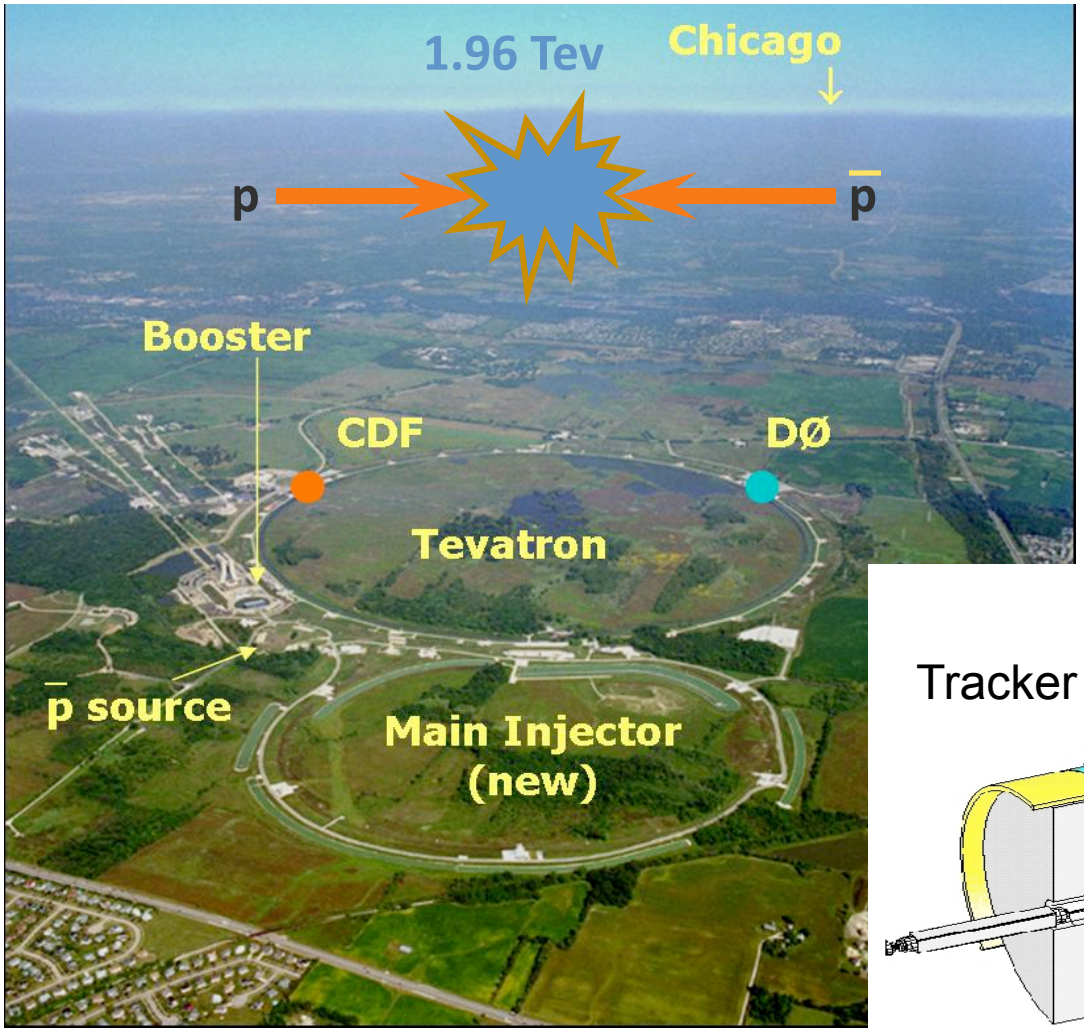
The Collider Detector at Fermilab (CDF) Scientific Collaboration's experience with Condor

Doug Benjamin

Duke University

Collider Detector at Fermilab (CDF)

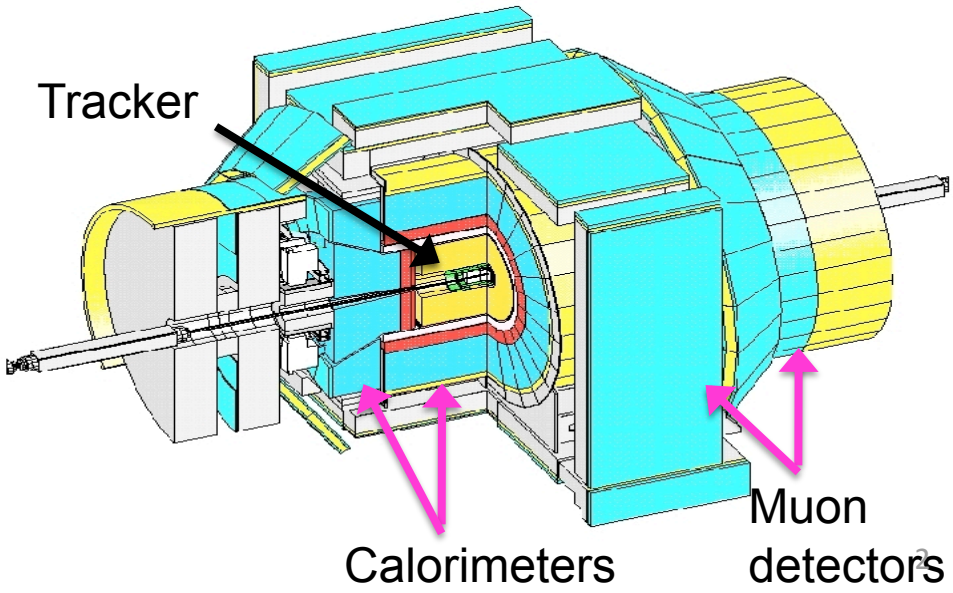


CDF :

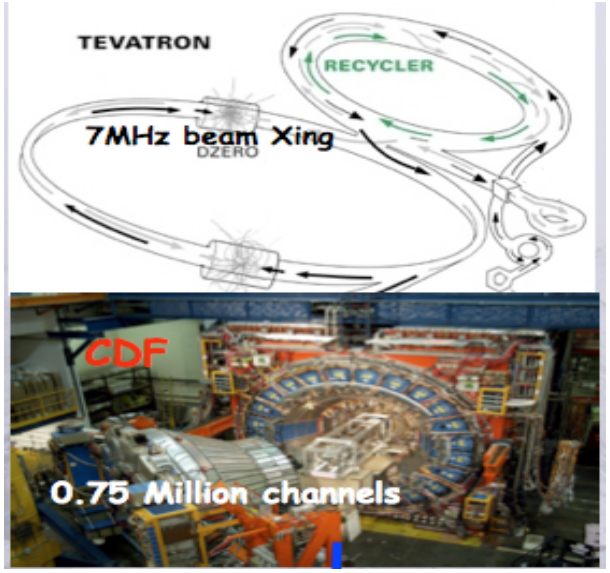
- Large multipurpose Particle Physics Experiment at Fermi National Accelerator Lab (Fermilab)

-Started collecting data in 1988

-Data taking will continue until at least Oct. 2009 (with desire to extend another year)

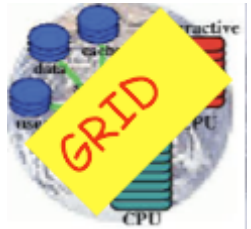


Computing Model

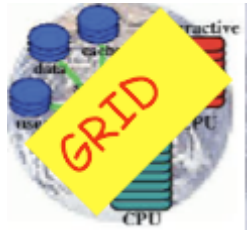


dCAF

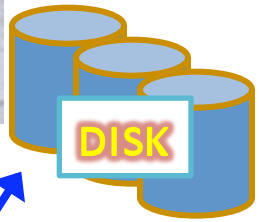
user jobs



Data



Data



Data

Data

Data

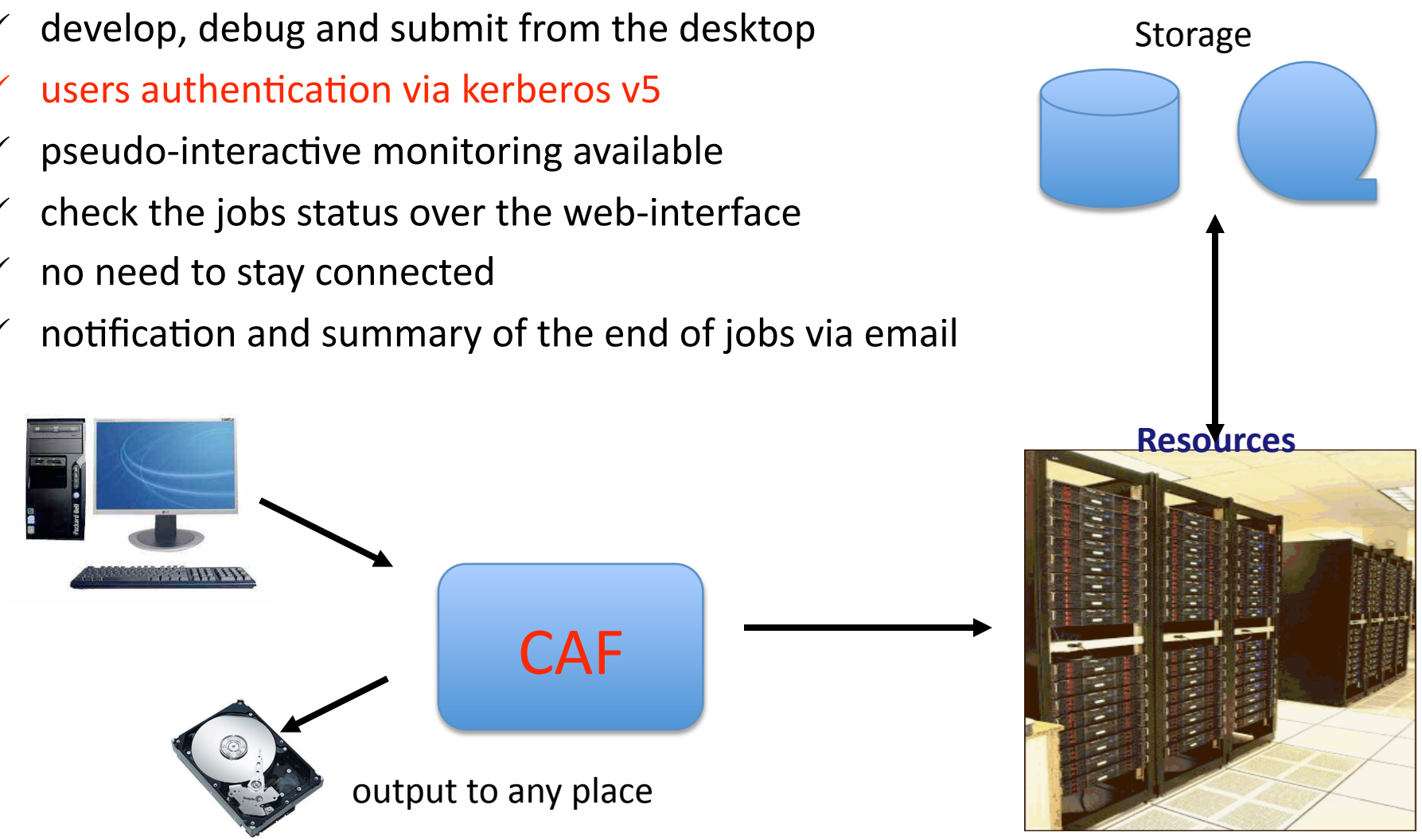
CAF

Robotic tape storage

- ✓CAF :production/User analysis FNAL hosts data : farm used to analyze them
- ✓decentralized CAF (dCAF): mostly Monte Carlo production

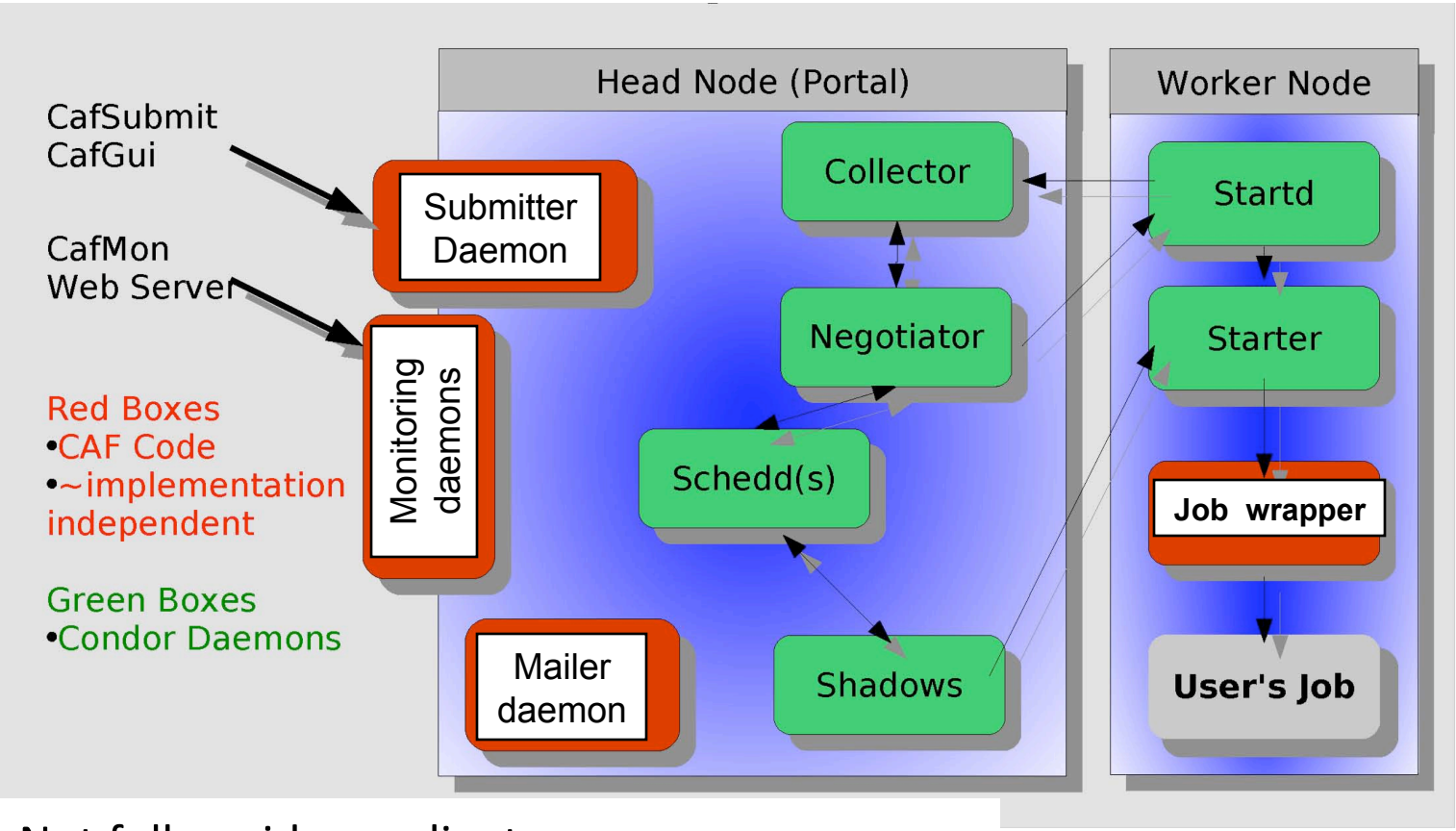
CAF : CDF Analysis Farm

- ✓ develop, debug and submit from the desktop
- ✓ users authentication via kerberos v5
- ✓ pseudo-interactive monitoring available
- ✓ check the jobs status over the web-interface
- ✓ no need to stay connected
- ✓ notification and summary of the end of jobs via email



Initial Condor implementation

- Dedicated pool of machines - Very successful
- In production since 2004



Not fully grid compliant

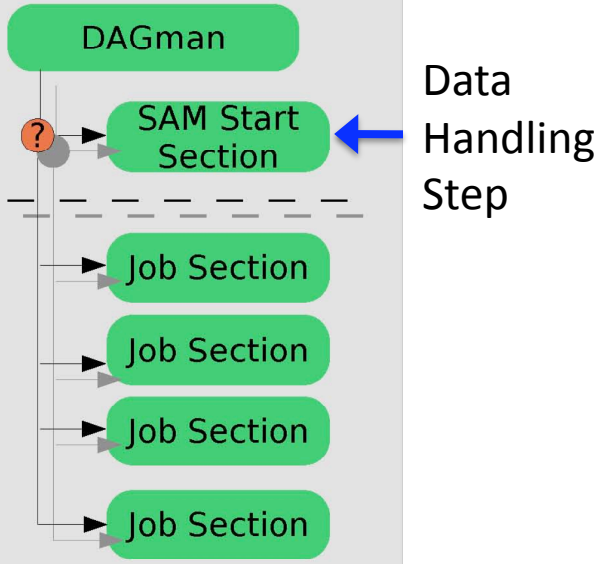
Remote sites used dedicated computer farms

Physicist view vs Condor's view of Computing job.

- We think of job as task with the related parallel tasks (Job sections)
- Condor – each job section – independent Job

DAGMan – allows CDF Users – Condor to work together

- A Condor "Scheduler Universe" job
  - This is jobs that manage other jobs
  - Runs on head node
- DAGman submits individual job sections automatically
  - Submission can be conditional
    - E.g. SAM start section first, other sections only if start works
- DAGman job finishes when all sections are done



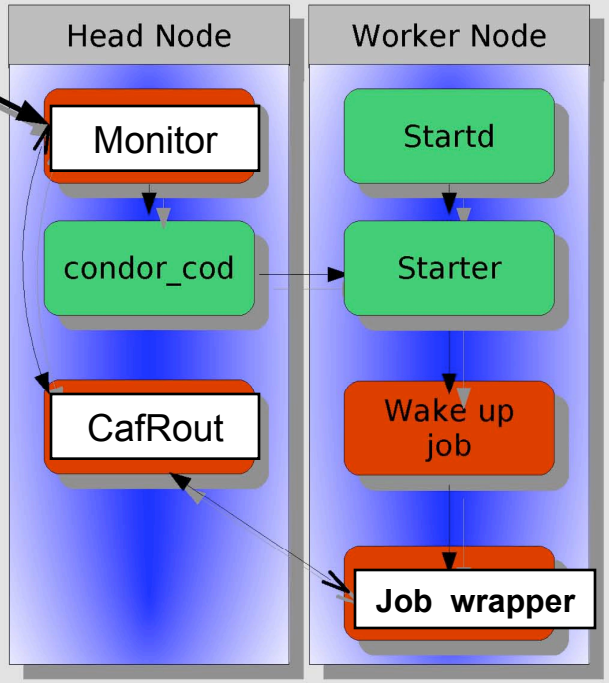
Data Handling Step

Interactive User Job Monitoring

- Condor Computing on Demand (COD) is used to allow users to monitor/interact with their job

CafMon

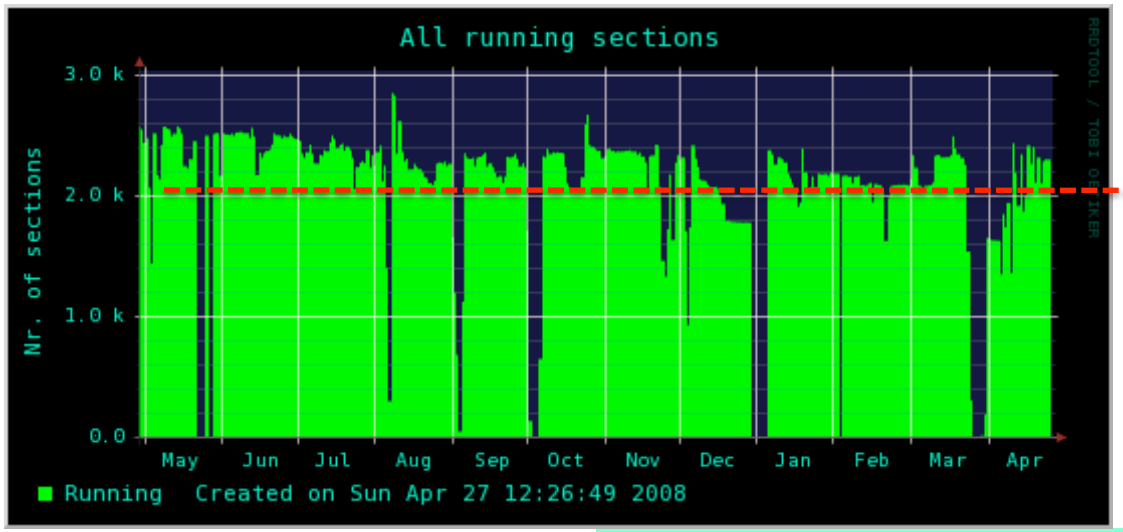
- Establishes connection with CafRout
- Uses Condor's Computing On Demand (COD) system
  - Uses the monitoring libraries to find VM
- Monitor wakes up CafExe using COD
- CafExe calls back CafRout
- Now connection runs from CafMon to CafExe



CDF Code and Condor Code together give the users tools needed to get their jobs efficiently

w/ Condor COD Users can:
- look their working directories and files
check their running jobs (debug if needed)

CDF's Dedicated Condor Pools very successful



> 2000 slots - 1 year

Condor has made it easier for us to allocate resources (Quotas)

Sections by Accounting Group(ordered by AcctGroup)

Accounting Group	Quota	Running	Assigned	Idle	Wait	Held	Completed	Removed	Total	Jobs	
common	203*	354	0	2	2316	0	1612	768	5052	44	History
group_highprio	1822	1566	0	0	19	0	898	11	2494	15	History
group_italy	0	34	0	0	0	0	351	2	387	3	History
group_uk	0	0	0	0	0	0	5	35	40	1	History
Total (4)		1954	0	2	2335	0	2866	816	7973	63	

* - Estimated, no real quota

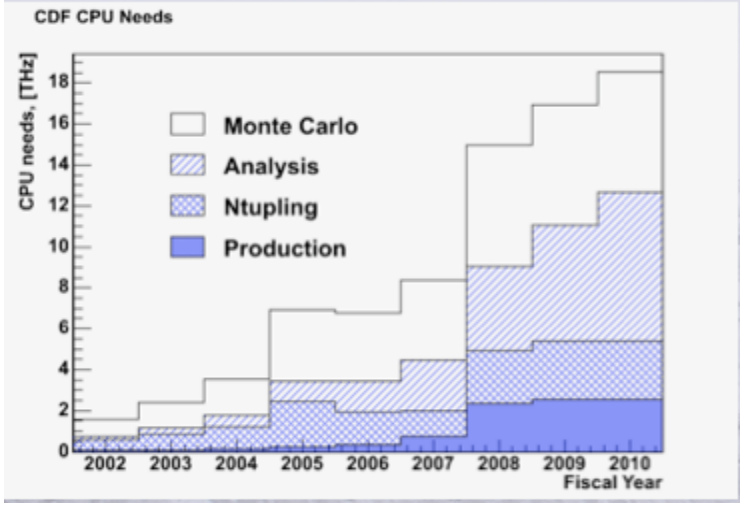
Accounting User Sections (ordered by AcctUser)

Accounting User	Priority	Running	Pending	Completed	Removed	Total	Jobs
common.balvarez	1627.55	25	0	663	0	688	12
group_MCprod.benjamin	5167.37	41	0	780	1	822	7
group_MCprod.njg	466.20	3	0	723	118	844	2
group_MCprod.strolog	576.08	3	0	731	0	734	1
Total (4)		72	0	2897	119	3088	22

Condor priority mechanism Dynamically adjusts for usage

CDF Run II computing transition to Grid

- Initially distributed computing in dedicated CAF's (North America, Asia and Europe)
- dedicated resources → shared grid resources
  - all new computers at FNAL in grid based farms (2006)
- Data volume will **grow 2.5 times**
- Need more CPU power than can be found at FNAL
  - Missing resources have to be found on the grid



CDF CPU requirements by Fiscal Year – and by activity

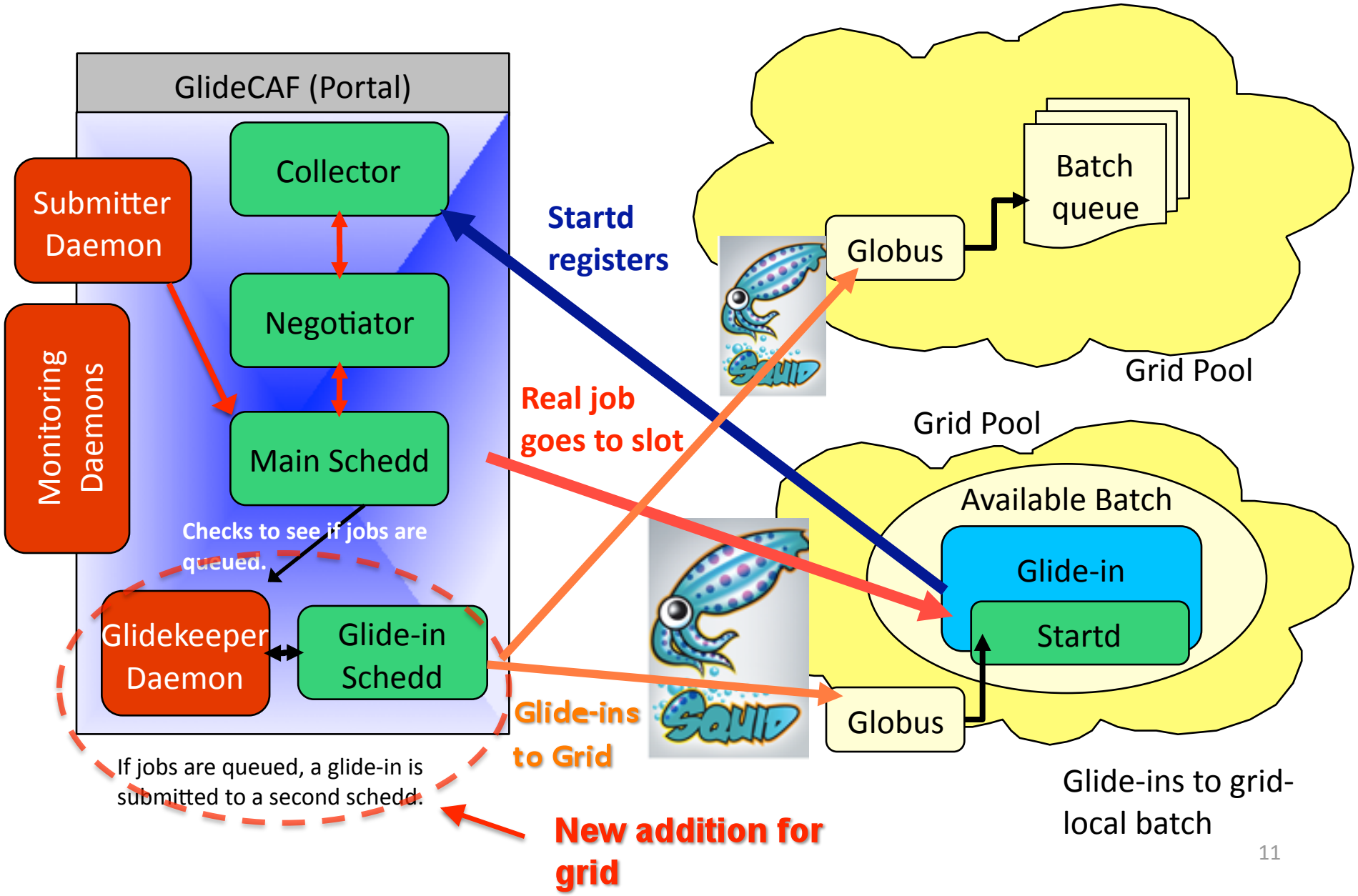
All Monte Carlo activities on the Grid

- Simulated data
- Additional calculations
  - due to advanced analysis techniques

GRID Transition requirements and solution

- Requirements
  - Minimize impact on CDF user – so user can focus on the science not the computing
  - Small changes to CDF middleware
  - Continue to use the Kerberos V5 authentication for the users
    - X509 Authentication used across grid components
  - users needs to be able to run over data @ FNAL
  - submission and access must be centralized
  - Run via Globus at each grid site
- Solution
  - Condor glideins using Condor startd's run at remote sites
  - A Virtual Condor pool is setup
    - User jobs run like a dedicated Condor pool

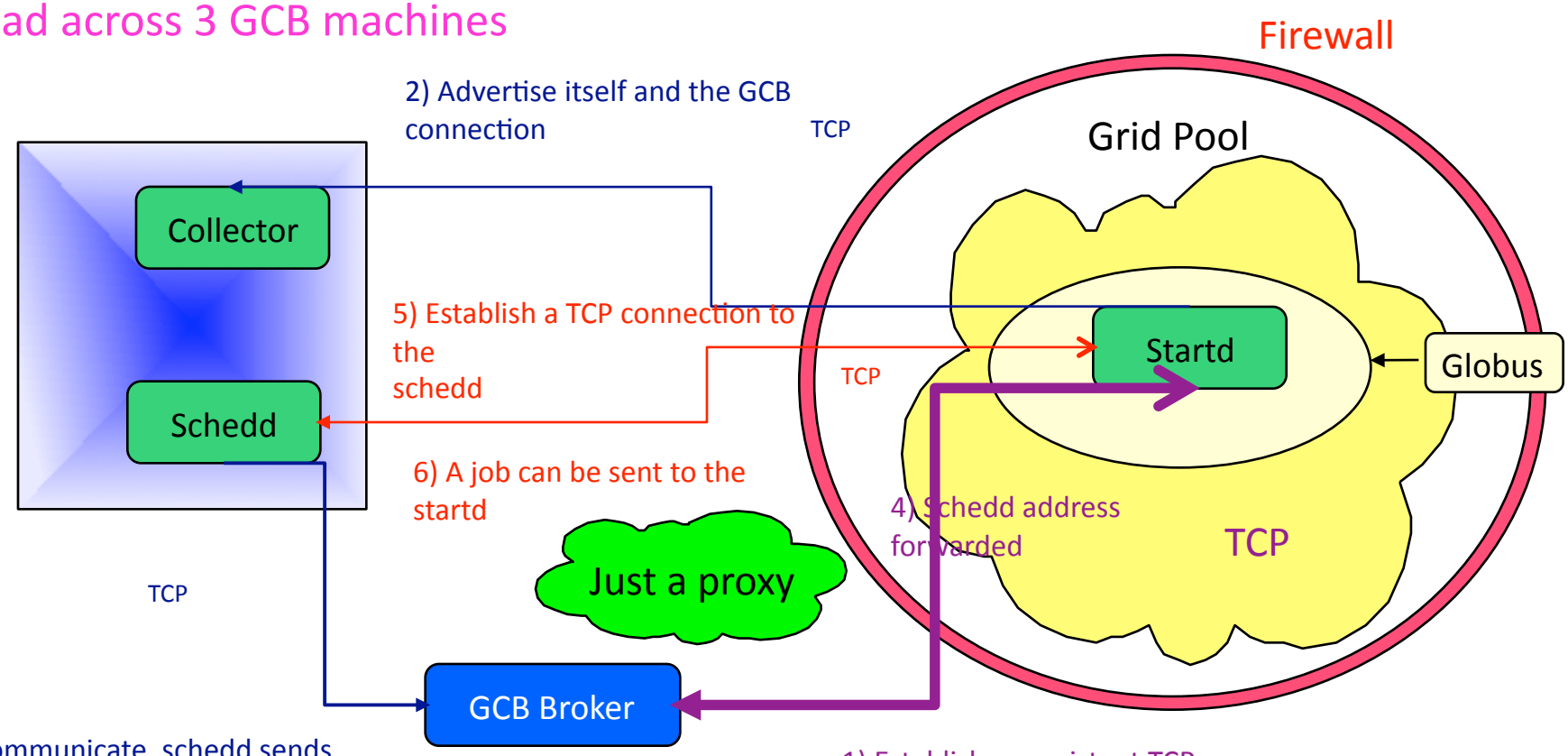
GlideCAF overview



Firewall challenge - Condor GCB solution

(Generic Connection Brokering)

Works well for CDF - < 1000 connections
spread across 3 GCB machines

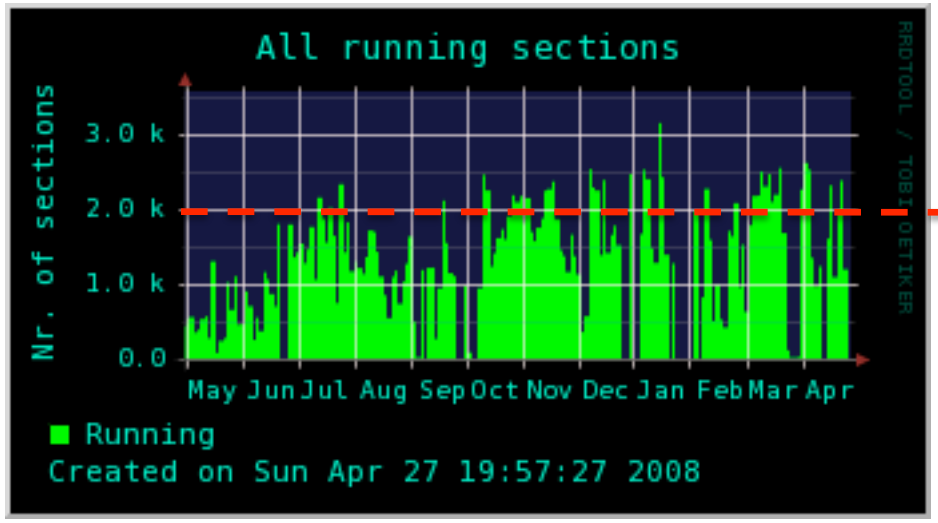


Slide from Igor Sfiligoi –CDF week- Elba 2006

Glidein Security Considerations

- Glidein submitted to Globus resource starts with pilot job credentials
  - CDF end users credentials are transferred to glidein when it receives the user job
  - Some sites require the user job to run with its own credentials not the pilot job
- gLExec program is used run the user job with the users credential. (see previous Europe Condor week 2006 for more details)
- FNAL requires that CDF jobs use gLExec as part of the authentication chain.

GlideCAF performance on Fermigrid



2000 user slots
1 year

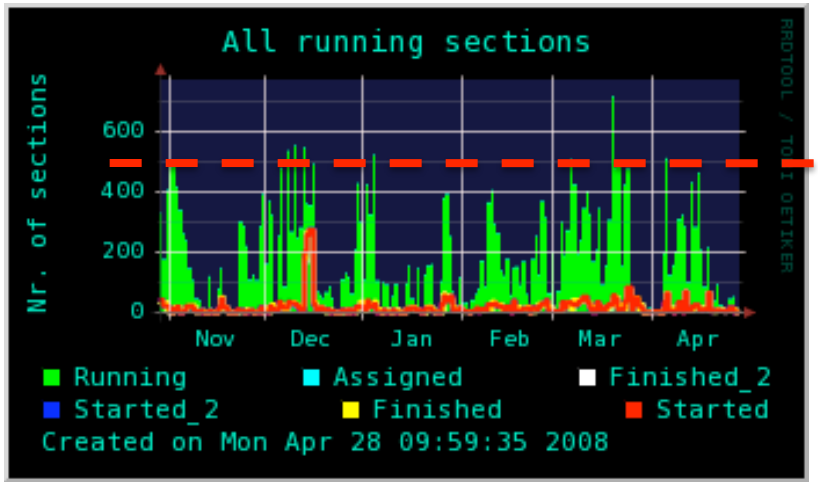
- In past year or so installed > 6 different Condor versions. Had to use 6.9.1-6.9.5 then 7.0.0, 7.0.1.
  - Upgrades required for bug fixes that we often found after deployment
- In Oct had to split collector/negotiator to separate node we could not fill Fermigrid compute element with glideIns

  (N.B.– dedicated Condor w/o Glidein all on one node)

GlideCaf experiences

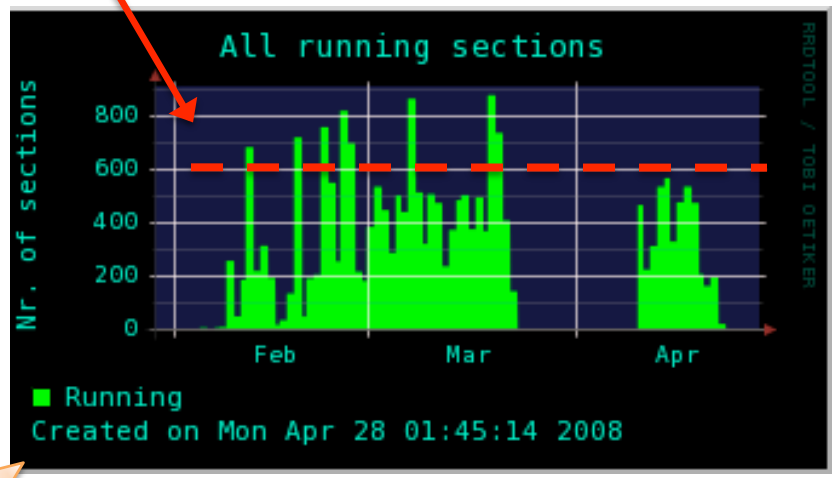
- Overloaded head nodes susceptible to losing glideins
- Less efficient at using similar resources
  - need more hardware to provide same number of slots
- Added Grid layer makes debugging problems more challenging
  - Requires help from Grid Site admin.'s
- With each new release new ``features'' found
  - E.g. Condor 7.0.1, 7.1.0 – broke COD – glidein's were killed after COD command – had to revert to 7.0.0 for glideins
  - Need to use the new releases (Condor team diligence in bug fixes) 
  - Request that Condor prereleases be available for testing
    - CDF continues to find new features not seen by others

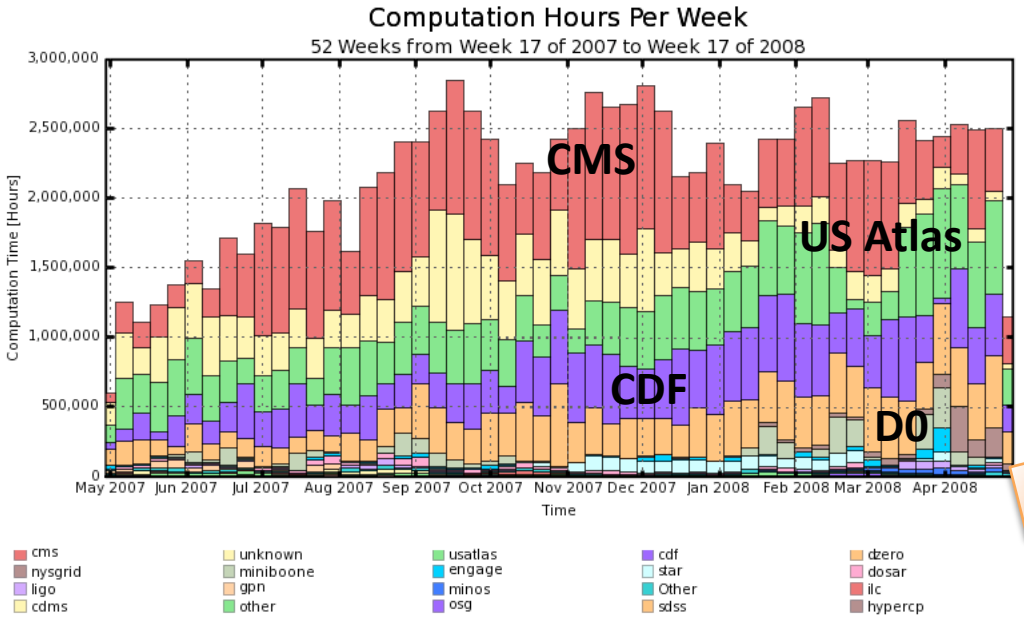
CDF GlideCAF performance in Europe,Asia and on Open Science Grid



Asian GlideCAF
500 slots 6 mon.

Europe GlideCAF – CNAF Italian T1 –
600 slots 3 mon.





Open Science Grid

Future – GlideCaf improvements

- Migrate to GlideinWMS
- Installation of more powerful head nodes
  - 8 cores, 2 GB/core
- Migrate Stand-alone Condor Pool into Glidein Pool
- Further Scaling improvements
  - Investigate moving secondary schedd's from head node to additional hardware
  - Condor Team improvements in schedd performance helps to reduce the need for migration

Conclusions

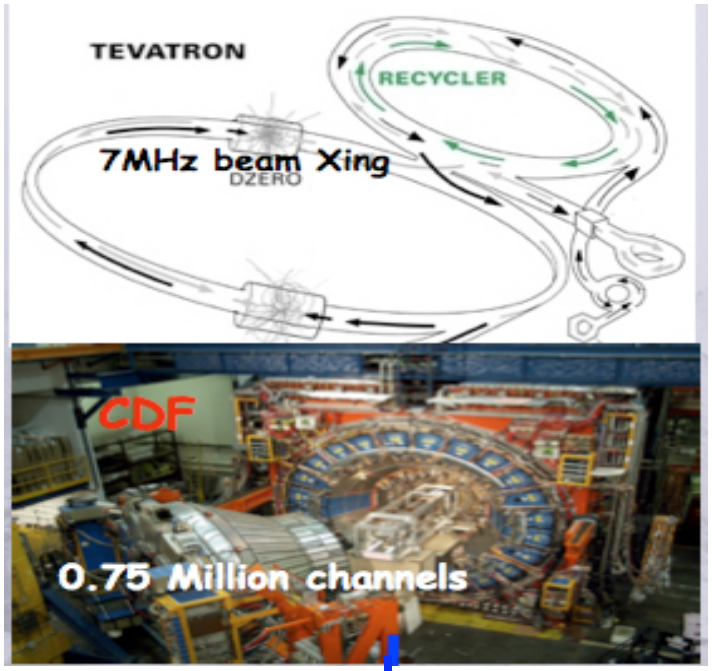
- CDF Computing continues to be very successful as a result of our use of Condor
- Through the use of Condor glideins were able to transition to the Grid with little/no impact on the Physics analyzing data
- Continued collaboration with Condor team desired (Thanks for all your help)

Acknowledgements:

- Previous CDF CAF developers – Frank Wuerthwein, Mark Neubauer, Elliot Lipeles, Matt Norman and Igor Sfiligoi
- Current CDF CAF team – DB, Federica Moscato, Donatella Lucchesi, Marian Zvada, Simone Pagan Griso, Gabrielle Compostella, Krzysztof Genser
- Fermigrid Department – especially Igor Sfiligoi and Steve Timm

Backup Slides

Computing Model



CDF Level 3
trigger ~ 100 Hz

Data

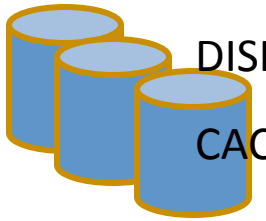
Data

Data Handling service

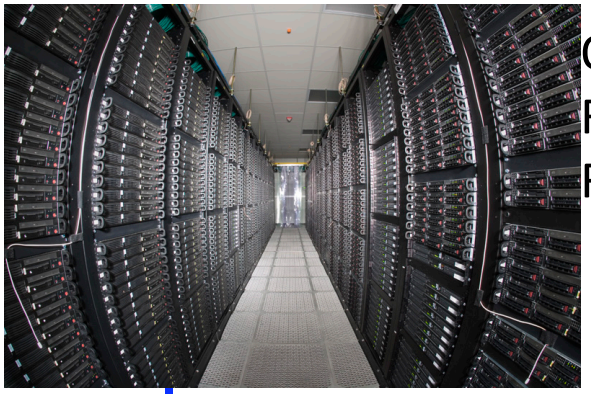
Archival Storage
(tapes)

Data

Data

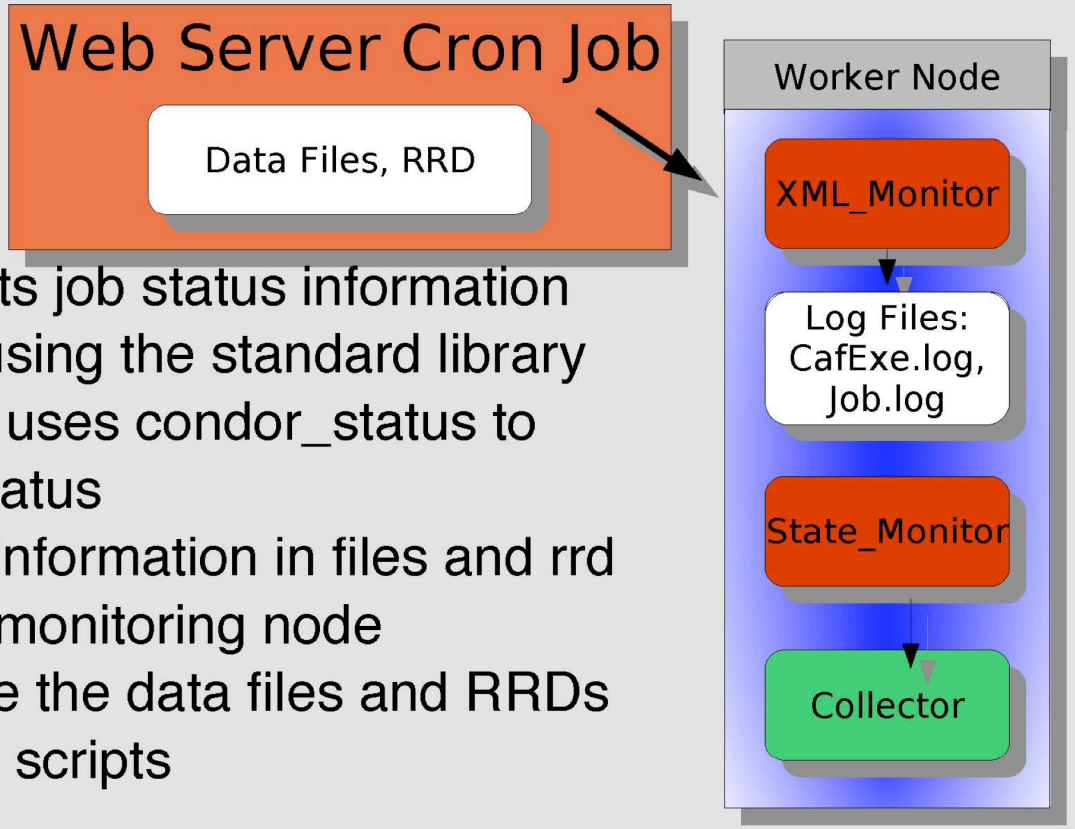


Offline



Offline
Reconstruction
Farm

Passive Web Monitoring



- Monitor extracts job status information from log files using the standard library
- State_Monitor uses condor_status to collect node status
- Cron jobs put information in files and rrd databases on monitoring node
- Web pages are the data files and RRDs served by CGI scripts

Condor daemons are queried to job status, user priorities

- Provides for users to see the history of their jobs
- Provides a mechanism – Data reconstruction teams to monitor their work and submit jobs as needed

CDF Dedicated Condor Pool Head Node Configuration

- Hardware - Dell PowerEdge 2850
  2 - Xeon 3.4 GHz (hyperthreaded) memory - 8GB
- Software - Scientific Linux Fermilab – 4.5 – Condor 6.8.6
  - Tried Condor 7.0.1 but collector problems forced a roll back
  - Our use of Kerberos authentication makes us different from many others – We often discover new behaviors in Condor
- Condor Daemons
  - Collector, Negotiator, Primary Schedd (DAGman), 4 Secondary Schedd's (User job sections), Shadows (up ~ **2400** user sections)
- CDF Daemons
  - Submitter, Monitor, Mailer

CDF Fermigrid Glidein Pool Head Nodes Configuration

- Collector Negotiator Node:
  - Worker node class hardware- dual HT Xeon – 8 GB RAM
  - Condor 7.0.1 - Condor Collector/Negotiator Daemons
- Job and Glidein Submission Node:
  - Hardware - Dell PowerEdge 2850
    2 - Xeon 3.4 GHz (hyperthreaded) memory - 8GB
  - Software - Scientific Linux Fermilab – 4.5 – Condor 7.0.1
    - Condor 7.0.0 for glideins Condor Daemons
  - Condor Daemons
    - Primary Schedd (DAGman), 4 Secondary Schedd's (User job sections), Shadows (up ~ **2700** user sections), Glidein schedds and Shadows
  - CDF Daemons
    - Submitter, Monitor, Mailer, Glidein submitter