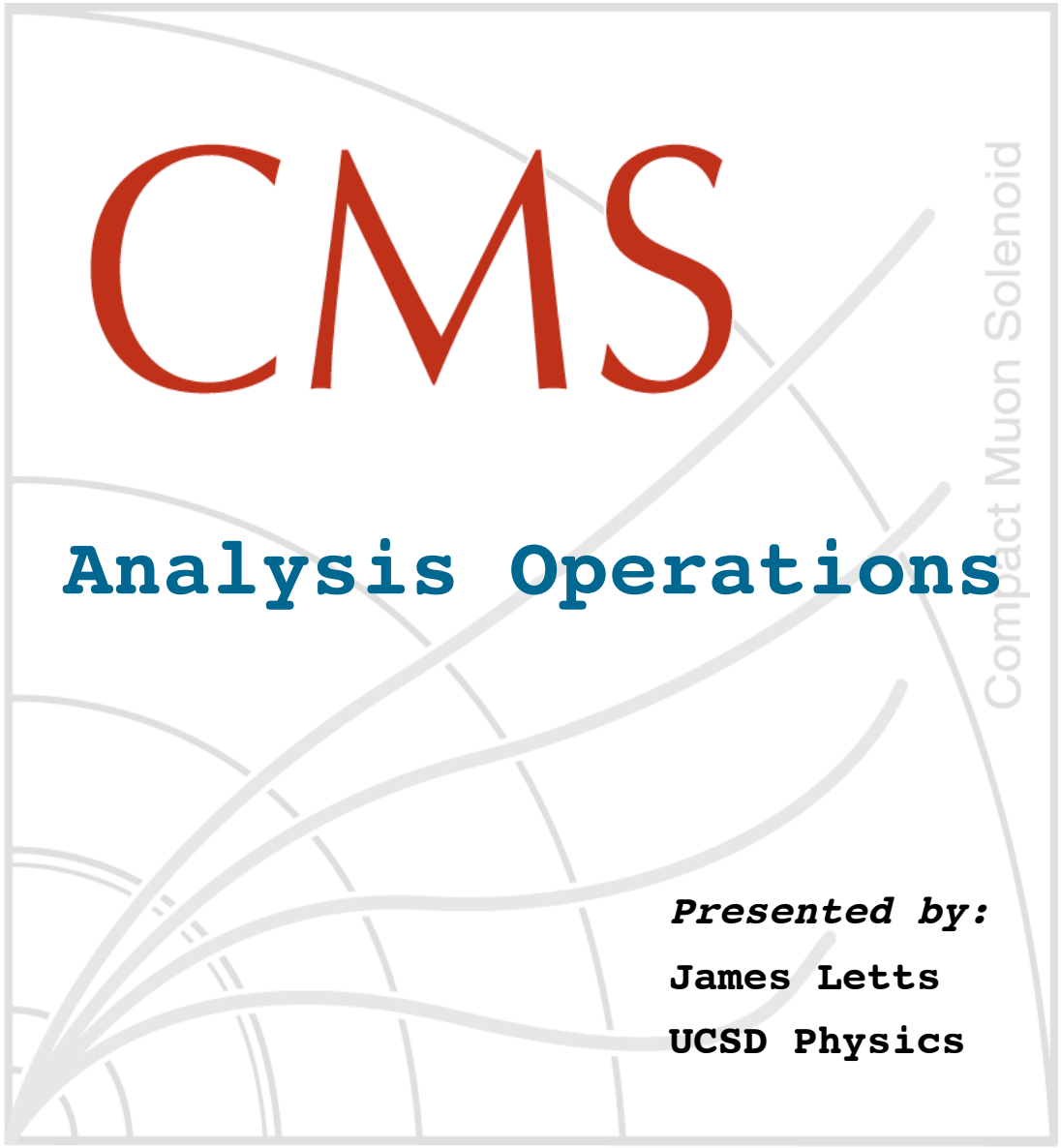


Analysis Operations

Presented by:

James Letts

UCSD Physics

The Challenge of Analysis Operations

- How to scale CMS analysis operations from today's level of usage:
  - A few hundred users
  - No intense competition for resources
- To the future:
  - Over a thousand active analysis users
  - Distributed at some 50+ sites world-wide
  - With increased competition for scarce resources
- Need to reconsider the scalability of current methods of:
  - User support for analysis operations
  - Monitoring tools
  - Debugging/testing procedures

Task Force

- In 2008 CMS created an Analysis Operations Task Force to meet these challenges by:
  - Evaluating the currently available job monitoring tools - are they sufficient to enable us to understand why jobs fail, failure rates etc.
  - Evaluating the current user support techniques
  - Developing tools and techniques to address issues at sites and with users
  - Getting direct feedback from users about their experiences with the distributed computing environment in a large experiment-wide USER SURVEY
  - Establishing "Reference Sites" with enhanced job monitoring and validated data samples for debugging user code problems
- This task Force reports to the CMS Integration Program (I. Fisk and J. Hernandez)

Current Analysis Support

Current model of analysis support is based on the following elements which already exist:

- **CRAB – the CMS analysis software, and its development community**
- **CMS Dashboard for Job Monitoring – job reporting at various stages of job submission, execution, completion, with archived information.**
- **Education and Documentation – tutorials and workbooks**
- **"Traditional" debugging support:**
  - **CRAB Feedback mailing list, an informed user community of experts and developers**
- **Data Placement with PhEDEx, the CMS data transfer platform, according to needs of physics groups, but also by supply and demand, managing of resources as competition for them increases**

Evolution of Analysis Support

Several additional technical and management elements (which we will review in the following slides) would be useful as Analysis Support evolves to a larger user community:

- **Debugging support:**
  - **Dedicated Analysis Ops Debugging Team**
    - **Pro-actively find and debug problems**
  - **Expert tracking of problems with ticketing system (savannah) or other documentation (twiki) – established but not heavily used for Analysis Operations yet**
  - **Reference Sites for debugging user code**
  - **Real time job monitoring – WM Core DashBoard Reporting and its extensions:**
    - **Admin access to user log files, configuration files etc.**

Evolution of Analysis Support

Additional technical and management elements, continued …

- **Metrics - to monitor and measure success (or failure)**
  - **Real time job monitoring, also useful for admins!**
  - **Other tools (Association Rule Mining)**
- **Integration and Development - someone to interface between Analysis Ops and CRAB Software developers, PhEDEx data transfers, Dashboard and other monitoring developers, etc.**

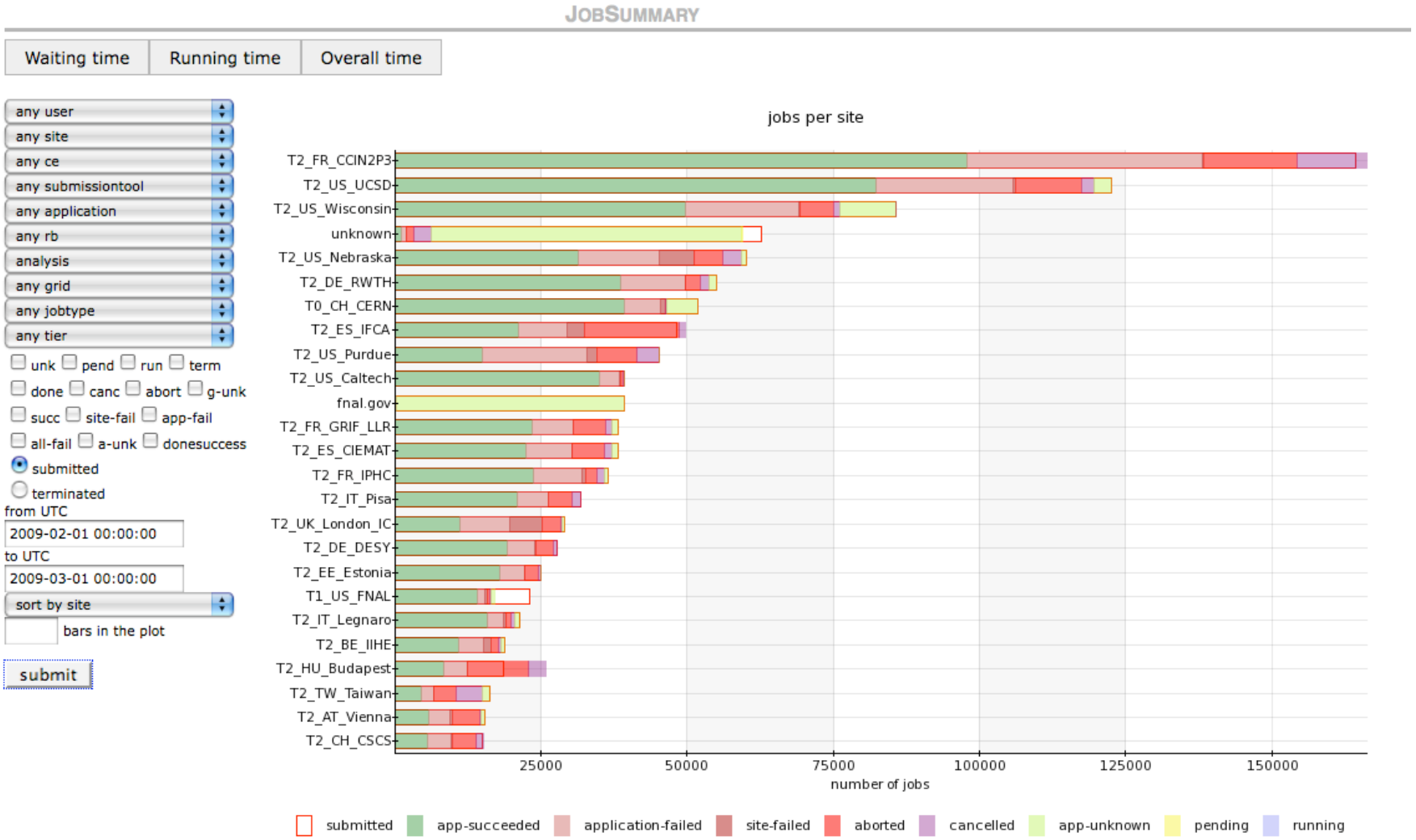
- **Will review most of these functionalities in the following slides.**

CMS Dashboard

- The CMS DashBoard is the current web interface to job monitoring - *dashboard.cern.ch*
- Job reporting at various stages of job submission, execution, completion, with exit codes for different stages.
- Provides an archive of the outcome of CRAB analysis and other activities, such as Monte Carlo production
- DashBoard does not provide access to user's log files - useful for debugging:
  - User (and admin!) feedback is that this functionality would be very useful
  - Detailed debugging information at the "Reference sites" - site level job information
  - Various ideas have been proposed to provide access to log files either at the end of jobs or even during the job execution (e.g. with CRAB Server)

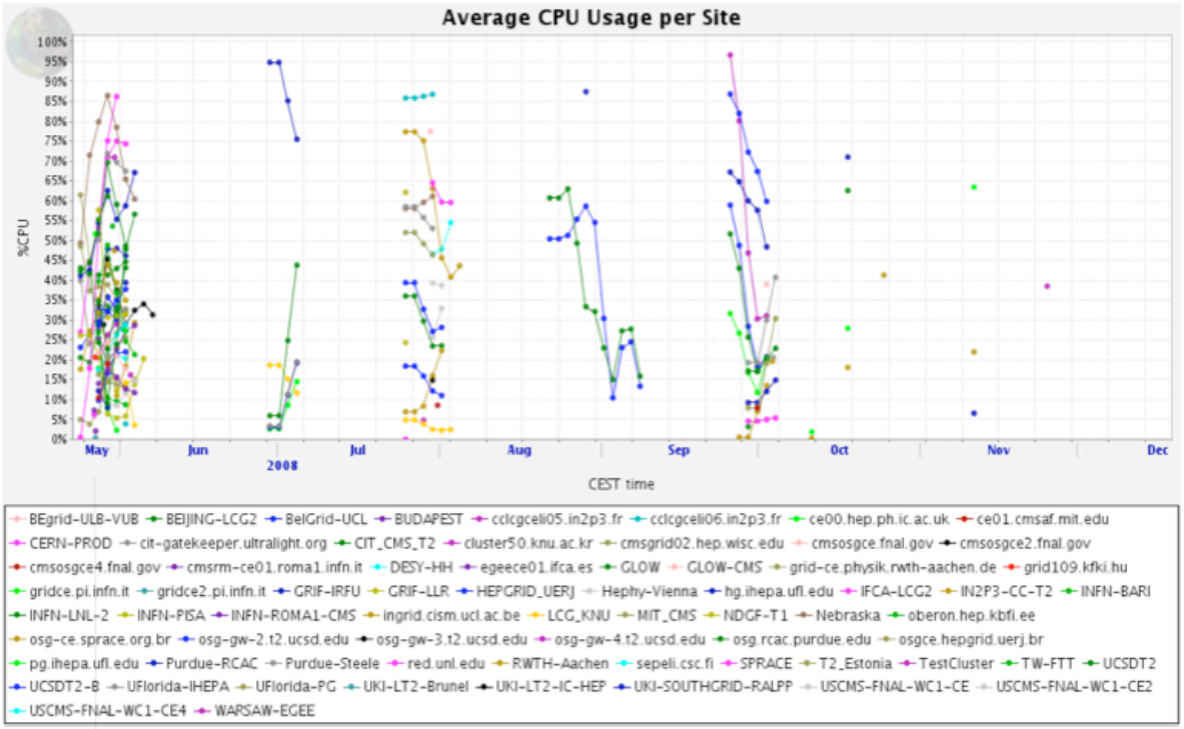
CMS Dashboard

- **Example: all analysis jobs from February 2009, by site:**



Real-time Monitoring

- WM Core DashBoard API was developed to provide a uniform reporting framework for all types of CMS jobs (analysis, production, etc.) to the DashBoard
- Not yet implemented in CRAB, soon?
- API can also be used to collect real time monitoring information from jobs and put it in a database, including log file information…



Example of job reporting during CCRC 08 (2008 LHC data challenge)

Other Tools

ASSOCIATION RULE DATA MINING

- **Data Mining based on combinations of attributes: dataset, user, site, CE, etc.**
- **Look for significantly high probability of a consequent ERROR from those "rules"**
- **Rules with high confidence may alert us to a problem at a site, or with a particular user's code, dataset, e.g.**

CMS dashboard — QAOES — SEND A COMMENT — GET MORE INFO

The page will be refreshed every hour.

Timerange:	last 6 hours	Description: jobs in the selected time range are the data set input for the data mining process.
Min number of jobs:	100	Description: minimum number of jobs, to list a created rule. Low number results in long calculation time!
Confidence:	80	Description: confidence% of jobs in the input data set that contain the attributes in the antecedent also contain the consequent attribute.

Show Rules

Link to jobSummary	Antecedent	Consequent	Support in % /number of jobs	Confidence in %	Lift
click!	dataset=**/QCDpt30/W...**	ERROR=60303	0.639/179	100.000	31.890
click!	ce=**cmsgrid02.hep.wisc.edu** queue=**cmsgrid02....** dataset=**/QCDpt300/...**	ERROR=60303	0.800/224	98.678	31.468
click!	user=**RachelAWilken** ce=**cmsgrid02.hep.wisc.edu** queue=**cmsgrid02....**	ERROR=60303	2.229/623	98.578	31.436
click!	dataset=**/QCDpt470/...**	ERROR=60303	0.729/203	97.608	31.127
click!	user=**RachelAWilken** site=**T2_US_Wisconsin**	ERROR=60303	2.693/754	96.667	30.827
click!	user=**RachelAWilken**	ERROR=60303	2.707/758	95.466	30.444
click!	dataset=**/QCDpt300/...**	ERROR=60303	1.207/337	94.944	30.277
click!	user=**RachelAWilken** ce=**cmsgrid01.hep.wisc.edu** queue=**cmsgrid01....**	ERROR=60303	0.464/130	88.435	28.202
click!	ce=**cmsgrid01.hep.wisc.edu** queue=**cmsgrid01....** dataset=**/QCDpt300/...**	ERROR=60303	0.407/113	88.372	28.181
click!	user=**Asciaba** site=**T2_RU_SINP** dataset=**/QCD_pt_0_...**	ERROR=8001	0.839/235	81.597	10.711
click!	site=**T2_RU_SINP**	ERROR=8001	0.839/235	81.034	10.637

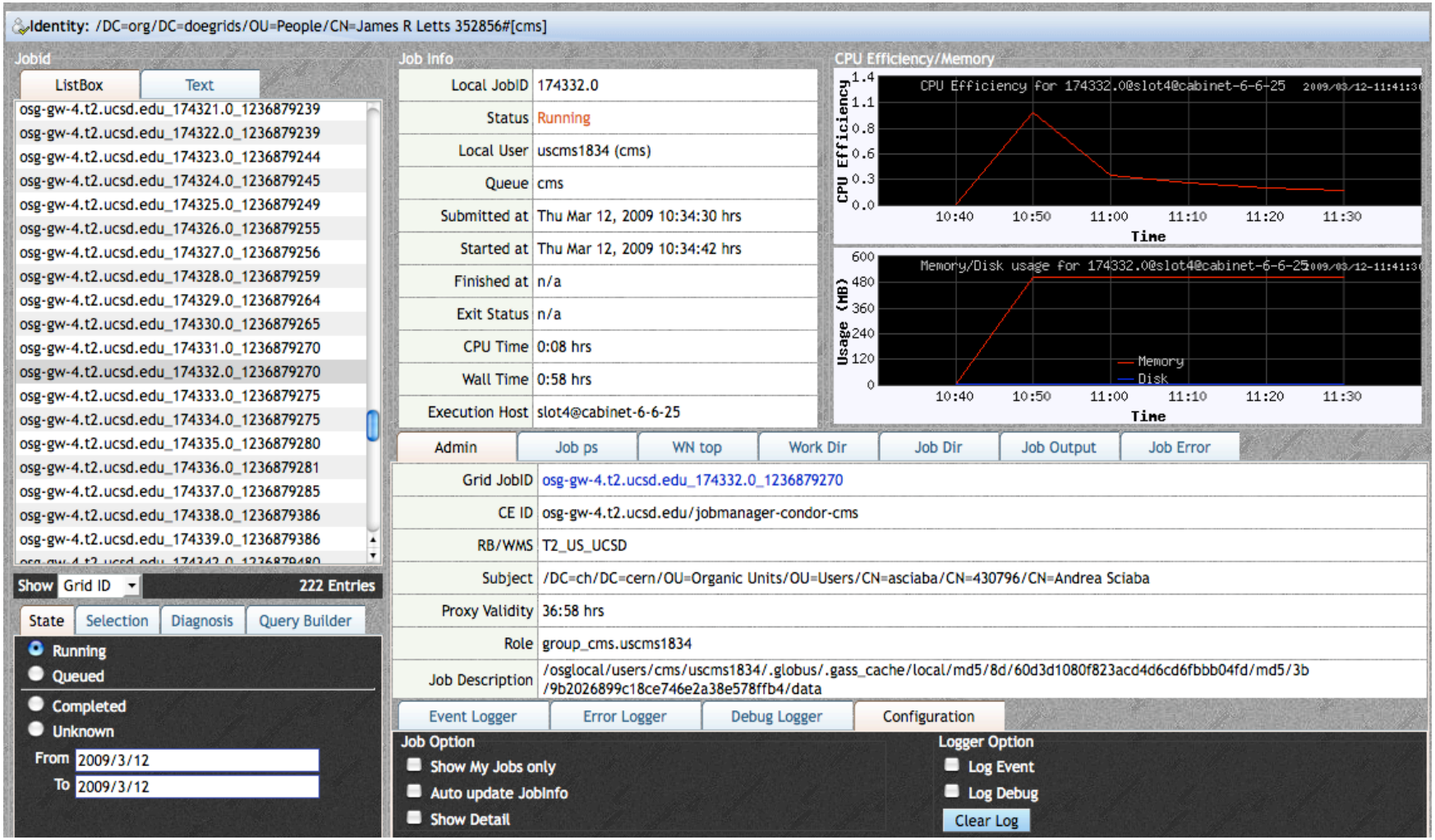
Reference Sites

- A Reference Site is a site with enhanced job monitoring capabilities to allow detailed debugging of analysis jobs even as they are running (which is generally not possible)
- Small standardized data samples, validated for each version of the analysis software
- Fast queue for quick job turnaround
- Monitoring tool information available via a web interface

- Subir Sarkar's (INFN-Pisa) Job Monitoring software is being deployed at several sites (Pisa, San Diego, Fermilab, Belgium IIHE):
  - Real-time monitoring
  - Interfaces to LSF, PBS, CONDOR
  - Access based on Grid Certificate, for privacy
  - Admins can have broader access.

Reference Platforms

- **Example of the job monitor, with a CONDOR batch system**



Metrics

- Establish metrics to measure success or failure, alert to problems; breakdown errors into categories
- Data Access errors:
  - On Feb. 12th, the task force started Data Consistency Check campaign
  - Verify that physical disk and databases (TMDB, DBS) are consistent
  - Reduced data access errors to 1.4% that week
  - Regular consistency checks are needed to sustain low error rates.
- StageOut of user data - need timeouts, alternative solutions as in CMS Monte Carlo production system

Week	Begin	End	Njobs Submitted	Njobs Terminated	All Fail	Njobs rc=0 Submitted	Job Rate of Success	StageOut rc Failures (%)	Data Access rc Failures (%)
2	05-Jan-09	12-Jan-09	206434	206142	42323	110277	53%	17.1%	1.5%
3	12-Jan-09	19-Jan-09	185361	185325	30686	98715	53%	9.8%	1.9%
4	19-Jan-09	26-Jan-09	225622	225590	58591	125667	56%	8.1%	1.3%
5	26-Jan-09	02-Feb-09	243593	242583	49520	123148	51%	7.4%	2.4%
6	02-Feb-09	09-Feb-09	258319	258309	57569	145316	56%	9.0%	2.2%
7	09-Feb-09	16-Feb-09	323411	323006	42532	161346	50%	15.8%	1.4%
8	16-Feb-09	23-Feb-09	360766	359930	55717	201412	56%	10.1%	2.3%
9	23-Feb-09	02-Mar-09	383254	372725	106938	196655	53%	13.8%	3.0%
10	02-Mar-09	09-Mar-09	442165	423373	50543	251295	59%	32.5%	4.1%
TOTAL			2628925	2596983	494419	1413831	54%	13.7%	2.2%

Metrics

- CPU, WallClock (WC) time success rates (79%) show resources that are actually used successfully.
- Task Force investigated how situation can be improved further by identifying and fixing problems to increase the success rates.
- Real-time monitoring would be a useful enhancement
  - No need to archive beyond a few days

Week	CPU Submitted	CPU rc=0 Submitted	CPU Rate of Success	WC Submitted	WC rc=0 Submitted	WC Rate of Success	CPU/WC Efficiency	CPU/WC rc=0 Efficiency
2	258427144	188279622	73%	738442642	480104317	65%	35%	39%
3	427255384	305773699	72%	3227320573	3030718600	94%	13%	10%
4	432423802	367532671	85%	1155467203	748561095	65%	37%	49%
5	506978830	423558777	84%	992185407	813237123	82%	51%	52%
6	743412864	653799576	88%	1304721403	1045722974	80%	57%	63%
7	808942137	642462156	79%	1406548572	1046536702	74%	58%	61%
8	758806604	627778170	83%	1353954630	1090041442	81%	56%	58%
9	1045707756	796422157	76%	1805110388	1304998959	72%	58%	61%
10	1286579309	926280531	72%	2409760513	1754353272	73%	53%	53%
TOTAL	6268533830	4931887359	79%	14393511331	11314274484	79%	44%	44%

Examples of Analysis Operations

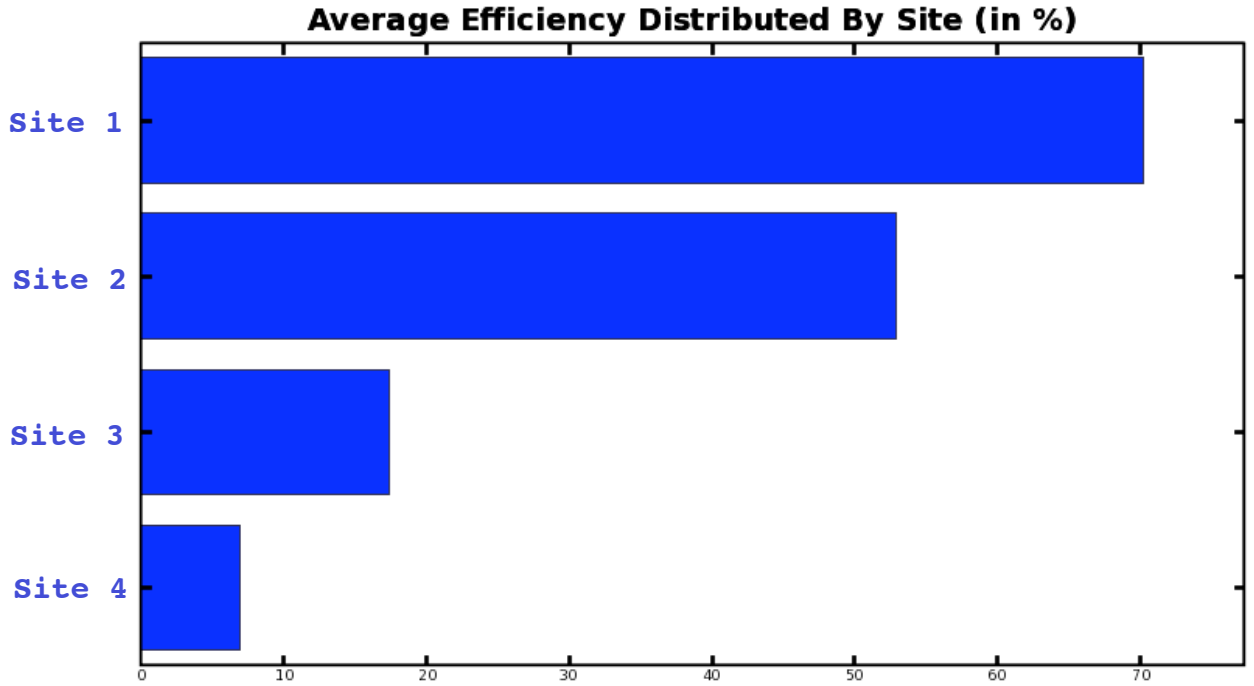
Examples of investigations, researching problems, and identifying areas for improvement:

1. Looking at a single analysis task (same user, same code, same dataset) running at different sites to compare performance (metrics)

2. Identifying sources of wasted resources at a particular site in detail

Same Task at Different Sites

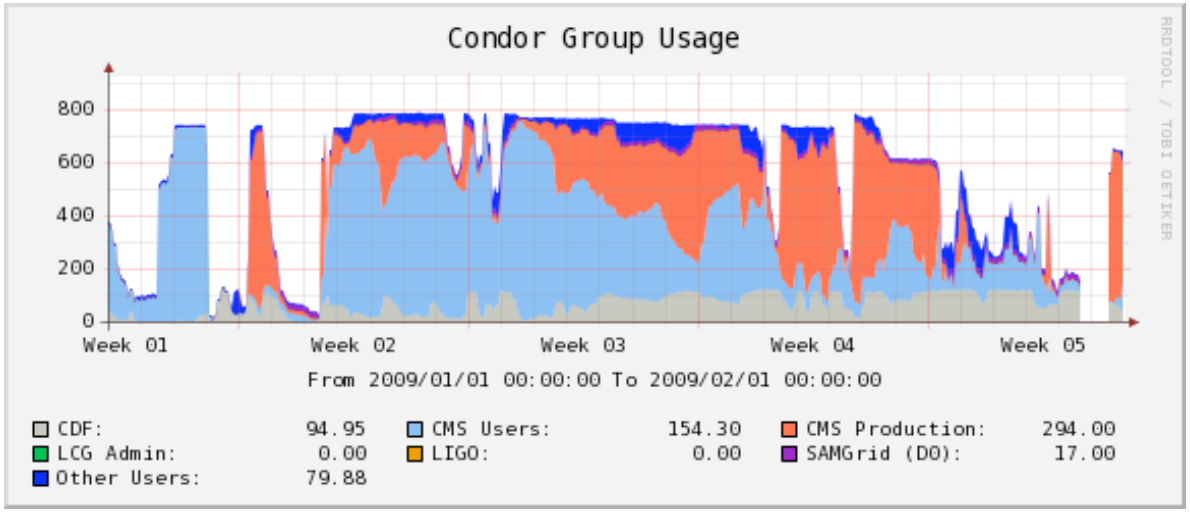
- CPU/WC efficiency of different parts of the same task (same user, code, dataset) run at different sites
- Can identify data access inefficiencies or problems.



- Task Force worked with Dashboard developers to be able to dump DashBoard DB and make different plots quickly, can be useful for Analysis Ops team to develop metrics

Compare Dashboard and Local Batch Accounting

- A detailed comparison of DashBoard CPU/WC usage accounting compared to local batch system accounting was done at one site.



- Found to be consistent, with only about ~10% of consumed analysis resources not being reported to the CMS DashBoard
  - Presumably this is user code outside the framework of CRAB (ntuple analysis, private code)

Looking for Wasted Resources

- An understanding of jobs which terminate with status "Unknown" in the Dashboard came from this exercise:
  - At this site, jobs which run longer than 36h are presumed to be stuck/hung/looping and are killed
  - 2% of jobs were running to this 36h WC time limit
  - In contrast: typical analysis jobs run ~40m
  - Therefore, 2%*36h/(2%*36h+98%*40m)=52% of resources were being wasted by 2% of the jobs in that period
- An analysis of the jobs revealed that:
  - About 1/3 were stuck in the final StageOut of the user files (not consuming CPU, just holding a slot)
  - ~1/3 were just very long jobs (consuming CPU)
  - ~1/3 were jobs which were caught in a loop, stuck reading a particular data file, or otherwise problematic
- These kinds of CPU/WC time limitations at sites are common! (24h, 36h, 48h, etc.)

Areas for Improvement

From these kinds of studies, the Task Force has identified two main areas to consider improvements:

1. **StageOut of user files**
   - **Process hangs, or mis-reports error code**
   - **Need a timeout, alternative solutions as in the CMS Monte Carlo production system**

2. **Matching between CPU/WC time limitations at sites and the CPU resources needs of CRAB jobs**
   - **Many sites are not publishing limitations, and they vary greatly by site**
     - GluePolicyMaxWallClockTime
   - **Some sites have no effective limitations and could conceivably waste all of their resources if not monitored**
   - **The CRAB application doesn't yet match resource requirements to site limits.**

User Analysis Survey

- The Task Force was also asked to conduct an experiment-wide analysis USER SURVEY
- Questions on experience, expectations in using CRAB, debugging problems, tutorials, etc.
- There were ~800 analysis users in 2008-2009 (users who submitted at least one job)
- 226 People filled out the survey conducted in February and March 2009.
  - Average level of experience doing analysis in CMS was between 1-3 years
  - Almost 90% of analysis users are using the CRAB software (consistent with ~10% private code, as we saw earlier)

Survey: Current Practices

- About 50% of users reported submitting via the CRAB Server (which takes care of the Grid submission for the user), and 50% via the CRAB client directly to the Grid.
  - Of those who are not using CRAB Server, the vast majority said they have not tried it yet or are not familiar with it.

- Retrieval of user results:
  - 44% StageOut to a Tier 2 or Tier 3 center
    - Tier 2 centers are providing storage space for user analysis results (/store/user areas). Goal to move users to this method.
  - 28% StageOut to CASTOR at CERN
  - 28% return files back to the submission machine (e.g. desktop)

Survey: Current Practices

- When jobs fail, ~90% of users look to the standard output and standard error log files
  - Not many reported using the various monitoring tools available (e.g. DashBoard) mostly because they are unfamiliar with them.
  - About half of users either browse or ask for help in the CRABFeedback mailing list when things go wrong.
    - Of these, >80% reported getting helpful feedback quickly from the mailing list.
  - About half of users also simply resubmit the jobs and investigate if they fail twice.

- Documentation and Education: 85% of users have followed or read the CRAB tutorials

Survey: Future Functionality

- 73% surveyed would use a Reference Site to debug a job.
- 78% would use a tool that gave a snapshot of a running job - Real-time Job Monitoring

- With newer features of the analysis infrastructure (CRAB Server, DashBoard developments), most users that are not using them because of unfamiliarity with them.
  - Underscores the importance of education and documentation as new functionalities and management are introduced into Analysis Operations.

Transition to Analysis Operations

Most of the technical requirements to do effective Analysis Operations exist today or in the development pipeline:

- **Existing:**
  - **DashBoard job monitoring and archiving**
  - **An active community of experts for feedback**
  - **Education and documentation for CRAB, CRAB Server etc., tutorials and workbooks**

- **In the pipeline:**
  - **Reference Sites for fast, detailed job debugging**
  - **WM Core DashBoard Reporting for real-time job monitoring:**
    - **Job Monitor: Real-time access to user log files for debugging purposes - needed to make Analysis Ops efficient.**

Transition to Analysis Operations

Analysis Operations team can do more active management, improve efficiency, and interface between the various core activities - what an Analysis Operations team might look like:

- **Integration and Development - someone to interface between Analysis Ops and the various stakeholders in Analysis activities:**
  - **Physics Analysis Groups**
  - **CRAB and CRAB Server software developers**
  - **PhEDEx (for data placement, consistency checks)**
  - **Sites where analysis takes place**
  - **Dashboard and other monitoring developers**
  - **Education and documentation efforts**
  - **Provide feedback to the developers and CMS computing management on performance of the analysis infrastructure**

Transition to Analysis Operations

The Analysis Operations Team (continued…)

- Dedicated people for pro-active job debugging and user support; this is in addition to and in coordination with the traditional mailing list based support community:
  - Tracking and documentation of problems (savannah, twiki)
  - 2-4 people in various time zones? Depends on need.

- Metrics/Monitoring development person:
  - Developing and reporting on various metrics to understand the system as a whole and to identify problem areas
  - Coordinate with developers of monitoring software

- Reference site commissioning
- Data Set Consistency

Acknowledgements

Many people participated in the activities of the Analysis Support Task Force: workshop at CERN in December, weekly meetings, related technical activities:

- **CMS Dashboard: J. Andreeva, G. Maier (IT)**
- **CRAB Developers: D. Spiga, F. Fanzango, et al.**
- **Job Monitoring: S. Sarkar, S. Padhi, H. Pi**
- **WM Core DashBoard API: S. Padhi**
- **Members of the Task Force: M. Calloni, D. Colling, J. Klem, J. Maes, P. Van Mulders, S. Padhi, I. Villella,**
- **Task Force Coordinators: Prof. J. D'Hondt and myself**