CDF computing and event data models

F.D. Snider
CDF/Fermilab

Outline

- Introduction
- The computing model
- Grid migration
- Event data model
- Summary

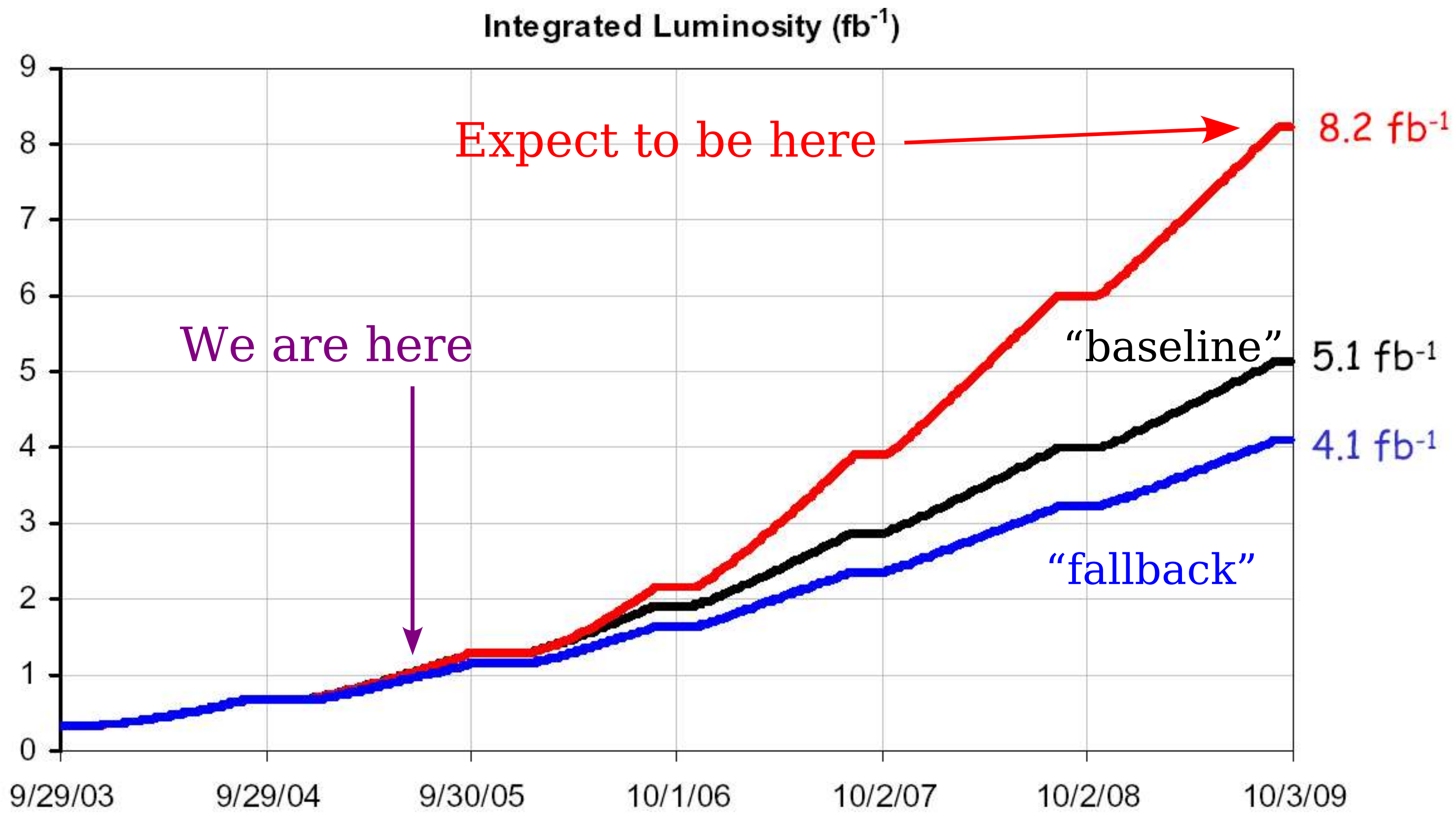
Hadron Collider Physics
July 4 – 8, 2005
Les Diablerets, Switzerland

Introduction to the computing model

- General features of the computing problem
  - Computing required to produce physics scales (approximately) linearly with:
    - Total number of events
      - CPU for analysis
    - Total data volume
      - Disk, tape, networks
    - Average event logging rate
      - CPU for reconstruction
  - For some analyses, integrated luminosity is important scaling parameter

Introduction to the computing model

- Expected delivered luminosity



Introduction to the computing model

- Data volume vs. time
  - Total = 1.2 PB

Raw + reconstructed + MC + user data

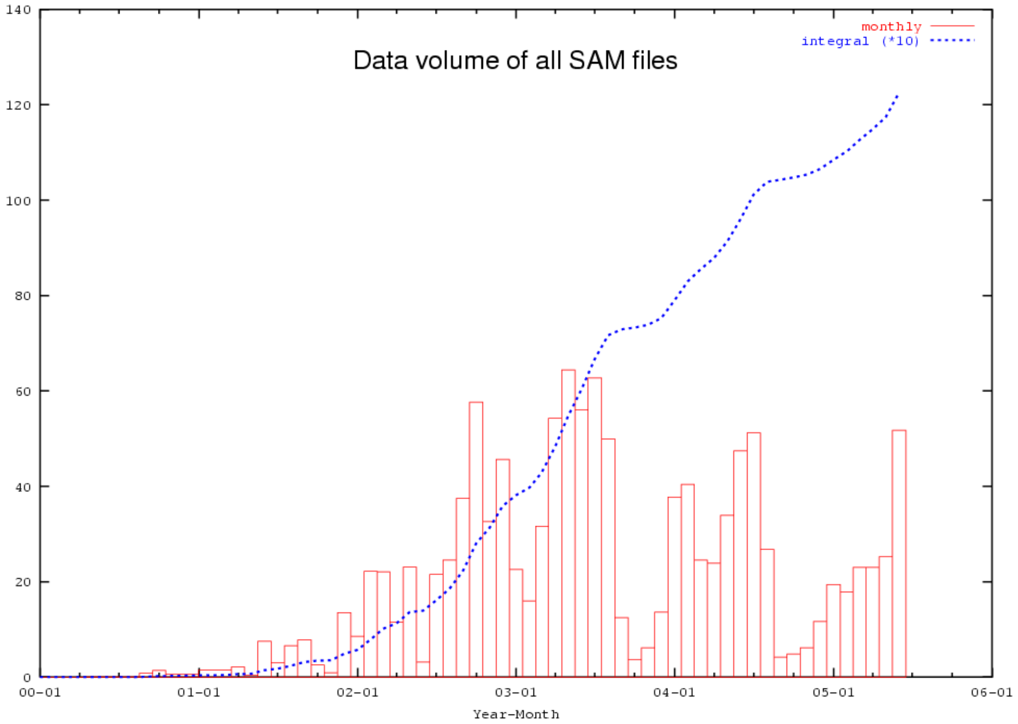
Estimated volume of
about 5 PB by 2009

Data volume (PB)

1.2

0.8

0.4



Date 1/02 1/03 1/04 1/05

Introduction to the computing model

- Specifics of the computing problem

Data logging rate triples
from 2004 to 2006

Event rate quadruples due
to increased compression

Expect ~10^{10} events by end of run

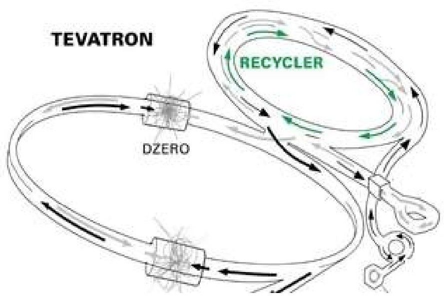
Computing problem is not static
— Becomes more difficult with time

	FY	Int L. (fb^-1)	Evts (10^9)	Peak rate (MB/s)	Peak rate (Hz)
Actual	2003	0.3	0.6	20	80
Actual	2004	0.7	1.1	20	80
Estimated	2005	1.3	2.4	40	220
Estimated	2006	2.2	4.7	60	360
Estimated	2007	3.9	7.1	60	360
Estimated	2008	6.0	9.5	60	360
Estimated	2009	8.2	12	60	360

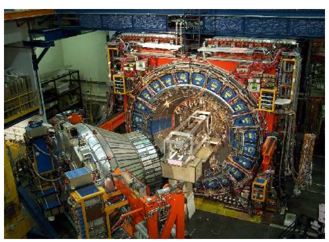
CDF computing model

- General strategy of the solution
  - Automate, centralize control of common computing tasks
    - Full event reconstruction
    - Large-scale MC production, reconstruction
    - Stripping of most physics datasets
  - Distribute computing hardware as needed
    - Platform for user analysis and MC production
  - Provide simple interfaces to allow user access to broad range of computing resources
    - Present stable, common interfaces to users
    - Automate file tracking, delivery, job parallelization
    - (Eventually) provide access to remote resources via grid tools

Major hardware systems



CDF Detector



Production Farm

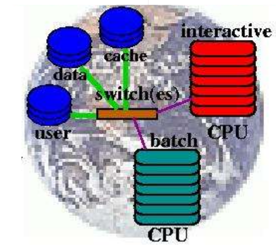
Reconstruction

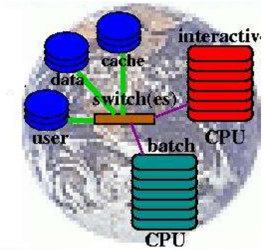
User Desktops

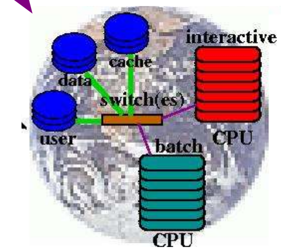
Job development,
Ntuple analysis

User job
submission

Remote CAFs







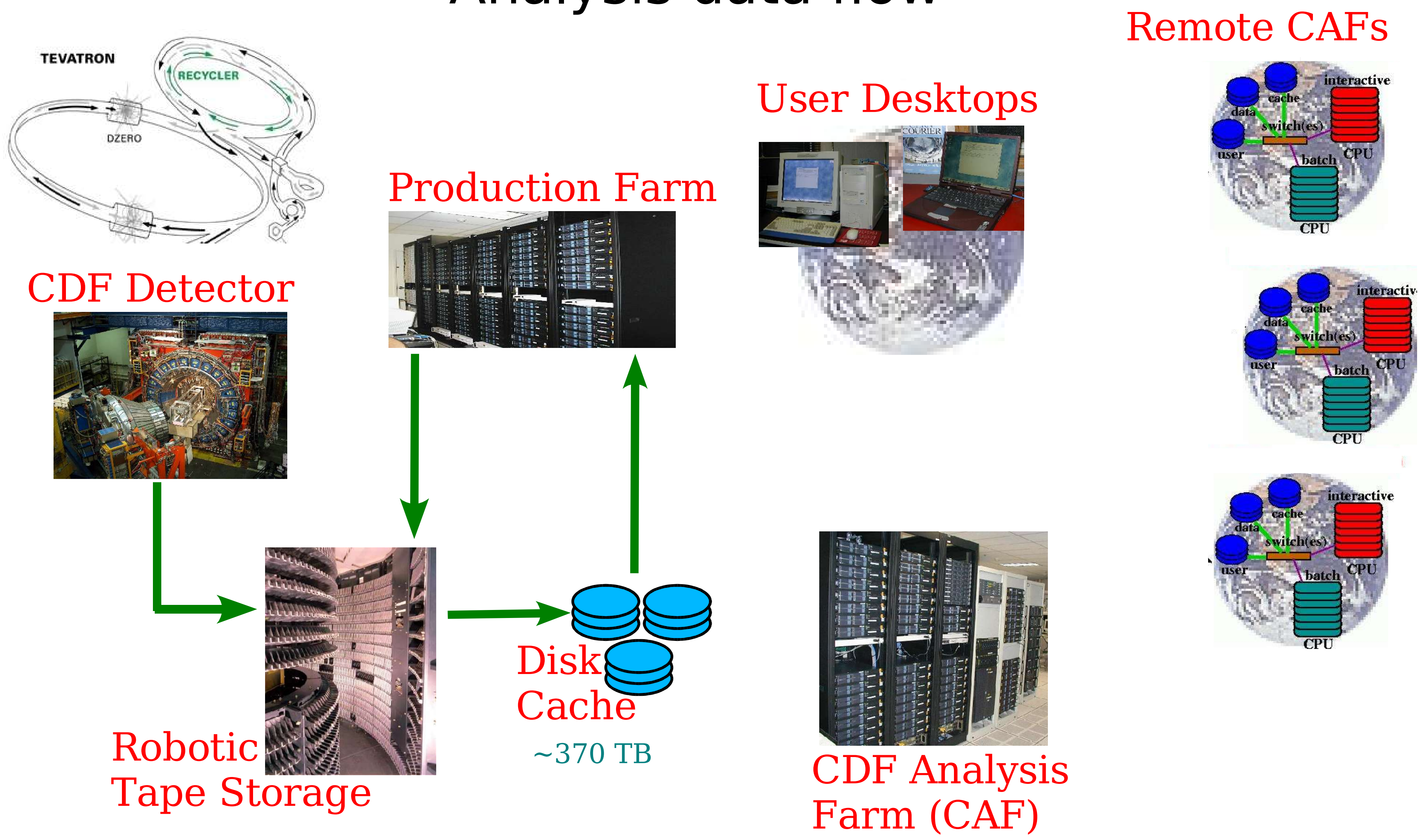
Simulation
and Analysis

Robotic
Tape Storage

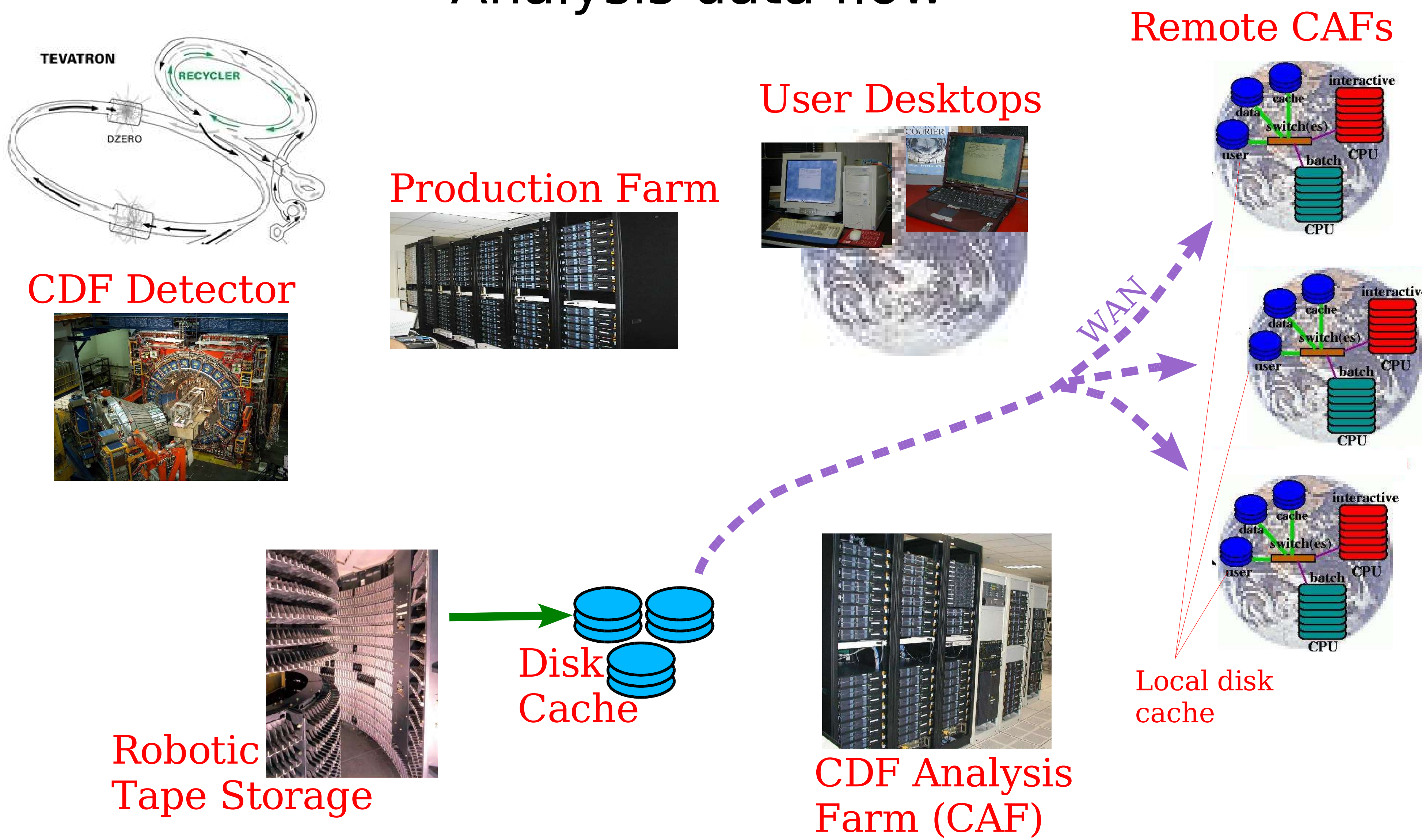
User Analysis

CDF Analysis
Farm (CAF)

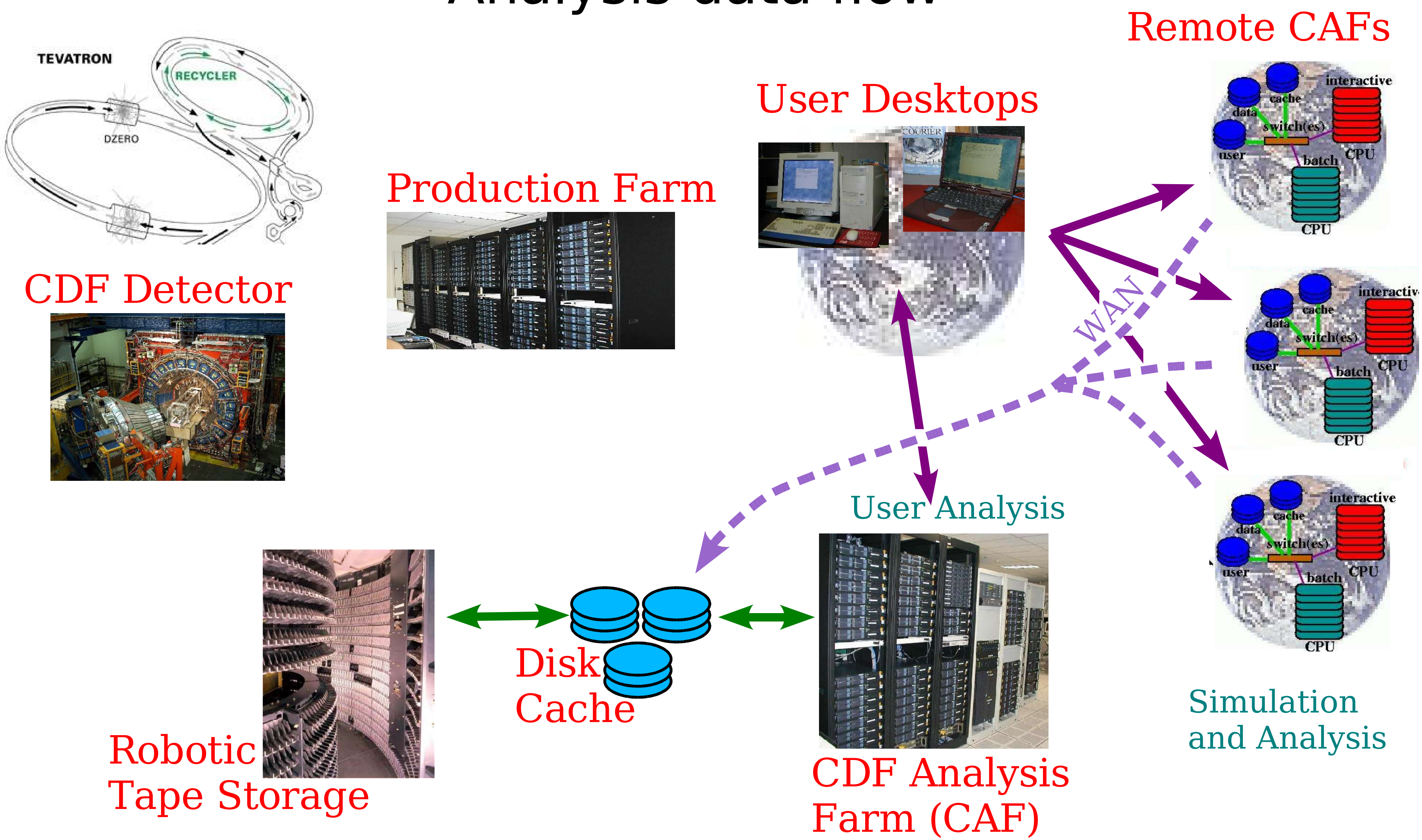
Analysis data flow



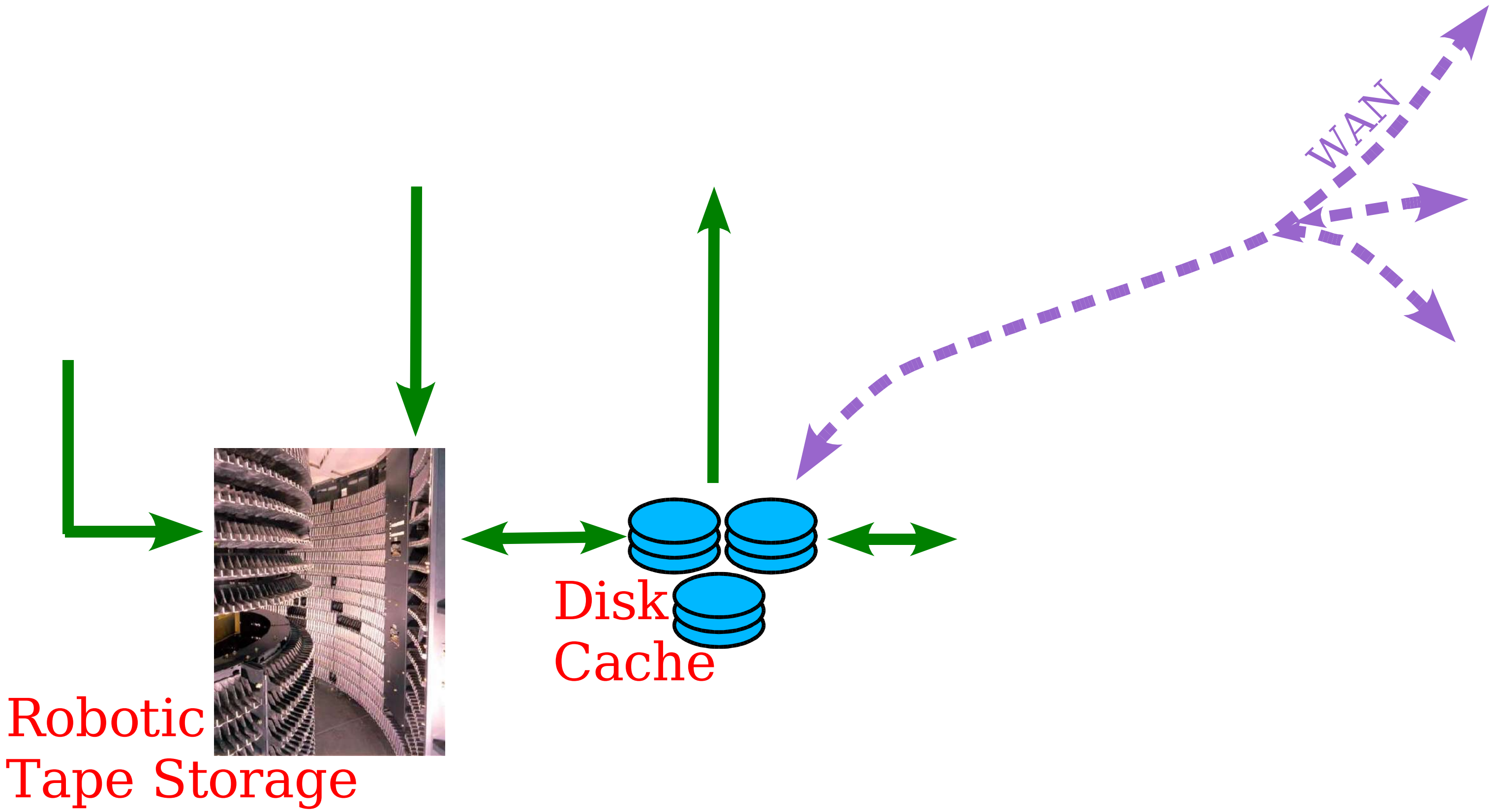
Analysis data flow



Analysis data flow



Data handling system

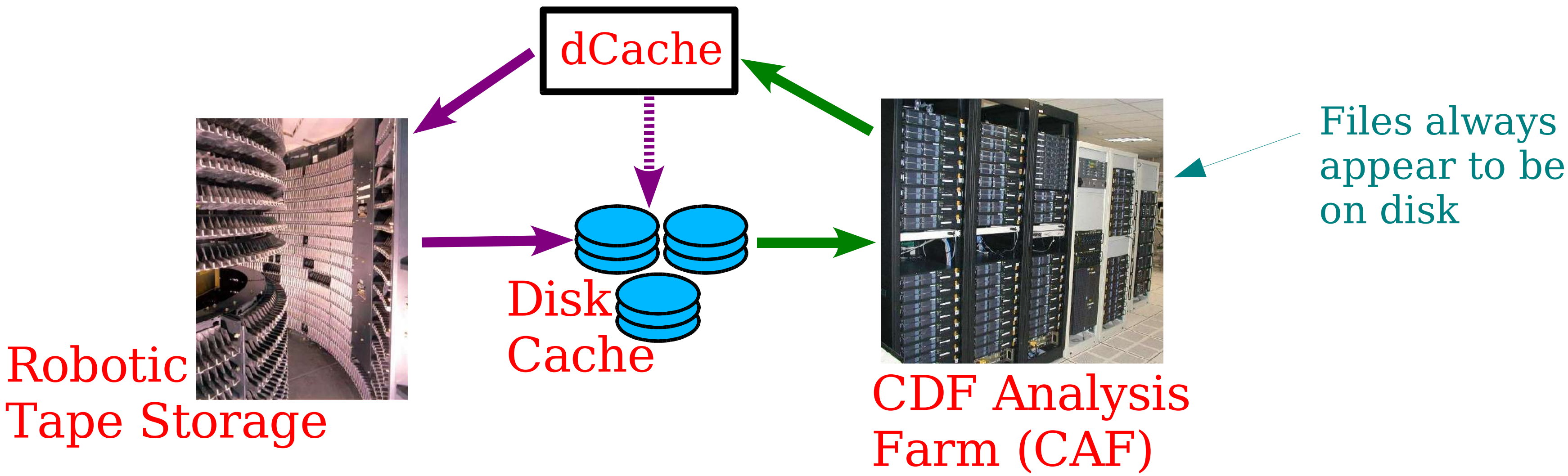


Data handling system

- Most important, technically demanding of the systems
  - Largest fraction of development effort
  - Performance and fault tolerance are paramount
- Role of data handling system
  - Data cataloging and archiving
  - Provide data access: locate and “deliver” files upon request
    - Handles details of copying from tape or another disk, checking file integrity, opening high BW channel to file, latencies, etc.
    - Underlying transactions are transparent to user
      - Typically does not need to know details such as file names
- Two major components: “SAM” and “dCache”

Data handling system

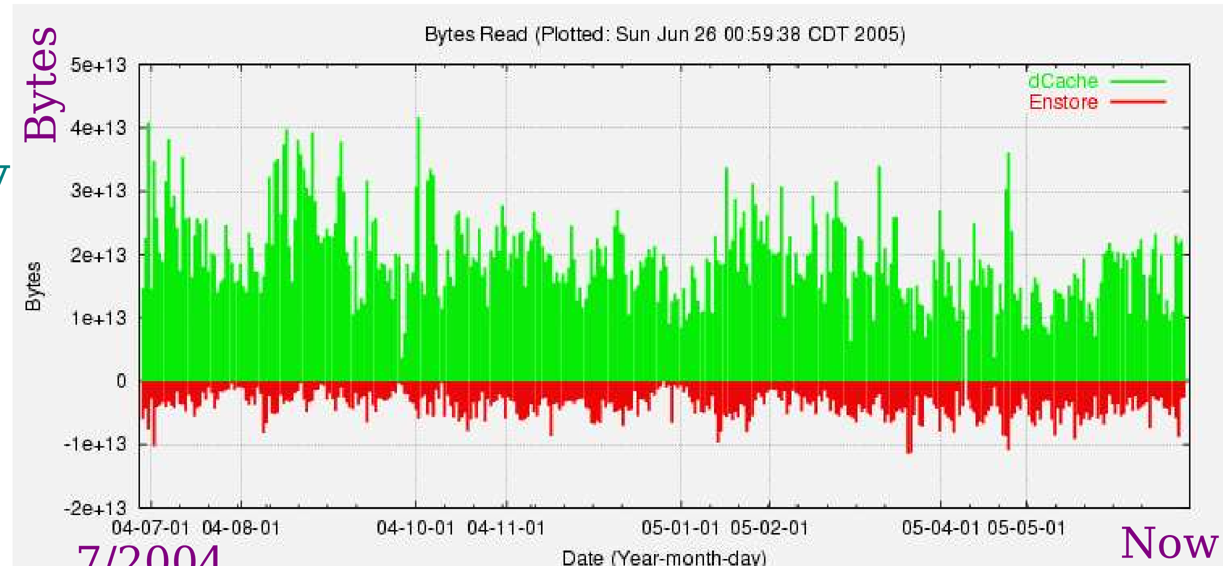
- dCache
  (Joint project of DESY, FNAL)
  - "Virtualizes" disk used for local cache
    - Data on tape or distributed across many local servers
    - Exact location hidden from user

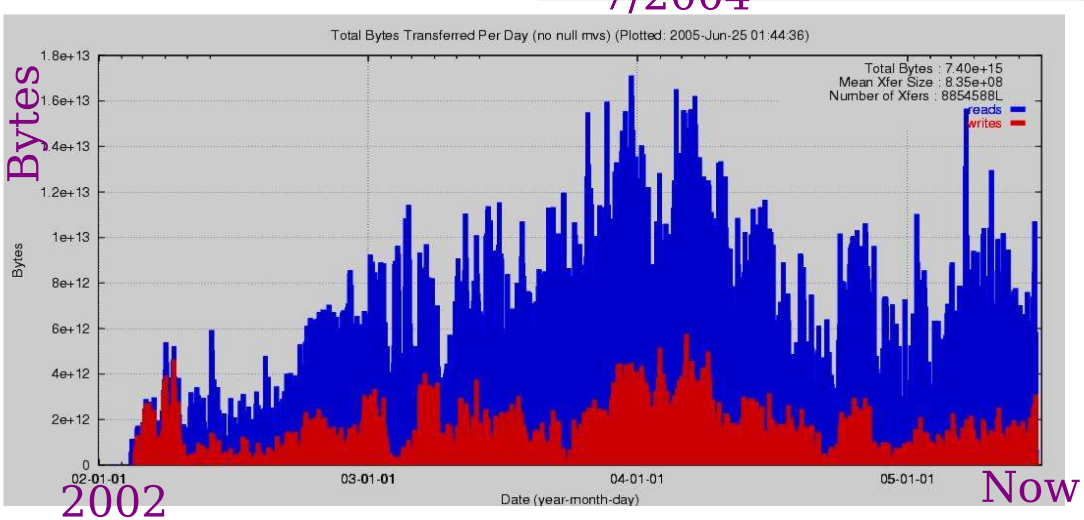


  - Used only this component and data catalog for > 2 years

Data handling system

Data from dCache
Typ. 10–25 TB/day

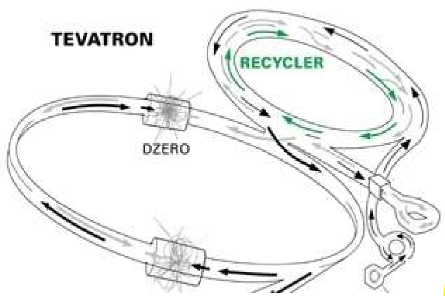




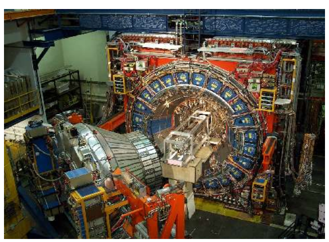
Data to/from archive
Typ. 5-10 TB/day

Data handling system

- SAM: Sequential Access via Metadata
  - New to central systems at CDF. Used at D0 for several yrs.
- Why?
  - Designed for highly distributed data
    - Better suited to increasing use of remote computing
  - A better tool to handle large datasets (needed this long ago)
    - Simple tools to define datasets based upon metadata
    - File tracking information
      - Location, delivery and “consumption” status
    - Allows process automation
      - Already used to run production farm
      - Will become central tool in user processing



CDF Detector



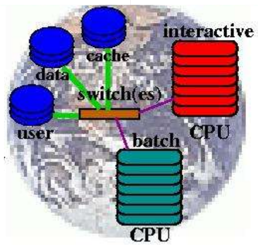
Production Farm

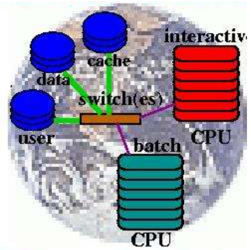
Reconstruction

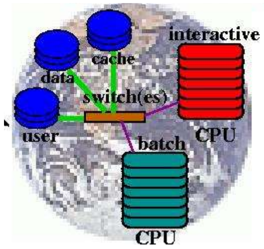
User Desktops

Job development,
Ntuple analysis

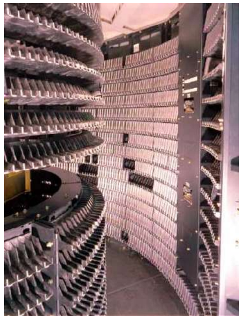
Remote CAFs







Simulation
and Analysis



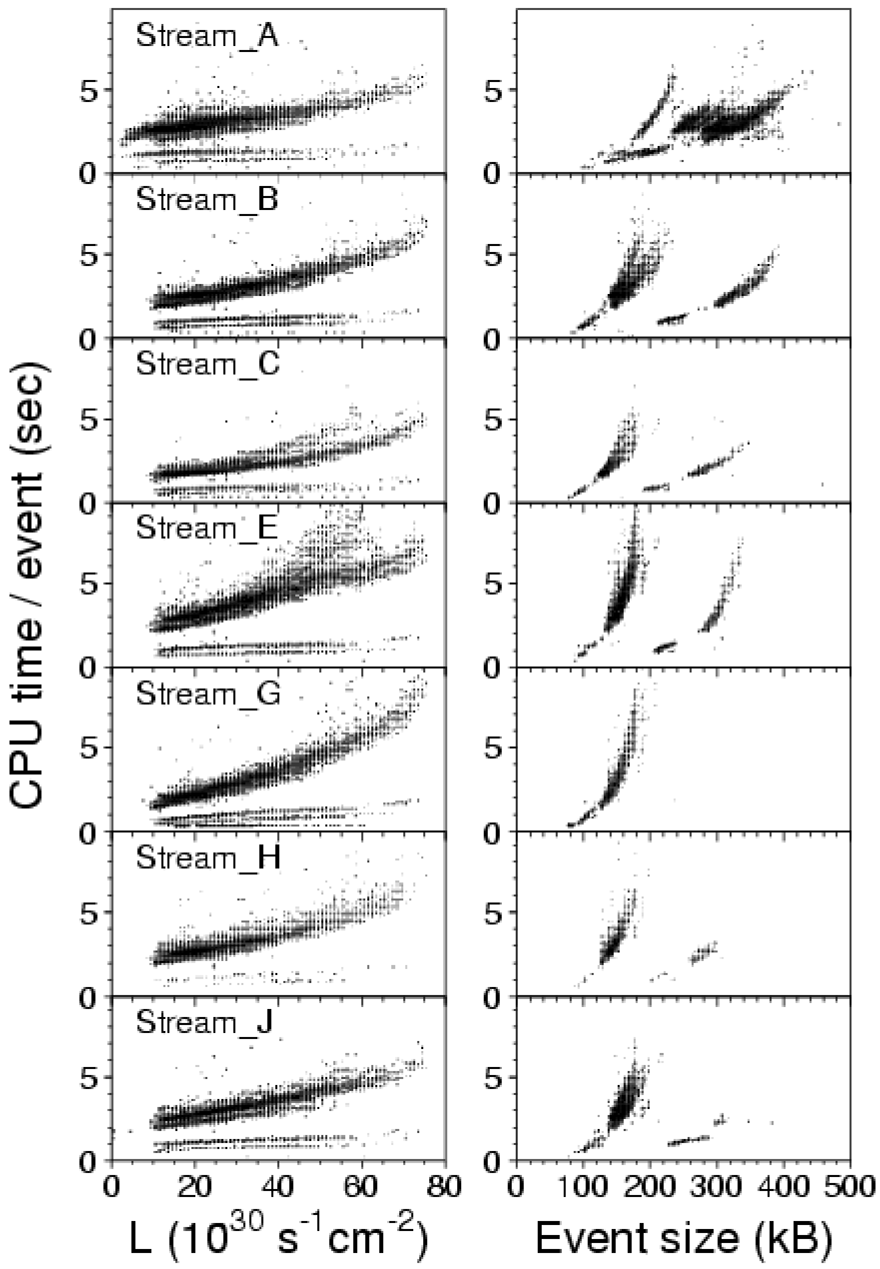
Robotic
Tape Storage

User Analysis

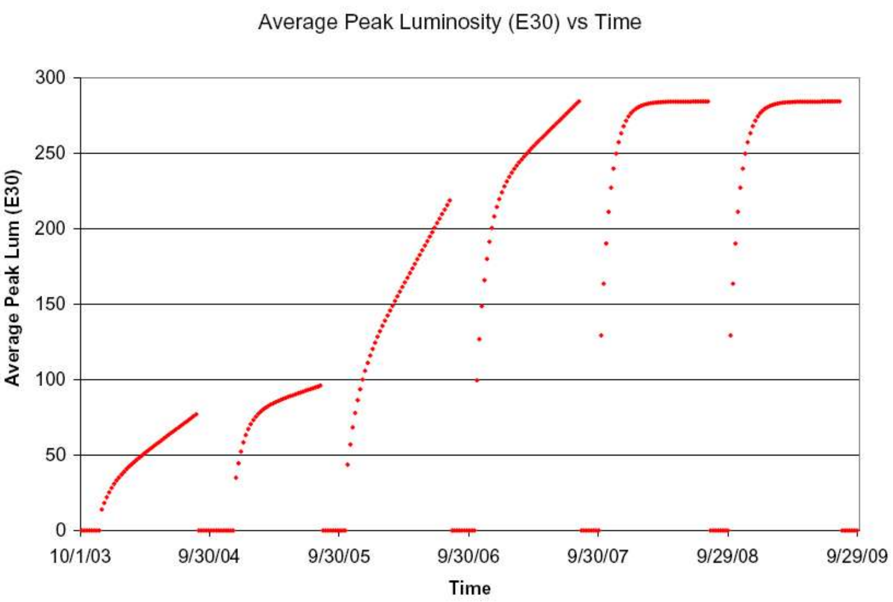
CDF Analysis
Farm (CAF)

Production farm

- Objectives
  - Perform full reconstruction of all data
    - First step in all analyses
  - Deliver results as soon as possible after data taking
- The most predictable of the computing problems
  - Can be completely automated
  - Required computing is easily calculated



Event processing time and input event size depend upon type of trigger and instantaneous luminosity



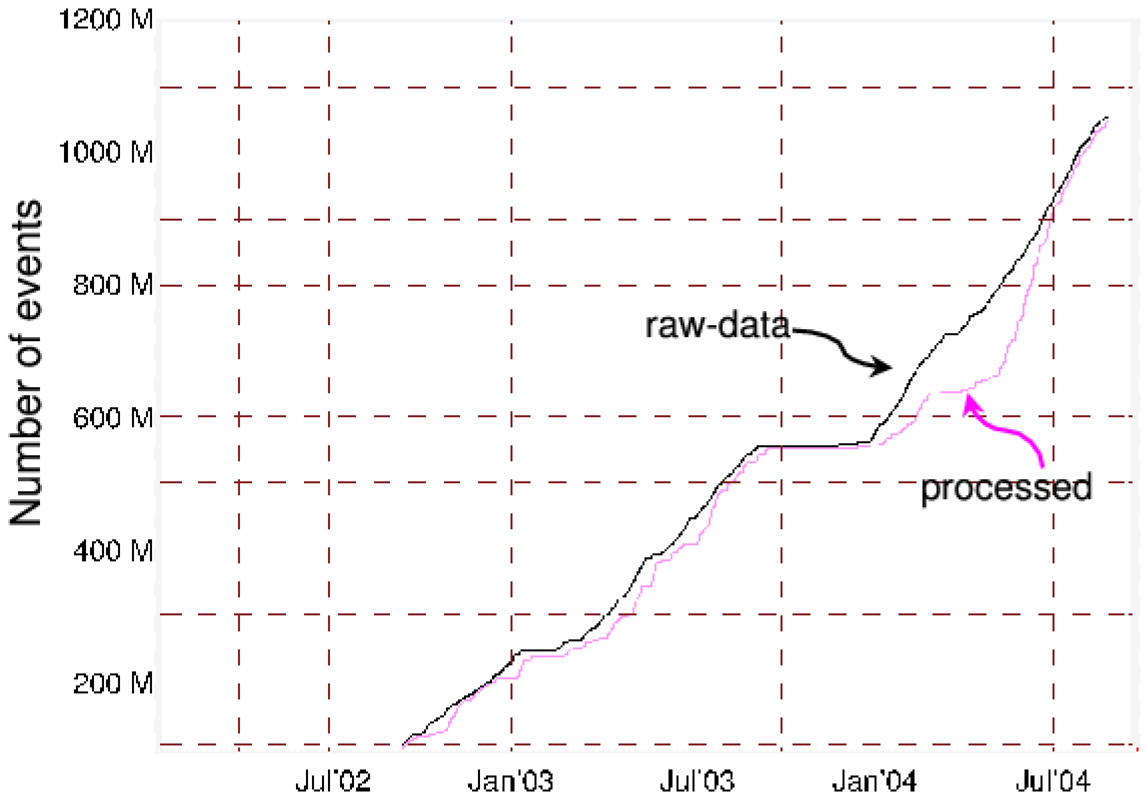
Production farm

- Processing strategy
  - Provide monitoring data within 3 days of data taking
  - Full reconstruction of all events with final calibrations
    - Deliver within 1 – 2 months of data taking (new this year)
    - Requires processing all data 1.3 times in that time
- Average event properties
  - Reconstruction time: 2.7 sec/event (1 GHz PIII)
  - Event rate: 130 Hz (FY05) to 220 Hz (FY06+)
  - Event size: 150 kB (input), 120 kB (output)

    Includes raw data
- Conclusion: Need about 150 duals
  - Catching up now using about 100

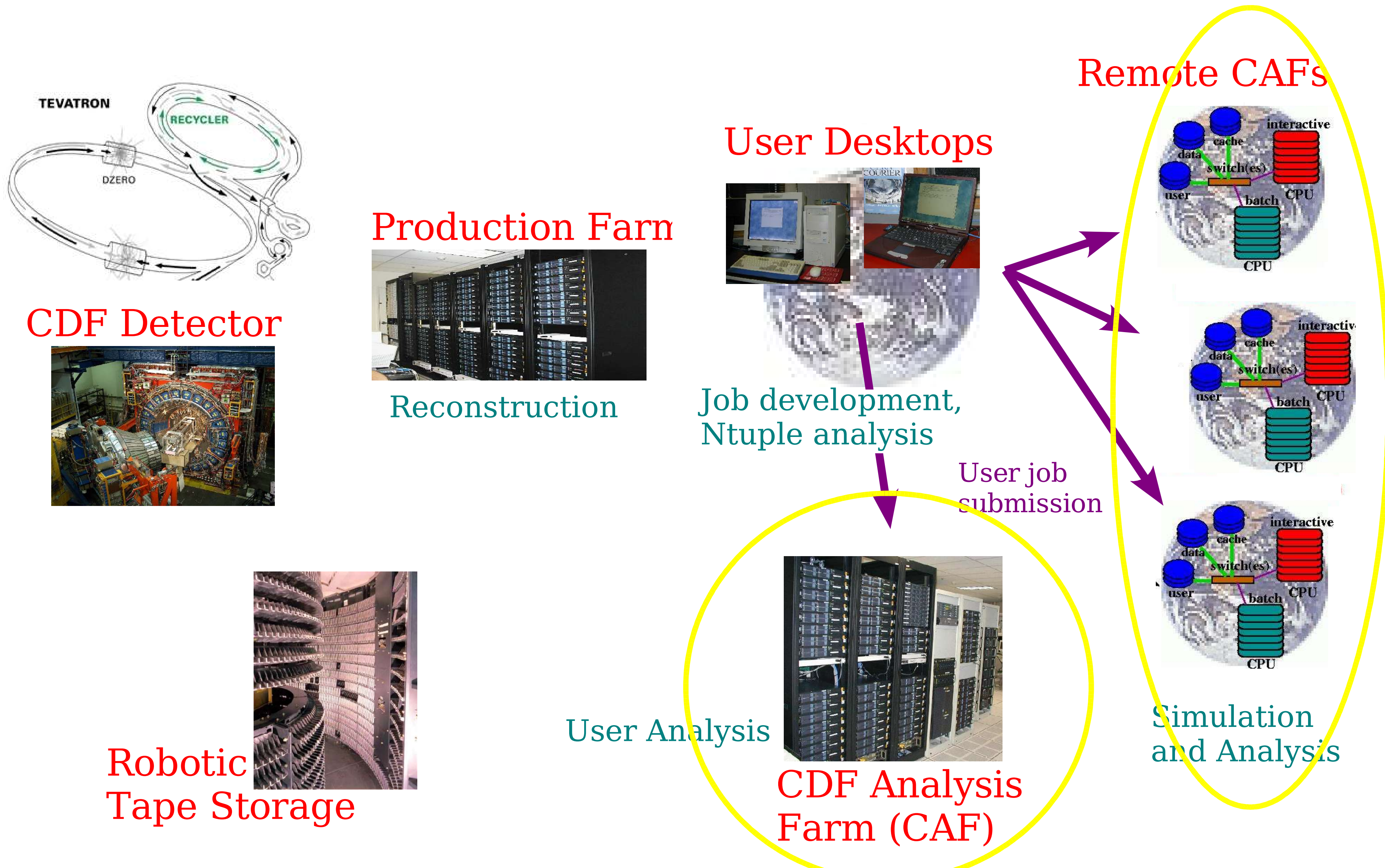
Production farm

- Processing history through 2004



Production farm

- Other features of production farm
  - Currently a total of 1.2 THz PIII equivalent (480k SpecInt2k)
  - Farm processing automated using SAM
  - Job management based upon analysis farm infrastructure
    - Dynamically expand into analysis farm resources as needed
  - System can, in principle, be distributed to remote sites


TEVATRON
RECYCLER
DZERO
CDF Detector
Production Farm
Reconstruction
User Desktops
Job development,
Ntuple analysis
User job
submission
Remote CAFs
interactive
cache
data
switch(es)
user
batch
CPU
Simulation
and Analysis
Robotic
Tape Storage
User Analysis
CDF Analysis
Farm (CAF)

CDF Analysis Farms (CAF)

- Primary analysis platform for the experiment
  - User analysis (the least predictable computing problem)
    - Ntuple creation
    - Ntuple analysis
    - Many other CPU intensive calculations
  - Semi-coordinated activities
    - Secondary, tertiary dataset production
    - MC event generation, detector simulation and reconstruction
- CAF contains the bulk of available computing capacity
- Computing in clusters located around the world

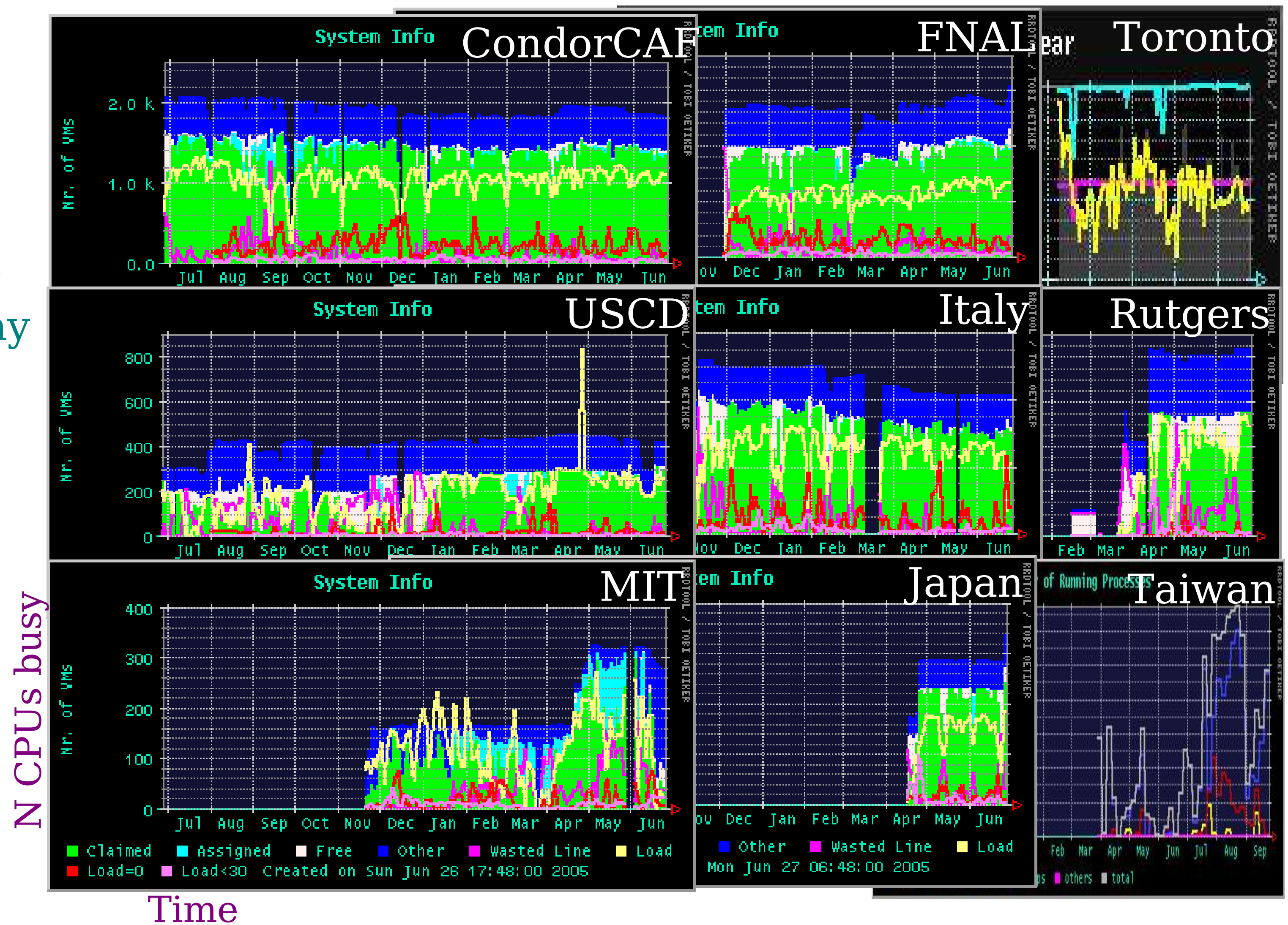
Current CPU and disk resources in CAFs

Current Resources [*]			
Cluster Name and Home Page	Monitoring and Direct Information Links	CPU (GHz)	Disk space (TBytes)
Original FNAL CAF	queues, user history, analyze, ganglia, sam station, consumption	1000	370
FNAL CondorCAF (Fermilab)	queues, user history, analyze, ganglia, sam station, consumption	2200	(shared w/CAF)
CNAFCAF (Bologna, Italy)	queues, user history, analyze, resources, network, sam station, datasets, consumption	480	32
KORCAF (KNU, Korea)	queues, user history, ganglia, sam station, datasets, consumption	178	5.1
ASCAF (Academia Sinica, Taiwan)	queues, user history, ganglia, sam station, datasets, consumption	134	3.0
SDSC CondorCAF (San Diego)	queues, user history, analyze, ganglia, sam station, datasets, consumption	380	4.0
HEXCAF (Rutgers)	queues, cpu, sam station, datasets, consumption	100	4.0
TORCAF (Toronto CDF)	queues, user history, analyze, ganglia, disk status, sam station, datasets, consumption	576	10
JPCAF (Tsukuba, Japan)	queues, user history, ganglia, sam station, datasets, consumption	152	10
CANCAF (Cantabria, Spain)	queues, user history, ganglia, sam station	50	1.5
MIT (Boston, USA) (MC only)	queues, user history, analyze	322	3.2
	Current Totals [*]:	5572	448

Utilization is high as soon as a site becomes available.

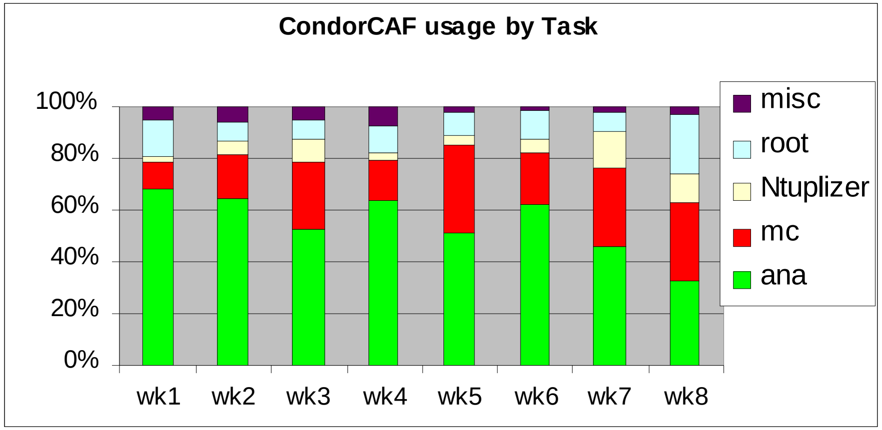
400 active users

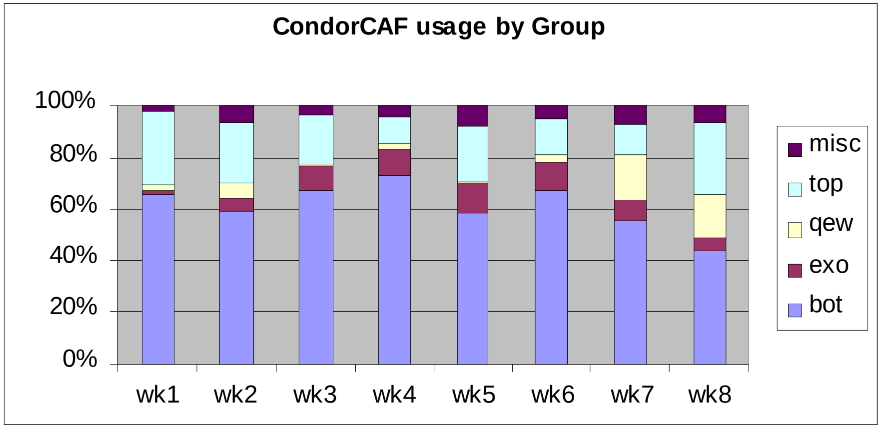
FNAL:
- > 10k jobs for ~100 users/day



CDF Analysis Farms

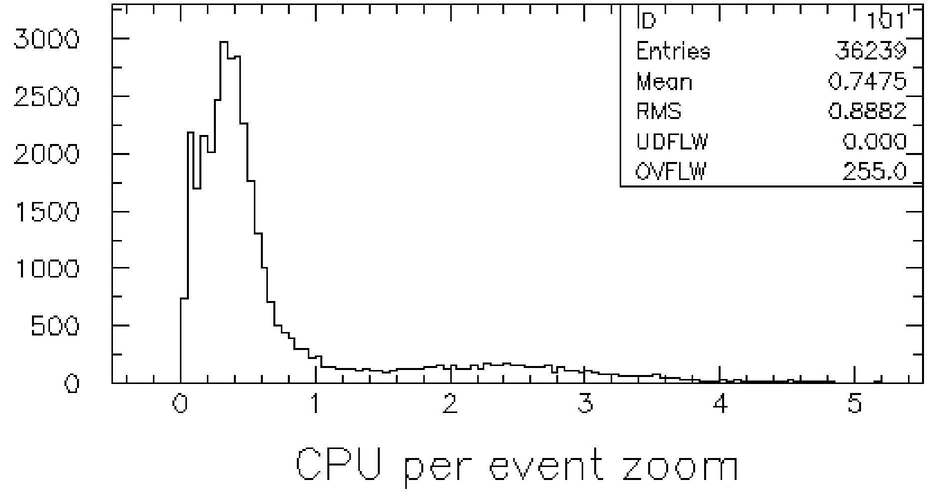
- Usage patterns at FNAL from summer of 2004
  - CPU by task
    - 50% of load in analysis of production output files
    - 20% in MC
    - Balance in ntuple analysis, other tasks
  - CPU by physics topic
    - B-physics group consumes majority of CPU cycles





CDF Analysis Farms

- User analysis on prod data
  - Average of 0.75 sec/event
  - About 20% use > 1 sec/evt
    - 40% of total prod data CPU
  - Event read + unpacking + minimal analysis
    - 0.06 sec/event

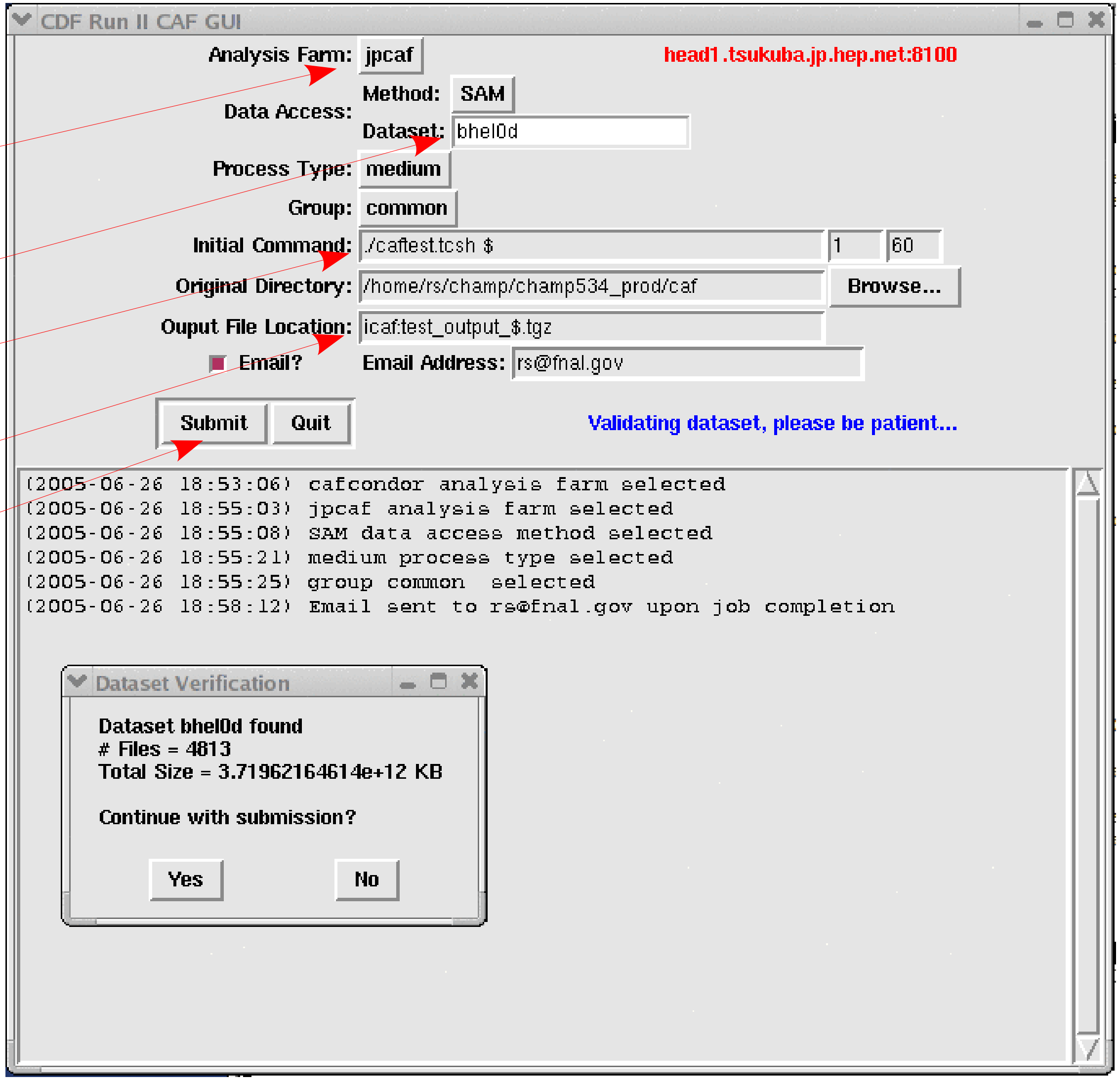


- What processing contributes to the tail?
  - Track re-fitting and vertex finding/fitting
    - Follows from needs of B physics and use of precision tracker
  - Both require full analysis framework

CDF Analysis Farms

- User's experience
  - Select site
  - Specify dataset
  - Startup script
  - Output location
  - Press “submit”

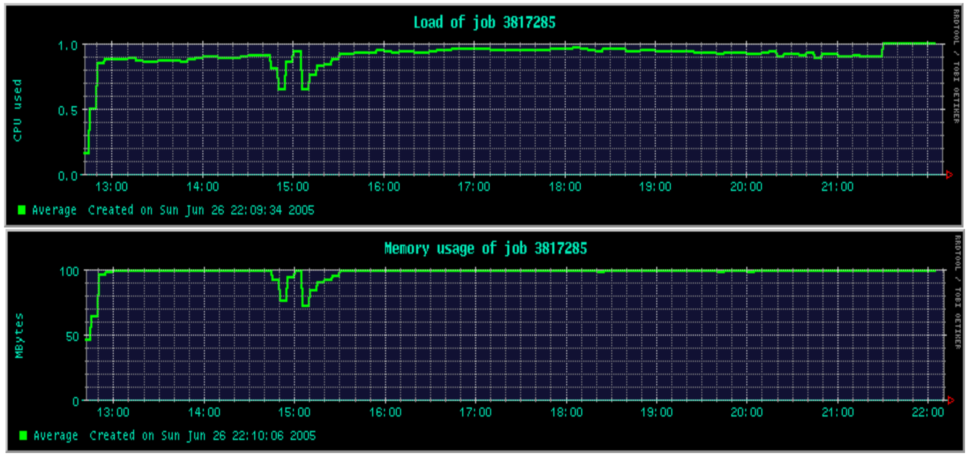
  User's context tarballed, sent to execution site



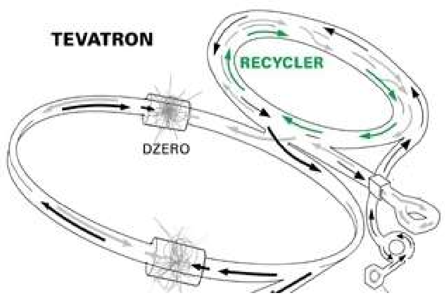
Same interface can be used for grid submission

CDF Analysis Farms

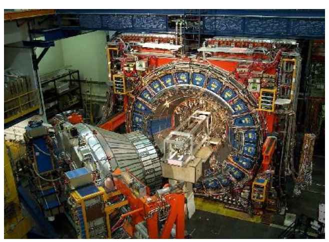
- User's experience
  - Monitoring
    - CPU, memory by process
    - Execution, return status
  - Control
    - Hold, resume jobs
    - Change execution priority for a process
    - Copy output to any machine with write access
  - Quasi-interactive features
    - Look at log file on a worker node
    - Directory listing in user's relative path
    - Connect debugger to a running process



Grid migration plans



CDF Detector



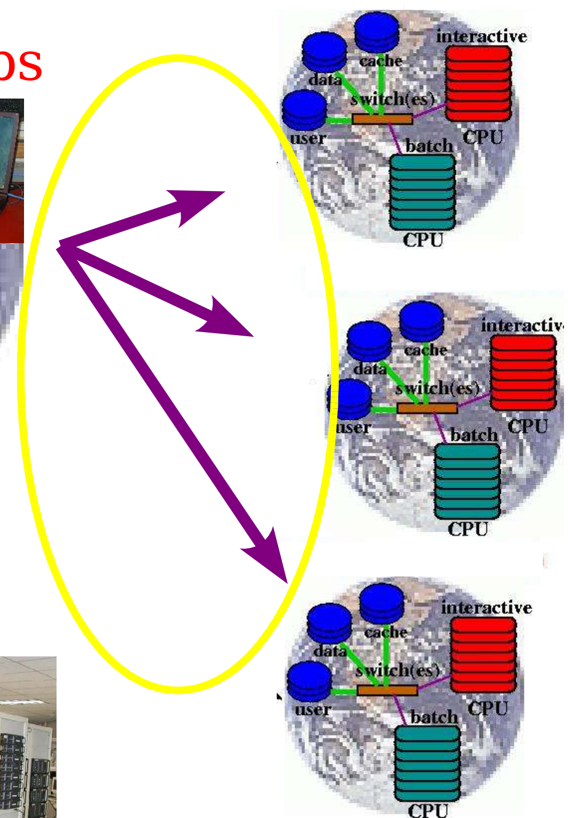
Production Farm

Reconstruction

User Desktops

Job development,
Ntuple analysis

Remote CAFs



Simulation
and Analysis

Robotic
Tape Storage

User Analysis

CDF Analysis
Farm (CAF)

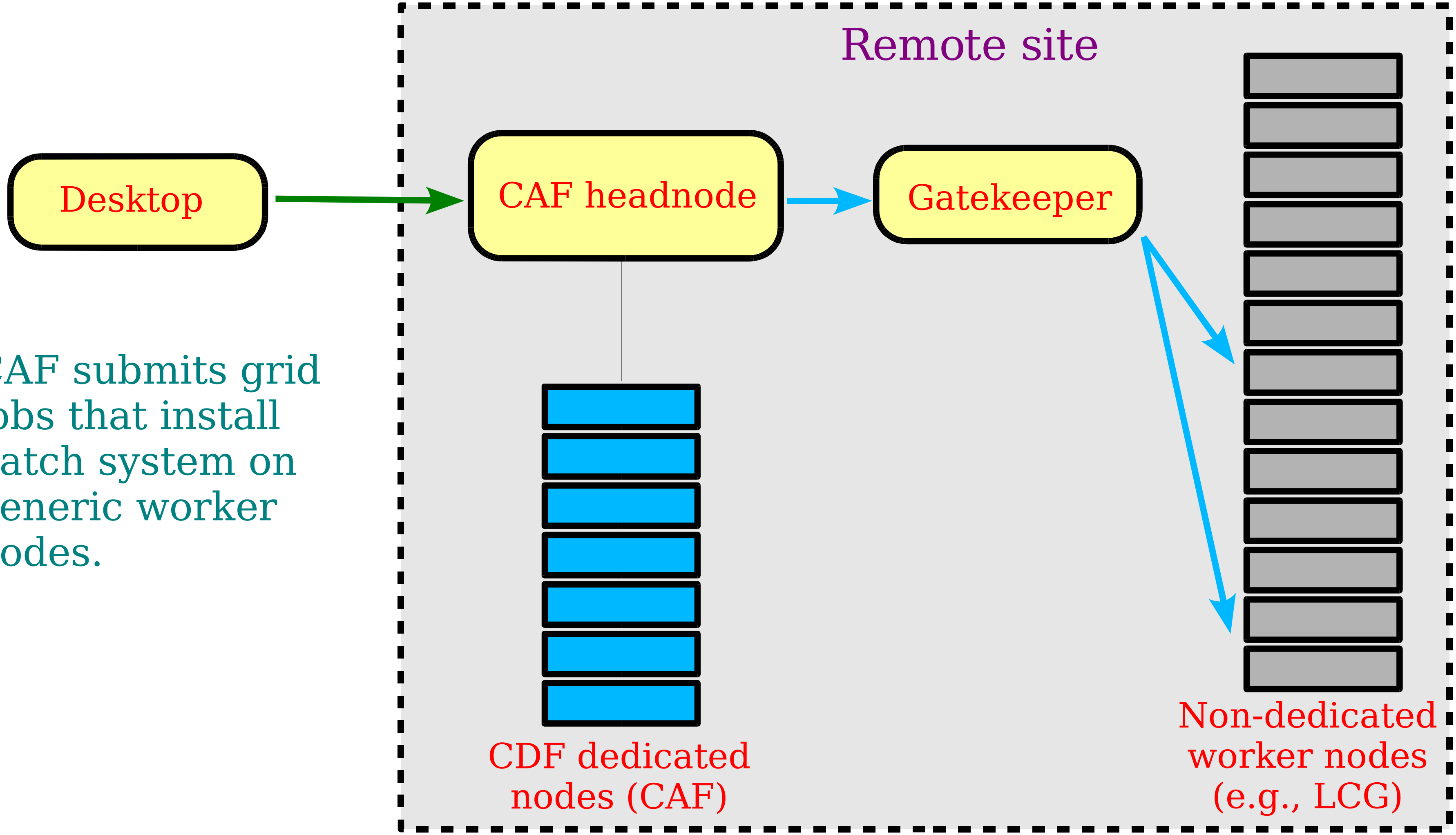
Grid migration plans

- Reasons to move to a grid computing model
  - Need to expand resources at FNAL and remote CAFs
    - Expect factor of eight more integrated luminosity
    - Will need to perform more analysis on remote CAFs
  - Most remote resources in dedicated pools
    - Only limited expansion possible in this model
    - May not be able to maintain access to existing resources
  - Resources at large
    - Estimated 30 THz currently in LHC and US-HEP grids
    - Small fraction of opportunistic access can be significant

Grid migration plans

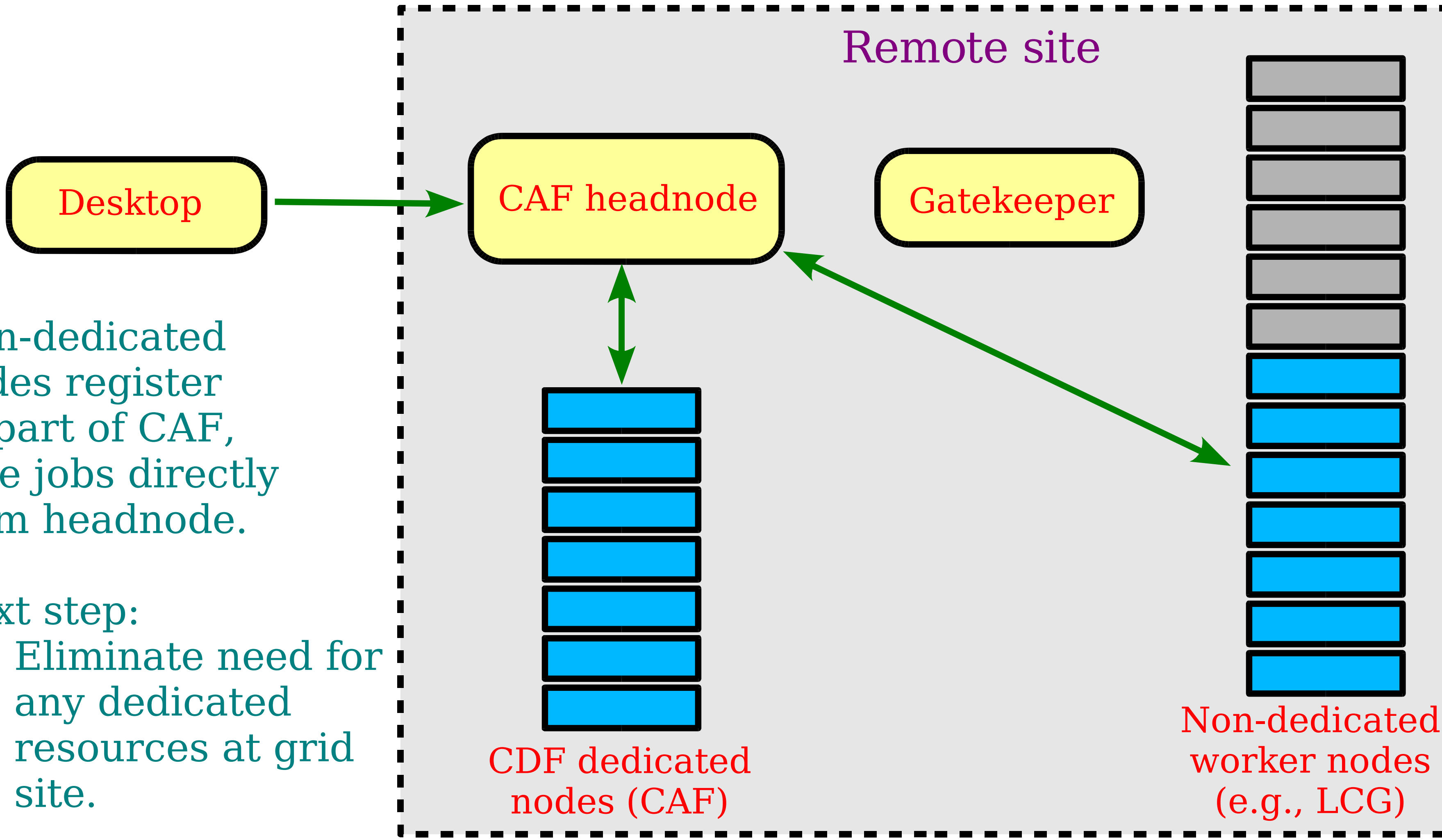
- Basic plan
  - Adopt incremental, staged approaches when possible
    - Partial solutions now to bridge time to develop for the long-term
  - Allow various levels of service to solve different problems
    - Predictable computing (production) vs. user analysis
  - Target European and US grid infrastructure aligned with other efforts at FNAL
  - Retain existing user interface

Grid migration: interim solution to eliminate dedicated resources



CAF submits grid jobs that install batch system on generic worker nodes.

Grid migration: interim solution to eliminate dedicated resources



Non-dedicated nodes register as part of CAF, take jobs directly from headnode.

Next step:

Eliminate need for any dedicated resources at grid site.

Grid migration plans

- On-going efforts
  - "Condor glide-in" for CAF
    - Remote CAF at CNAF in Italy uses this
    - Demonstrated opportunistic use of 1.3 THz of CPU
  - Re-implementing CAF using native grid tools
    - Eliminates need for any dedicated resources at grid site
    - Target user analysis applications

Event data model

- What is an EDM?
  - Set of structures for raw and reconstruction data
    - All stored within some larger, shared data structure
  - Associated interfaces, utilities to manipulate, serialize
  - Typically operates within a specific analysis framework
- Most simple example of an EDM
  - Ntuples
    - CDF physics groups supports several standardized ntuples
      - Vastly more efficient than all user-defined ntuples
      - Often created from data in coordinated fashion

Event data model

- Some features of EDM at CDF
  - Event data in fully featured C++ objects
    - Raw data objects are self-describing
      - Serialization automated for raw data objects
  - Objects cannot be modified once entered into event record
    - Retains history of event
  - Various general containers provided
    - Arrays of objects or references to objects
  - Utilities to locate objects based upon various criteria
  - Many “features” to prevent some common errors
    - Ex: difficult to have 3rd party change data beneath you
    - Many, despite benefits, are disliked by users

Event data model

- Common features to all objects in EDM
  - Unique ID number
  - Description string
  - “Process name” string
  - Print method, equivalence operators
  - Function to serialize data

Event data model

- Lessons from current experience (my own opinions)
  - Too much functionality in data objects
    - Ex: track objects
      - Include topological fitting interface, complex class heirarchy
      - Neither is used as intended
    - Can really be simple structures
  - EDM effectively tied to single analysis framework
    - Reconstruction tools that access EDM usable only in this context
      - Tracking, track re-fitting, vertex finding and fitting...
    - Problem largely stems from built-in serialization functionality
    - Should instead decouple reconstruction from any context
      - Write reconstruction interfaces to use simple structures
      - Make serialization an implementation detail

The best things we did

- Developed CAF and simple submission, monitoring tools for user analysis.
  - Made using large computing resources easy.
- Adopted structured data types for event data
- Established, maintained good physical design of software
- Defined lots of sensibly defined production output datasets
- Wrote a fast reconstruction

Summary

- CDF computing model has functioned well to provide needs to current time
  - Users can effectively utilize 5.6 THz of CPU distributed in many locations
  - Need to provide more user-level automation
- Much work to do to ensure systems will scale through the end of the run
- Grid migration will become an increasingly important component of computing model
- Simple, context independent EDM has good features for users

Summary

- Computing becomes more complex with the volume of data to be analyzed
  - Robust, scalable data handling is difficult
  - Distributed computing and emerging grid technologies
  - Other new technologies...
- Important to focus on making it easy for users to perform analysis within this hostile environment
  - Provide tools and automation to deal with large datasets and other common tasks
  - Keep primary user interfaces — EDM, data handling, job submission tools — simple
  - EDM, reconstruction, analysis tools should be context indep.

The end

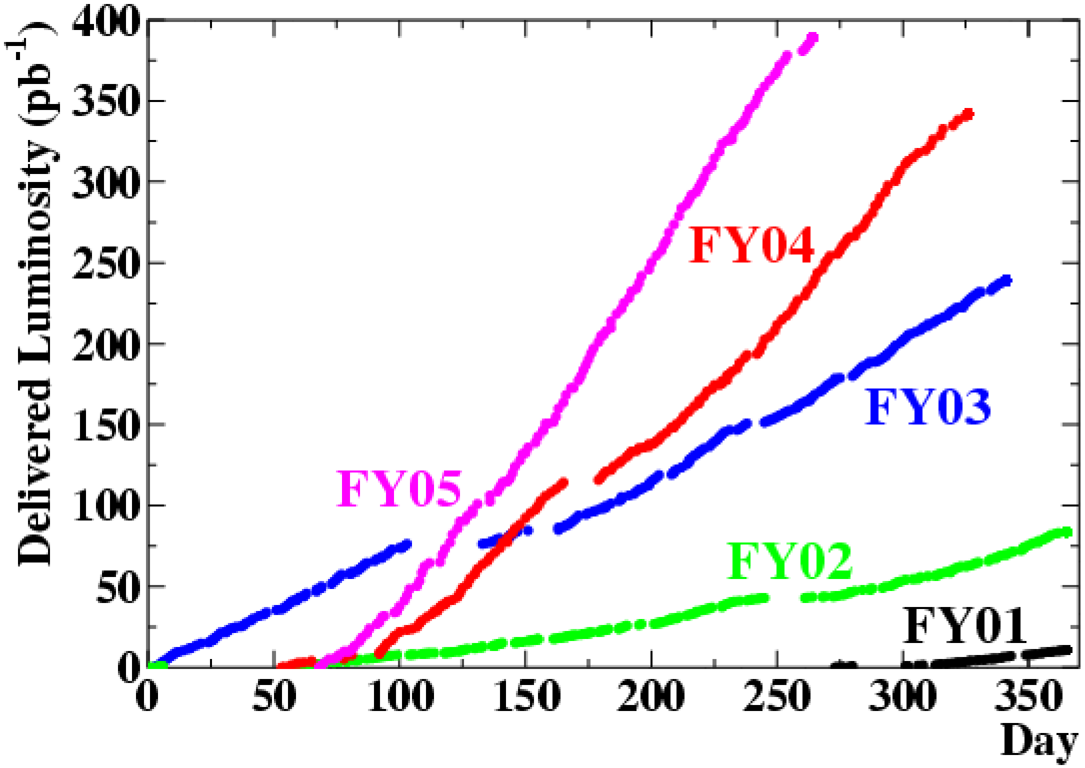
Backup slides

Introduction to the computing model

- Run II delivered luminosity

Rate into high-Pt datasets increasing by factor of two every year

Expected to increase another factor of 2.5 by FY2007

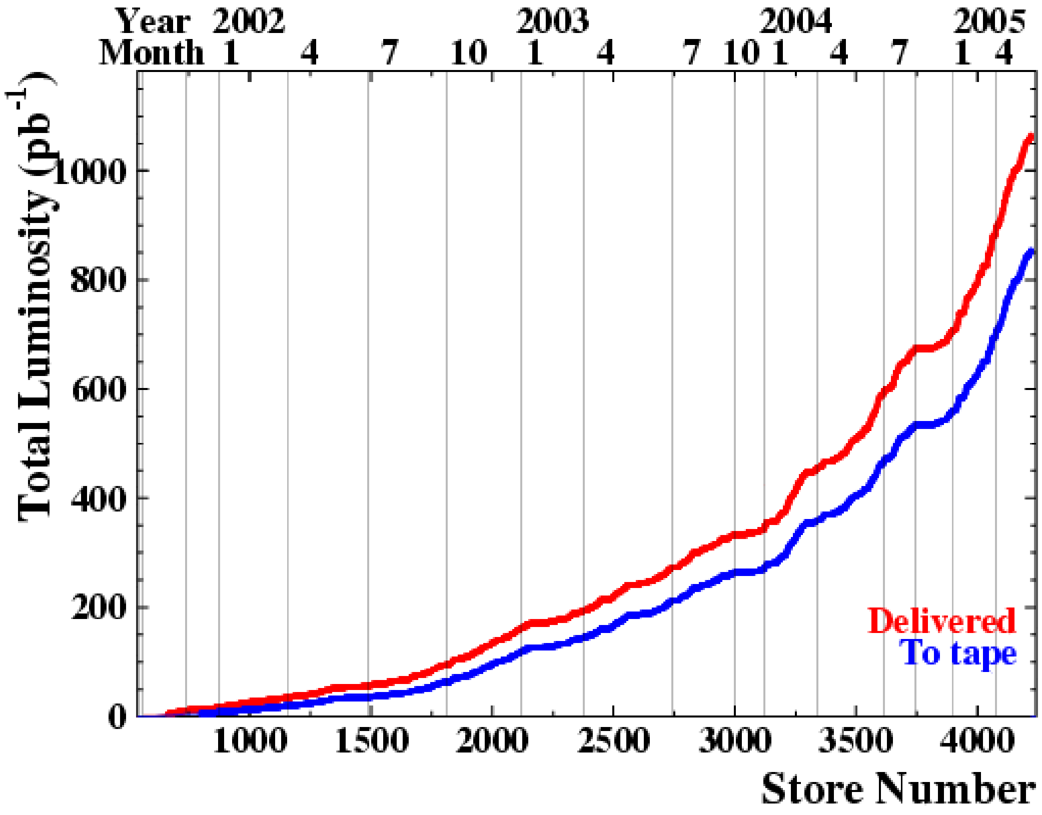


Introduction to the computing model

- Run II delivered, logged luminosity

Over 1 fb-1 delivered

About 850 pb-1 acquired



Computing requirements

•

FY	Int L. (fb^-1)	Evts (10^9)	Peak rate (MB/s)	Peak rate (Hz)	Ana (THz)	Reco (THz)	Disk (PB)	Tape I/O (GB/s)	Tape Vol (PB)
	Assumed conditions				Total requirements				
03A	0.30	0.6	20	80	1.5	0.5	0.2	0.2	0.4
04A	0.68	1.1	20	80	2.3	0.7	0.3	0.5	1.0
05E	1.2	2.4	35	220	7.2	1.4	0.7	0.9	2.0
06E	2.7	4.7	60	360	16	1.0	1.2	1.9	3.3
07E	4.4	7.1	60	360	26	2.8	1.8	3.0	4.9

A = actual (FNAL o E = estimated