



CDF CAF Project Status



Mark Neubauer

University of California, San Diego

- **CAF Services**
- **System implementation**
- **Milestones/deployment**
- **Performance/Utilization**
- **Recent enhancements**
- **Plans**



CD Project Status Meeting, 12/2003



CAF Services



Batch farm w/ high bandwidth access to DH resources

- **Job submission from desk/laptop anywhere**
- **Fair share of CPU resources for users and groups**
- **Interface to Data Handling services**
 - **Metadata catalog (DFC/SAM)**
 - **Data delivery to job (dCache/SAM/rootd)**
 - **Job output to scratch disk or tape archive (work in progress)**
- **Detailed summary of job after completion sent via email**
- **Web & command-line-based control & monitoring tools**



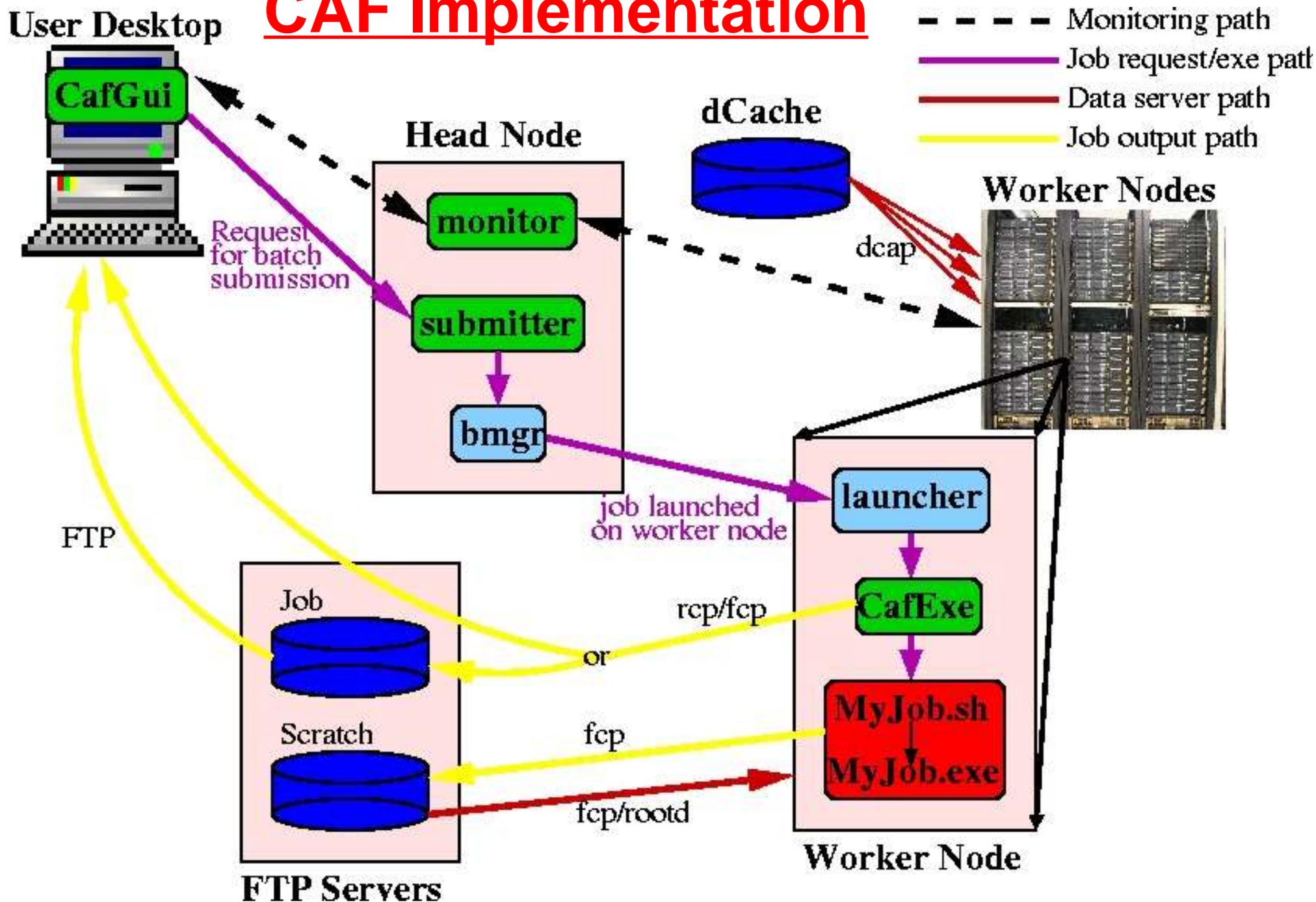
CAF Control/Monitoring Tools



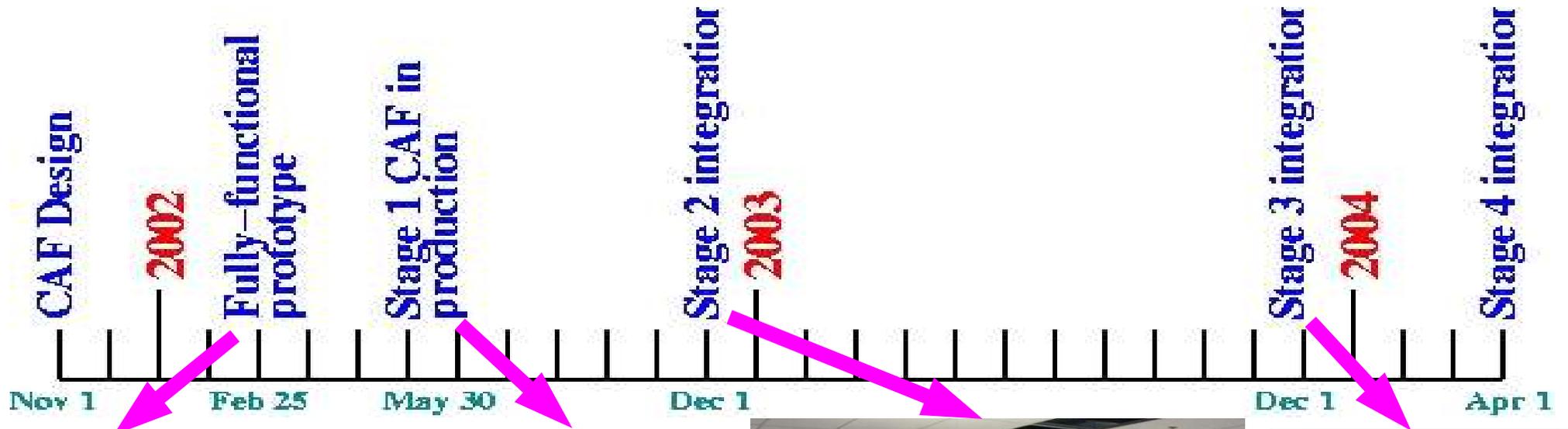
- CafMon jobs** Job list/status
- top** Show resource utilization of JS
- dir** File listing in working directory of JS
- head/tail/cat** Peek at files in working directory JS
- node** Return node on which JS is running
- log** Display progress of JS from CAF perspective
- kill** Kill a job or JS
- hold/release** Hold or release queue for launching new JS
- hostdir** File listing in absolute path of any CAF node
- cprio** Change relative priority of JS within a queue
- debug** Connect a debugger session (gdb) to JS process

JS = “job segment”, a parallel instance of a CAF job

CAF Implementation



CAF Milestones



ProtoCAF



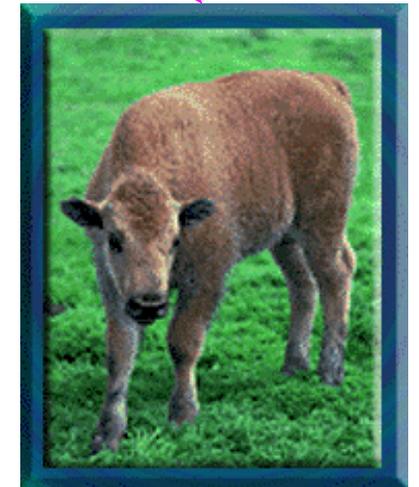
Stage 1

68 duals -> 176 GHz
16 FS -> 35 TB



Stage 2

256 duals -> 920 GHz
77 FS -> 169 TB
648 CPUs, 1.1 THz, 204 TB

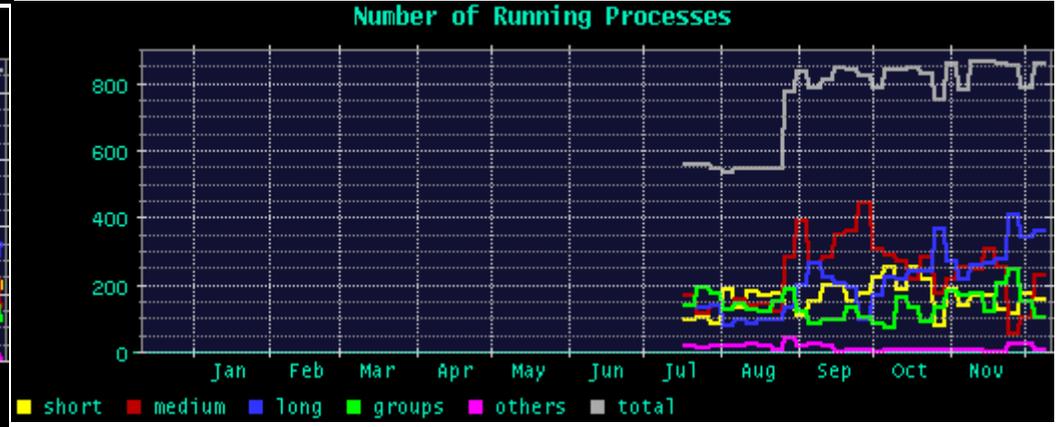
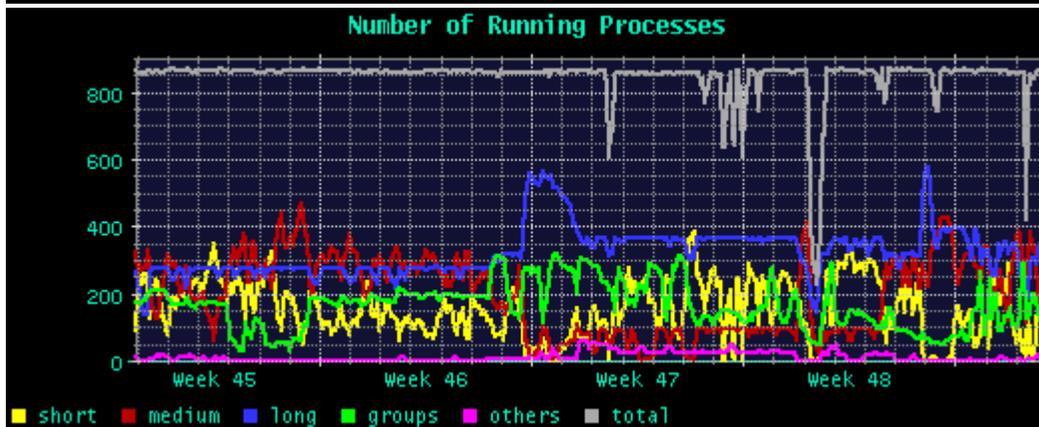
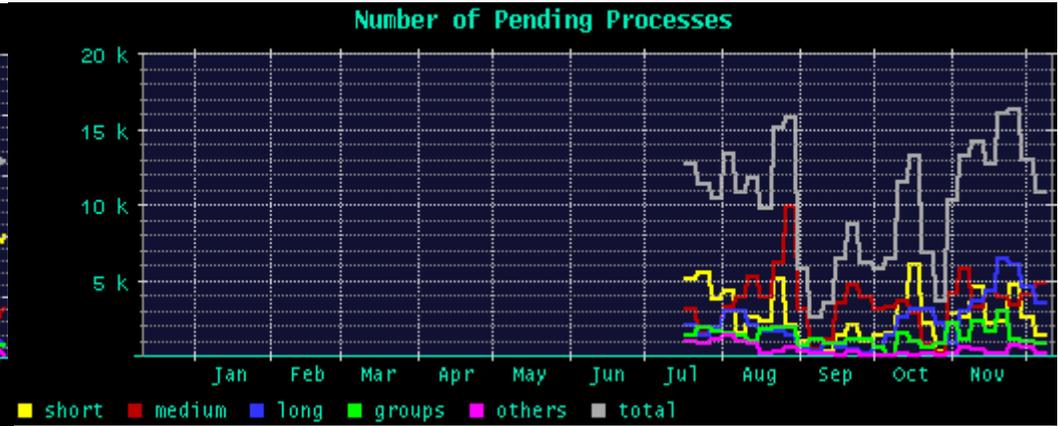
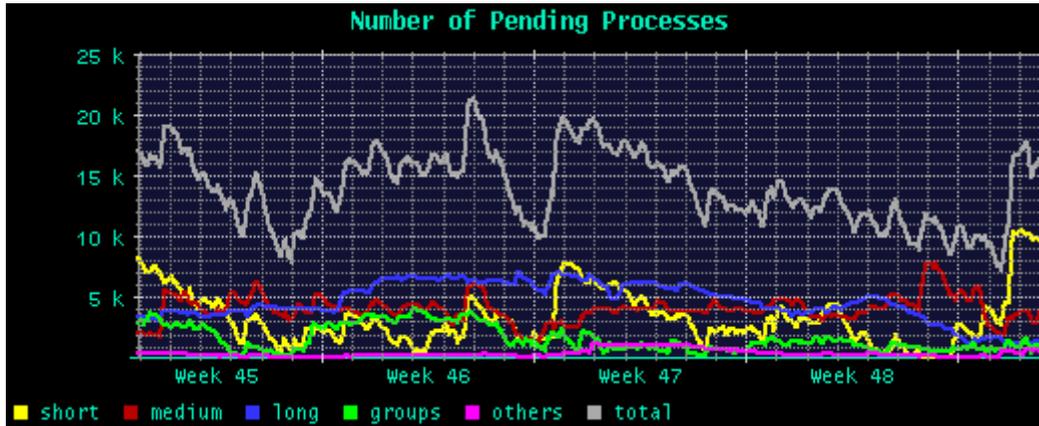


Stage 3

242 duals -> 1290 GHz
18 FS -> 90 TB
1132 CPUs, 2.4 THz, 294 TB

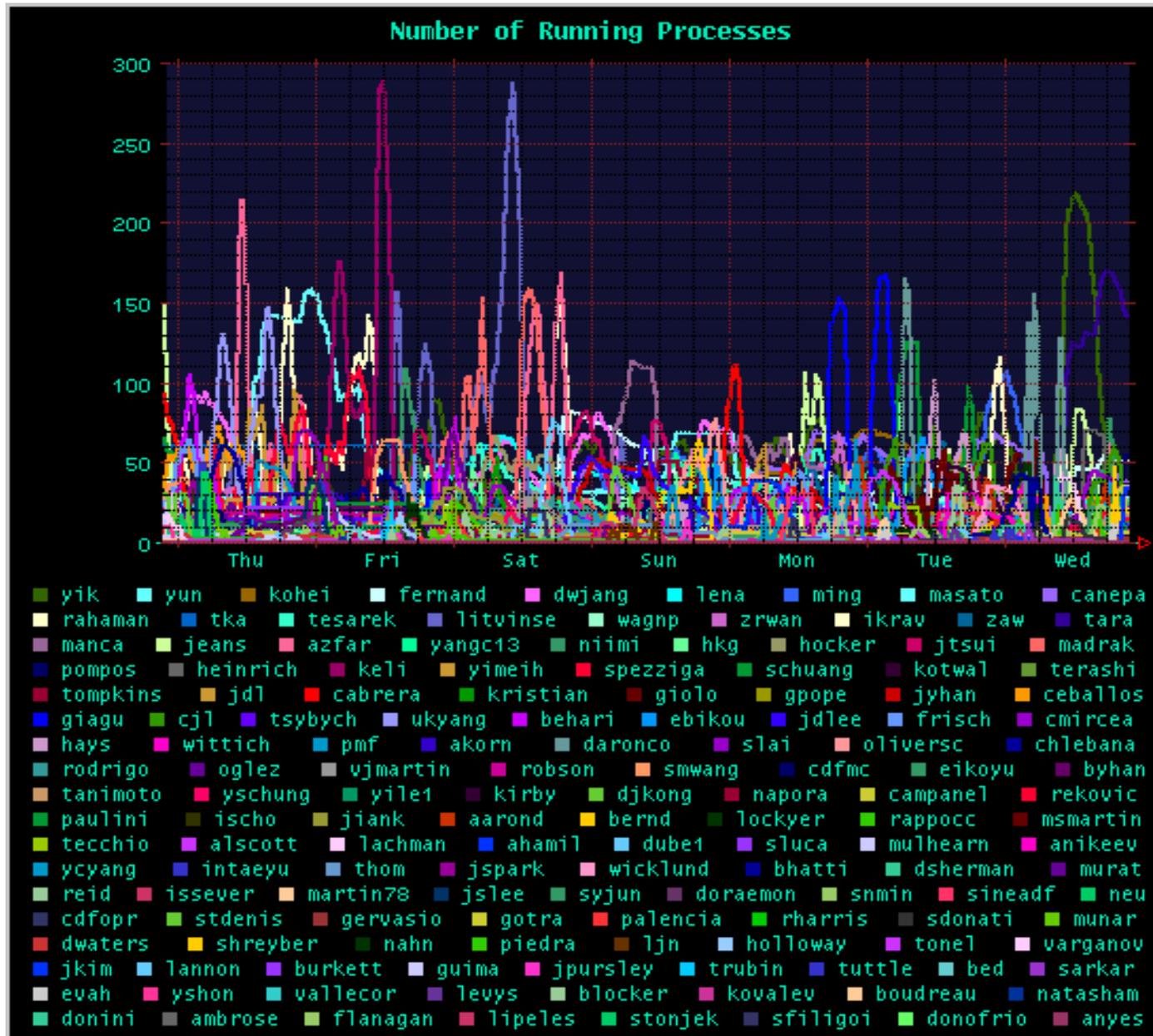


CAF Utilization





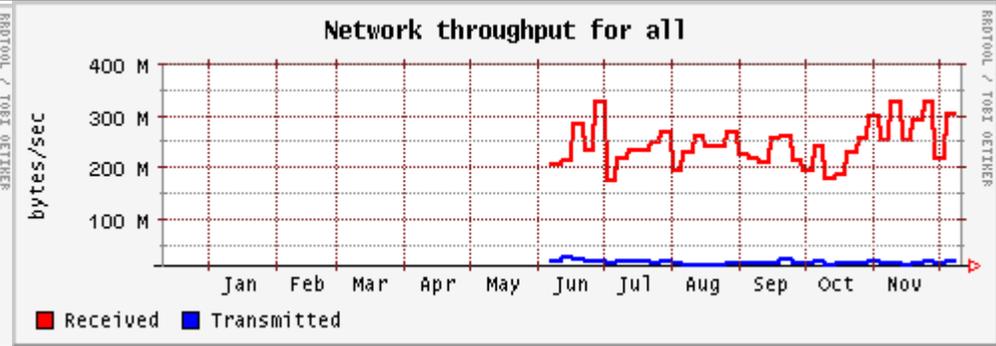
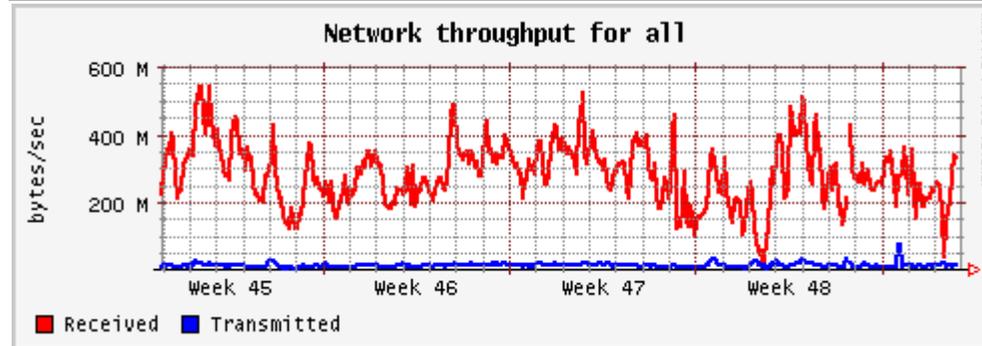
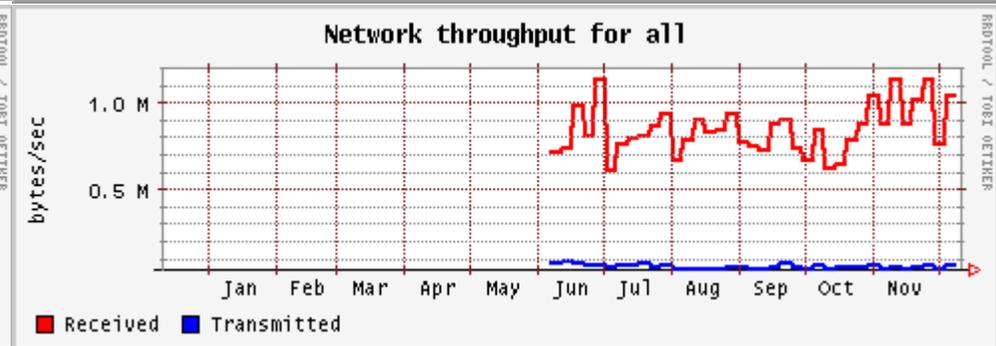
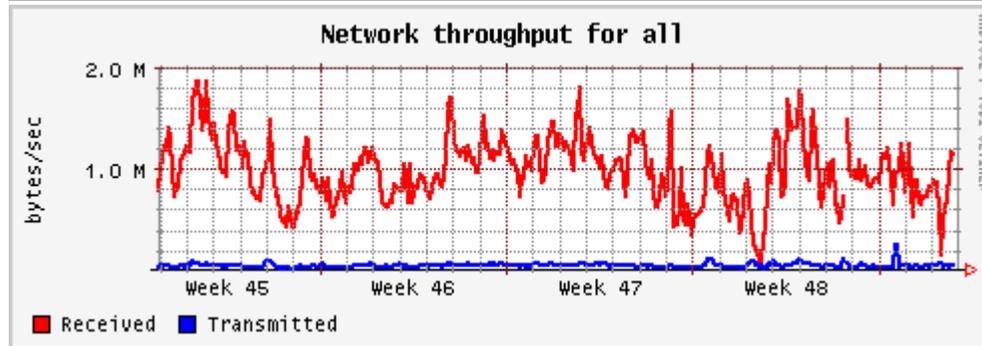
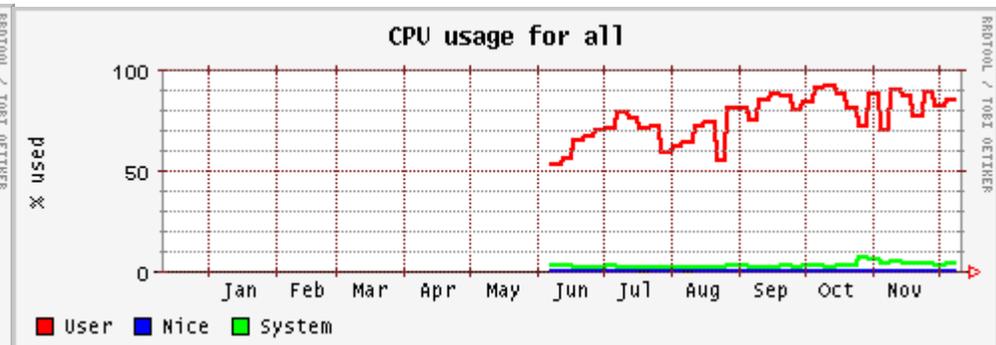
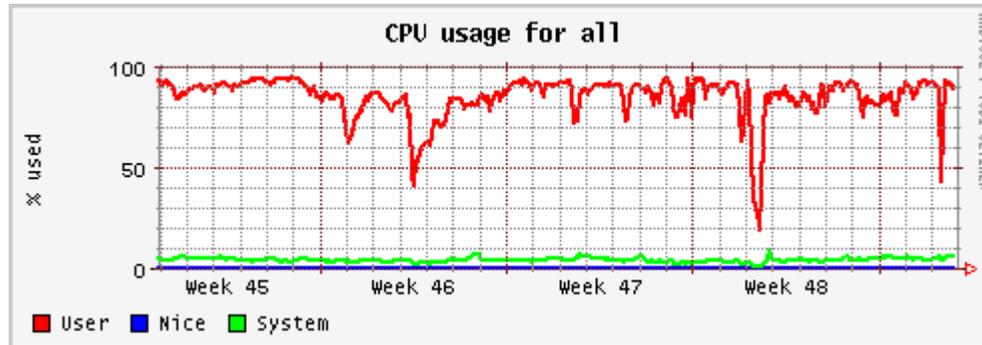
CAF Utilization



150 different users
running CAF jobs
last week!



CAF Utilization





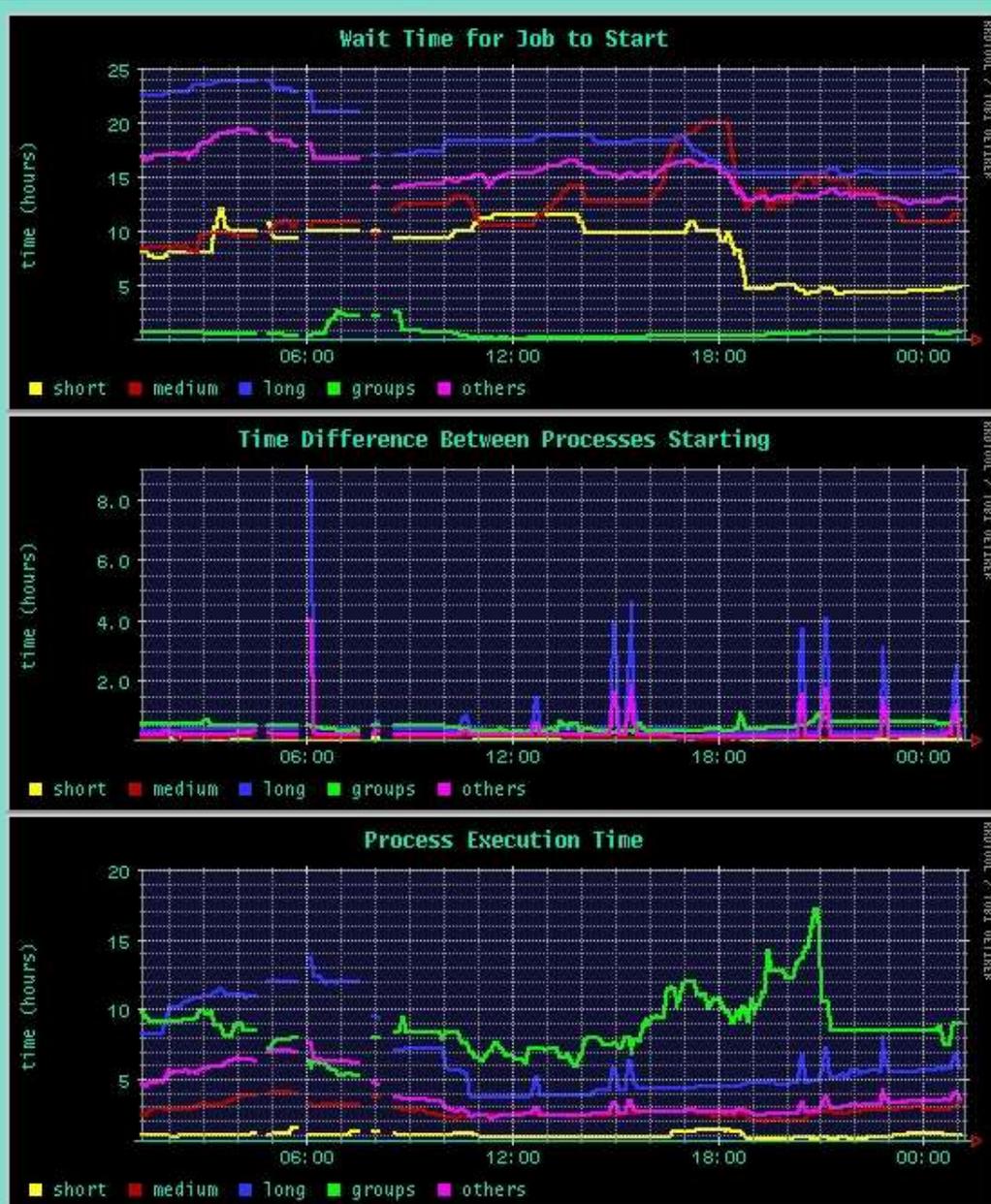
Recent CAF enhancements



- **Enhanced batch monitoring and reporting**



Batch status/trends



	Pending		Running		Wait time		Start time diff		Exec time	
	Current	Avg	Current	Avg	Current	Avg	Current	Avg	Current	Avg
Short	9529	9885	185	118	04:49	08:22	00:02	00:05	00:56	00:57
Medium	4216	3672	368	282	11:34	12:24	00:09	00:07	03:18	02:43
Long	1091	1104	212	270	15:22	18:27	00:15	00:32	05:43	06:33
Groups	1299	1099	111	146	00:43	00:37	00:41	00:29	09:03	08:43
Other	133	456	0	4	12:54	15:17	00:10	00:17	03:35	03:42
Total	16268	16218	877	823	Last updated: Thu Dec 11 01:05:00 2003					



Recent CAF enhancements



- **Enhanced batch monitoring and reporting**
- **Enhanced reporting to users**
 - job execution time, batch wait time
 - data handling information (this also is archived)

Subject: JID 829 summary

JID is 829.
Segment number from 1 to 5
Output location: icaf:temp_\$.tgz
Submitted: Sat Aug 23 19:10:33 2003
Ended : Sat Aug 23 20:34:53 2003
Job duration: 1:24:20

Job summary

started, ended,
duration

Segment times: Mean RMS
Real: 1:01:49 0:16:00
CPU: 0:43:05 0:01:24
Wait: 0:00:21 0:00:01

	#Segments
Completed with exit status = 0 (OK):	5
Completed with exit status != 0:	0
Canceled:	0
Timeout or possibly canceled while running:	0
User oriented error:	0
Caf system error:	0
Unknown error:	0
Total segments:	5

Segment summary

real/cpu time consumed,
wait time, errors

Segment	Node	Exit	Comment
1	fcdfcfa010	0	Real: 1:12:10 CPU: 0:44:10 Wait: 0:00:21
2	fcdfcfa009	0	Real: 1:12:26 CPU: 0:44:09 Wait: 0:00:21
3	fcdfcfa018	0	Real: 0:44:17 CPU: 0:41:29 Wait: 0:00:23
4	fcdfcfa019	0	Real: 0:44:25 CPU: 0:41:37 Wait: 0:00:22
5	fcdfcfa015	0	Real: 1:15:48 CPU: 0:44:04 Wait: 0:00:21

Data access summary

Datasets: aexp00,nbot0h

INPUT data summary:

	RecRead	EvtRead	RO(sec)	OC(sec)	Size(MB)	KbPerRec	KbPerEvt	FailOpen
Aggregate	7.8e+04	7.8e+04	26	18405	8.3e+03	--	--	0
Average	1.6e+04	1.6e+04	5.2	3681.0	1.7e+03	108.7	108.7	0

OUTPUT data summary:

	RecWrote	EvtWrote	OC(sec)	Size(MB)	KbPerRec	KbPerEvt
Aggregate	2.3e+05	2.3e+05	55308	2.5e+04	--	--
Average	4.7e+04	4.7e+04	11061.6	5.1e+03	111.4	111.5

INPUT specific data:

Segment	RecRead	EvtRead	RO(sec)	OC(sec)	Size(MB)	KbPerRec	KbPerEvt	FailOpen
1	15605	15598	6	4299	1.7e+03	108.7	108.7	0
2	15605	15598	6	4314	1.7e+03	108.7	108.7	0
3	15605	15598	4	2640	1.7e+03	108.7	108.7	0
4	15605	15598	4	2645	1.7e+03	108.7	108.7	0
5	15605	15598	6	4507	1.7e+03	108.7	108.7	0

I/O summary

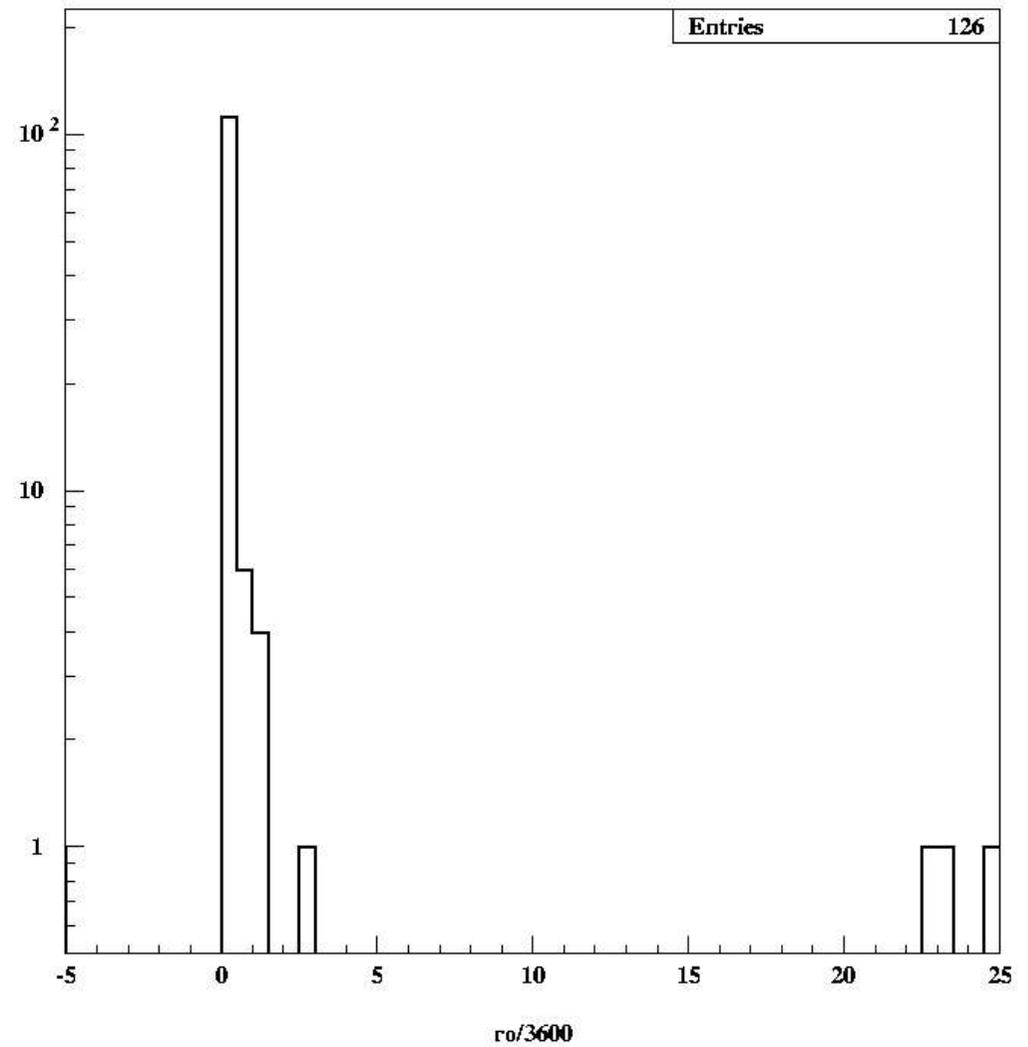
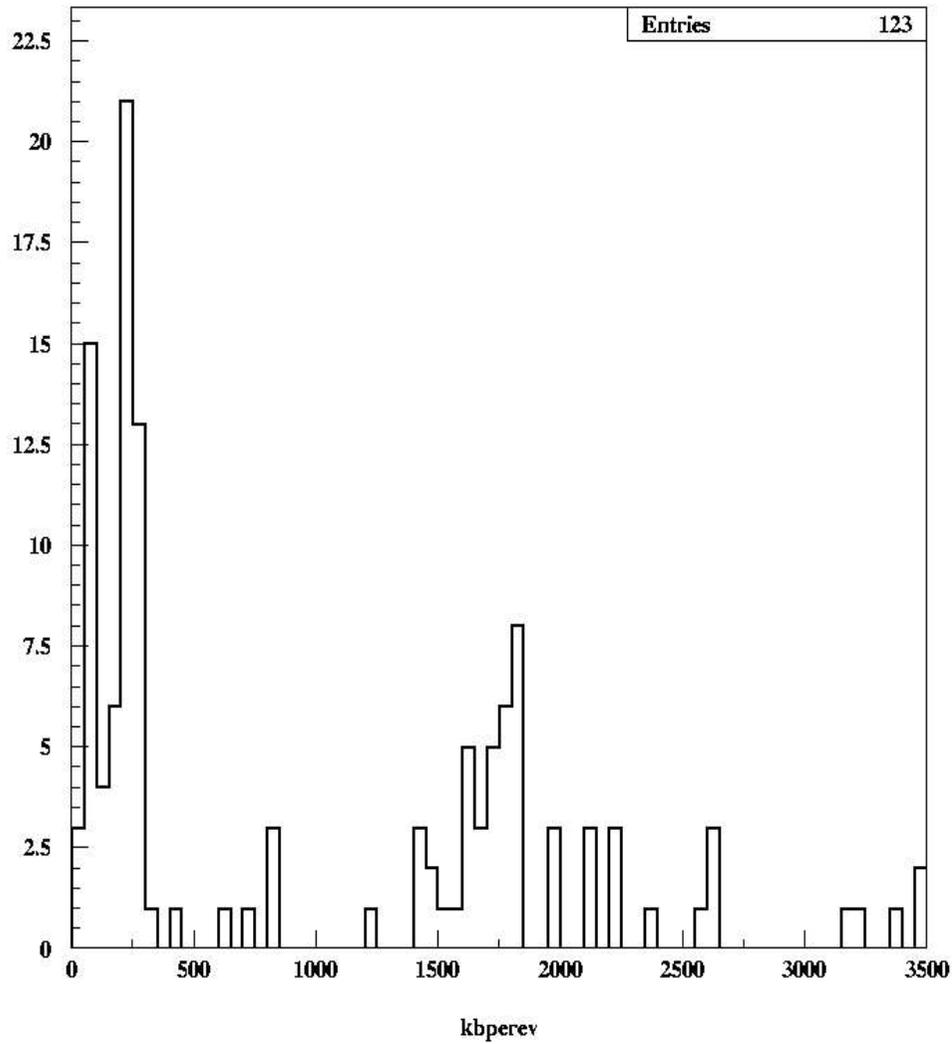
data volume I/O, kB/event,
wait times, errors

OUTPUT specific data:

Segment	RecWrote	EvtWrote	OC(sec)	Size(MB)	KbPerRec	KbPerEvt
1	46818	46794	12916	5.1e+03	111.4	111.5
2	46818	46794	12961	5.1e+03	111.4	111.5
3	46818	46794	7938	5.1e+03	111.4	111.5
4	46818	46794	7954	5.1e+03	111.4	111.5
5	46818	46794	13539	5.1e+03	111.5	111.5

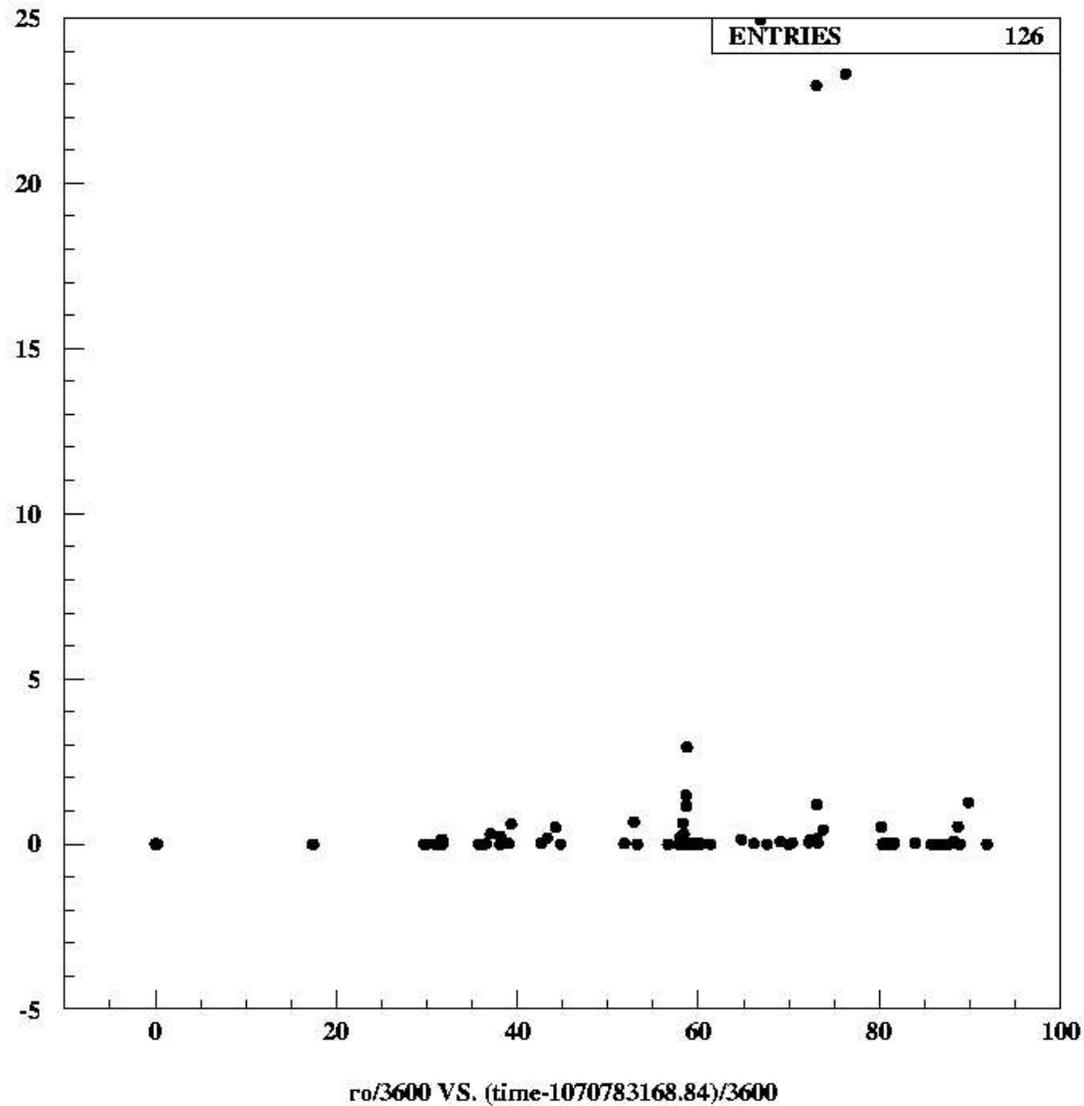


DH Logging for CAF jobs





DH Logging for CAF jobs





Recent CAF enhancements



- **Enhanced batch monitoring and reporting**
- **Enhanced reporting to users**
 - job execution time, batch wait time
 - data handling information (this also is archived)
- **SAM support on CAF**
 - declaration of SAM dataset to submission interface
 - generate project name and sam_start_project 1st section
 - checks for dataset integrity, total file size reporting, etc
 - sam_stop_project as last section of the job



SAM on CAF



Analysis Farm: caf fcdhead1.fnal.gov:8000

Specify SAM dataset? SAM Dataset ID: higgs

Process Type: short

Initial Command: /caf.sh 1 100

Original Directory: /home/msn/caftmp Browse...

Output File Location: icaftemp\$.tgz

Email? Email Address: msn@fnal.gov

Submit Quit Ready

(2003-12-11 01:31:58) Specifying of SAM dataset enabled

SAM dataset
declaration

- generate project name and sam_start_project 1st section
- checks for dataset integrity, total file size reporting
- other CAF sections are not run unless sam_start_project succeeds
- sam_stop_project as last section of the job



Recent CAF enhancements



- **Enhanced batch monitoring and reporting**
- **Enhanced reporting to users**
 - job execution time, batch wait time
 - data handling information (this also is archived)
- **SAM support on CAF**
 - declaration of SAM dataset to submission interface
 - generate project name and sam_start_project 1st section
 - checks for dataset integrity, total file size reporting, etc
 - sam_stop_project as last section of the job
- **Investigate feasibility of Condor-based CAF**



Condor-based CAF



Why?

- **Condor/Condor-G emerging as a frontrunner in job/resource management software for GRID applications**
- **Condor as a project is very active area for development and support**
- **Computing on Demand (COD) for PEAC (see next talk)**

Why now?

- **The success of our FBSNG-based CAF has given us the freedom to investigate other directions while physics continues to get done**



Condor-based CAF



Essentially all CAF functionality re-implemented

Still need:

- **Hierarchical fair share (2 layers ASAP for group accounting)**
- **Global resource quotas (have ugly workaround for short, med..)**
- **Ability to use VMx_USER with COD jobs**
- **Better monitoring**

Main problem is stability at 1/10th CAF-like scales

- **Communication timeouts between daemons under load causing jobs to fail at ~1300 jobs in Condor**
- **Numerous issues uncovered (particularly related to kerberos) that are being actively pursued by Condor team**

Bottom line - too early to tell



Future plans



Continue to operate and scale up the CAF

- **1132 CPUs, 2.4 THz now, ~1500 CPUs, 3.3 THz come spring**

Decision point for Condor-based CAF

Enhancements in collaboration with DH Group

- **SAM in open beta then production on CAF**
- **Virtualized scratch space**
- **Concatanation of CAF job output**
- **dCache write pools**

Better job of empowering users to predict job duration

CAF packaging/documentation for offsite installation