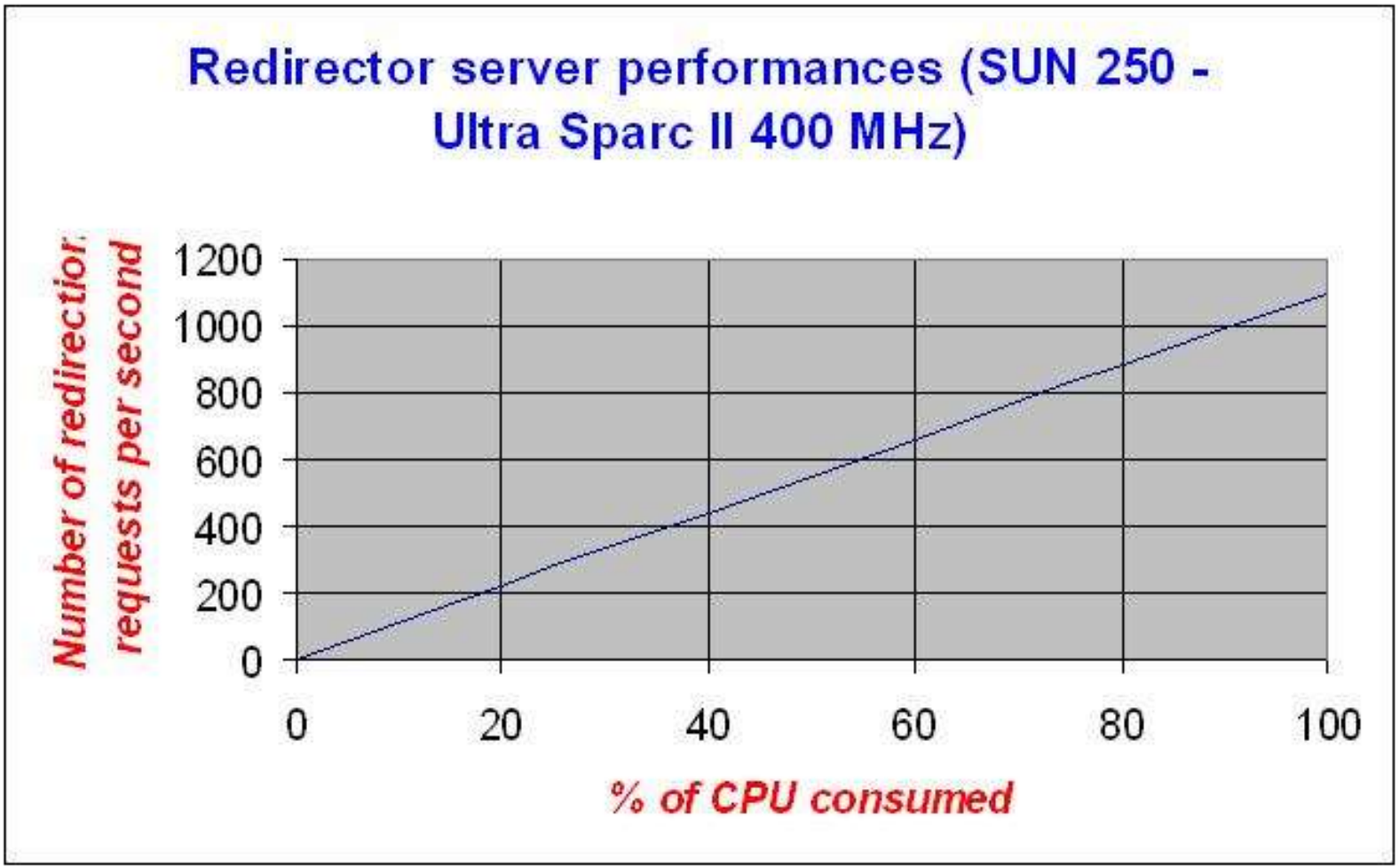
Load balancer daemon (olbd) “control” network with dynamic cache of file locations (no catalog)

Clients understands “structured peer-to-peer” network and handles redirection and failover seamlessly



Load balanced system

"Bottom heavy" system - redirect quickly to data servers



Large Example System (SLAC)

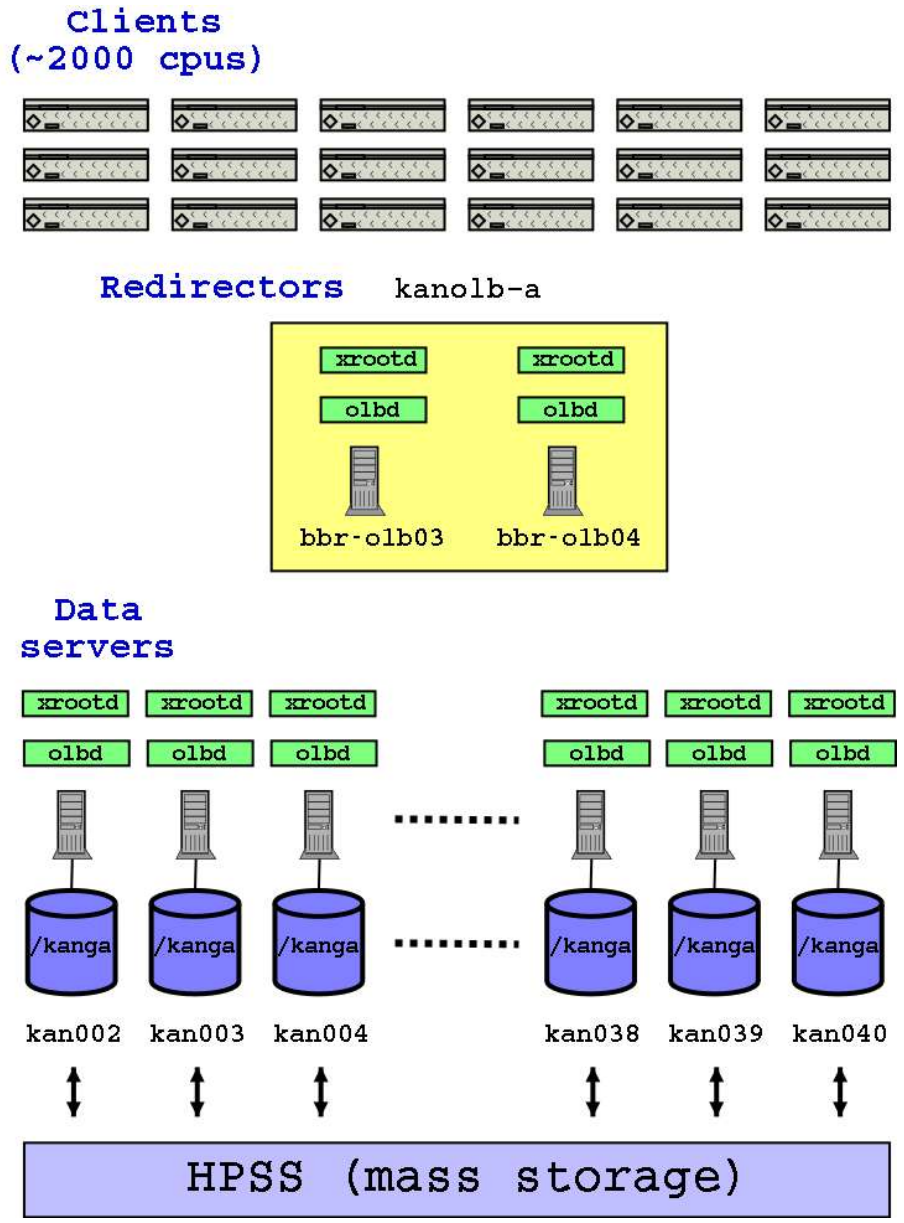
2000+ client cpus, 40 servers

200TB disk cache backed by HPSS (mass storage)

Production use since 2003

xrootd systems also deployed over the past couple of years at RAL, In2p3, FZK, INFN-Padova and CNAF plus many university sites

Test systems at BNL, Cornell and elsewhere



Clustering of many servers

Recent focus on very large server clusters

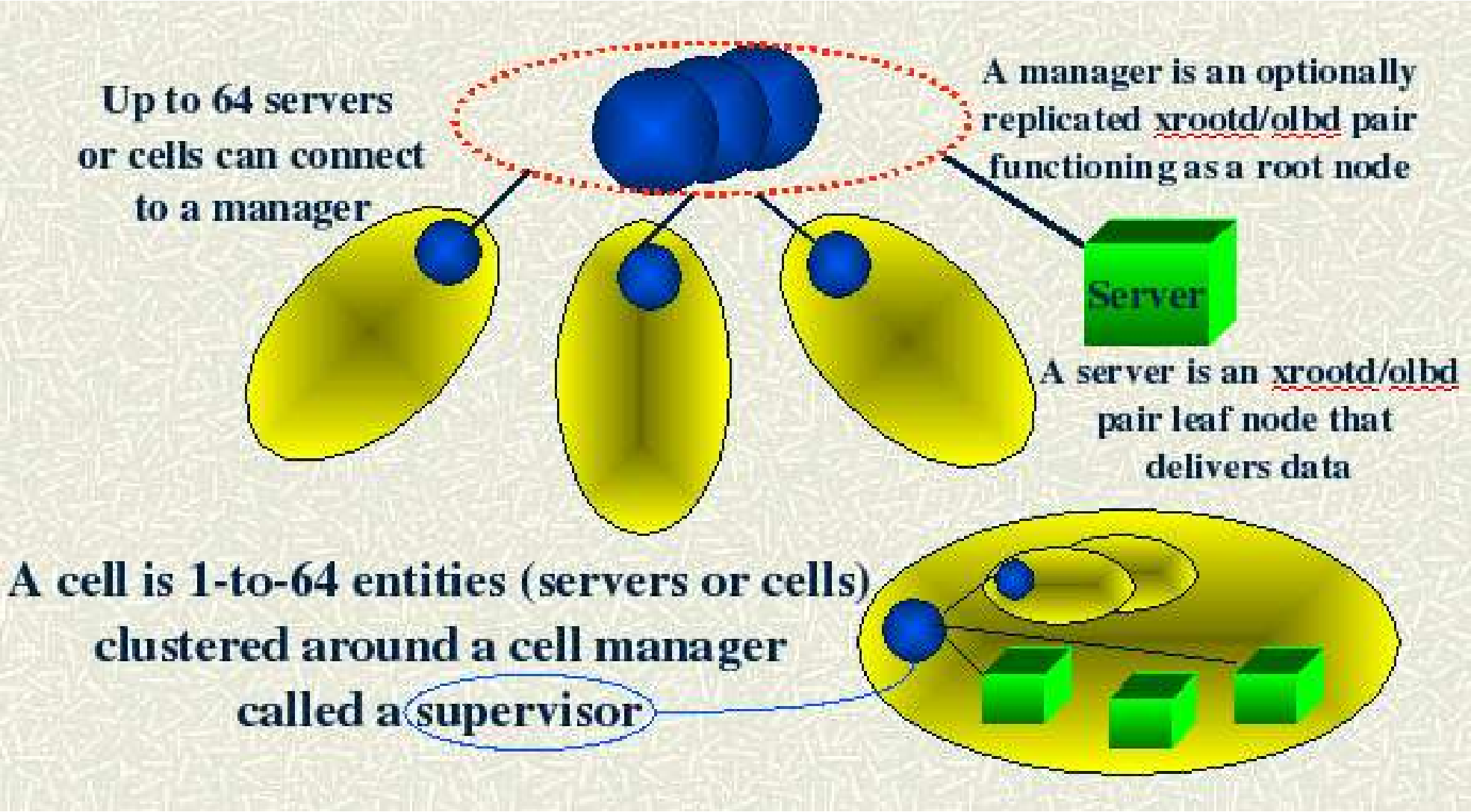
Many thousands of server machines

Hierarchical cell structure, with 64 server cells

Want scalability *without* complex configuration

Self-configuring system

Clustering of many servers



280 nodes cluster in 7 seconds, 890 nodes in 56 seconds

Monitoring

It is very interesting to monitor the data access patterns

- Want to do this in real time, across the entire system
- Must be lightweight to avoid impacting performance

Implemented a lightweight (udp based) protocol for extracting monitoring information

- Central collector daemon receives, logs and loads information into database
- Support both monitoring of file level information as well as detailed logging of each individual read

Monitoring

BaBar data access monitoring

Basic view

Top performers

List active
- users
- skims
- files
- servers
- clients
- jobs

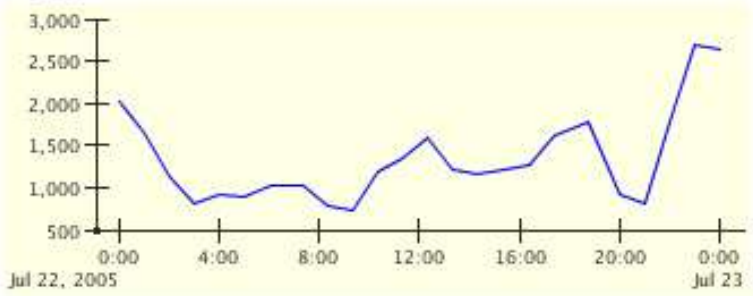
Debug
- user
- skim
- file
- server
- client
- job

Last updated: 2005-07-23 00:36:13

Time Period: Last Day | Update

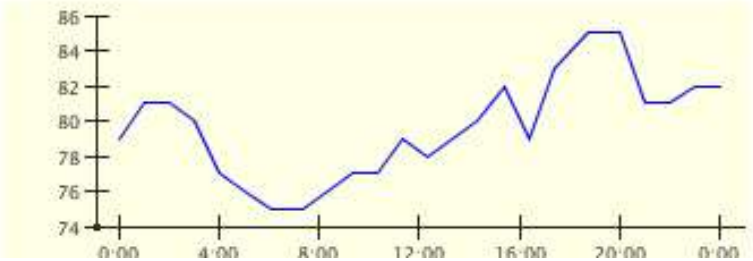
Number of running jobs

Now:	**1,552**
Change:	6 (0.39%)
Day's range:	636 - 2,900
Month's range:	718 - 3,099



Number of active users

Now:	**79**
Change:	0
Day's range:	74 - 87
Month's range:	48 - 94



Monitoring

BaBar data access monitoring

Basic view

Top performers

List active
- **users**
- skims
- files
- servers
- clients
- jobs

Debug
- user
- skim
- file
- server
- client
- job

Last updated: 2005-07-23 00:43:14

Enter a string to filter the users :

List of selected users : **276**

276 items found, displaying 1 to 15.[First/Prev] **1**, 2, 3, 4, 5, 6, 7, 8 [Next/Last]

User Name ↓
aagaard
abh
acal
acecchi
adamcun
adriand
aedwards
aeevr
agostini

Monitoring

Information for skim: **AllEvents**

Now		Last Day	
Number of current users	23	Number of past users	12
Number of jobs accessing skim	105	Number of jobs that accessed skim	1,905
Number of currently open files	1,303	Number of accessed files	210
Total size of open files [MB]	1,727,856	Total size of accessed files [MB]	6,749,043
		Volume of data read [MB]	178,472
		Volume of data written [MB]	0
		Total time files were open [hours]	16,798
Number of client hosts in use	88	Number of used client hosts	825
Number of server hosts in use	34	Number of used server hosts	37

Monitoring

Information for file: **/store/PRskims/R12/14.3.2/AllEvents/00/AllEvents_0031.01.root**
Size: **1710.55 MB**

Now		Last Day	
Number of current users	0	Number of past users	1
Number of jobs accessing file	0	Number of jobs that accessed file	8
		Total time file was open [hours]	29
		Volume of data read [MB]	403
		Volume of data written [MB]	0
Number of client hosts accessing file	0	Number of client hosts that accessed file	8
Number of hosts serving file	0	Number of hosts that served file	4

Security

XrdSec authentication framework

Originally provided kerberos authentication only

Recently ported password-based and GSI authentication from ROOT

Conclusions

xrootd is a data access system designed for Petabyte-scale data access in a distributed environment

- Provides the necessary scalability, fault tolerance and performance

Major features added over the past year:

- Self-configuring clustering of very large number of servers
- Full (and lightweight) data access monitoring
- Additional security/authentication protocols

Collaborators and Contributors

Core collaborators

- **Jacek Becla, Andy Hanushevsky** - Stanford Linear Accelerator Center, USA
- **Alvise Dorigo, Fabrizio Furano, Heinz Stockinger** - INFN-Padova, Italy
- **Peter Elmer** - Princeton University, USA
- **Derek Feichtinger, Gerri Ganis, Andreas Peters, Fons Rademakers** - CERN, Switzerland
- **Gregory Sharp** - Cornell University, USA

Contributors

- **Jean-Yves Nief**, CCIn2p3, Lyon, France
- **Chris Jones** - Cornell University, USA
- **Fulvio Galeazzi**, INFN-Padova, Italy
- **Enrica Antonioli**, INFN-Ferrara & CNAF/Bologna, Italy
- **Chris Brew, Manny Olaiya**, Rutherford Appleton Laboratory, Didcot, UK
- **Gregory Schott**, Technische Universitaet Dresden, Germany