Available on CMS information server

CMS CR -2009/088

The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

07 May 2009

CMS Analysis Operations

J. Andreeva, M. Calloni, D. Colling, F. Fanzago, J. D'Hondt, J. Klem, G. Maier, J. Letts, J. Maes, S. Padhi, S. Sarkar, D. Spiga, P. Van Mulders, I. Villella

Abstract

During normal data taking CMS expects to support potentially as many as 2000 analysis users. Since the beginning of 2008 there have been more than 800 individuals who submitted a remote analysis job to the CMS computing infrastructure. The bulk of these users will be supported at the over 40 CMS Tier-2 centres. Supporting a globally distributed community of users on a globally distributed set of computing clusters is a task that requires reconsidering the normal methods of user support for Analysis Operations. In 2008 CMS formed an Analysis Support Task Force in preparation for large-scale physics analysis activities. The charge of the task force was to evaluate the available support tools, the user support techniques, and the direct feedback of users with the goal of improving the success rate and user experience when utilizing the distributed computing environment. The task force determined the tools needed to assess and reduce the number of non-zero exit code applications submitted through the grid interfaces and worked with the CMS experiment dashboard developers to obtain the necessary information to quickly and proactively identify issues with user jobs and data sets hosted at various sites. Results of the analysis group surveys were compiled. Reference platforms for testing and debugging problems were established in various geographic regions. The task force also assessed the resources needed to make the transition to a permanent Analysis Operations task. In this presentation the results of the task force will be discussed as well as the CMS Analysis Operations plans for the start of data taking.

Presented at *CHEP 2009 (17th International Conference on Computing in High Energy and Nuclear Physics),21-27 March 2009,Prague,Czech Republic,15/05/2009*

CMS Analysis Operations

J. Andreeva[1], M. Calloni[2], D. Colling[3], F. Fanzago[4], J. D'Hondt[5], J. Klem[6], G. Maier[1], J. Letts[7], J. Maes[5], S. Padhi[7], S. Sarkar[8], D. Spiga[1], P. Van Mulders[5], I. Villella[5]

1. CERN, Geneva, Switzerland 2. Università degli Studi Milano-Bicocca, Italy 3. Imperial College, London, UK 4. Università degli Studi di Padova and INFN Sezione di Padova, Italy 5. Vrije Universiteit Brussel, Brussels, Belgium 6. Helsinki Institute of Physics, Helsinki, Finland 7. University of California San Diego, La Jolla, California, USA, 8. Università degli Studi di Pisa and INFN Sezione di Pisa, Italy

Email: `jletts@ucsd.edu`

Abstract. During normal data taking CMS expects to support potentially as many as 2000 analysis users. Since the beginning of 2008 there have been more than 800 individuals who submitted a remote analysis job to the CMS computing infrastructure. The bulk of these users will be supported at the over 40 CMS Tier-2 centres. Supporting a globally distributed community of users on a globally distributed set of computing clusters is a task that requires reconsidering the normal methods of user support for Analysis Operations. In 2008 CMS formed an Analysis Support Task Force in preparation for large-scale physics analysis activities. The charge of the task force was to evaluate the available support tools, the user support techniques, and the direct feedback of users with the goal of improving the success rate and user experience when utilizing the distributed computing environment. The task force determined the tools needed to assess and reduce the number of non-zero exit code applications submitted through the grid interfaces and worked with the CMS experiment dashboard developers to obtain the necessary information to quickly and proactively identify issues with user jobs and data sets hosted at various sites. Results of the analysis group surveys were compiled. Reference platforms for testing and debugging problems were established in various geographic regions. The task force also assessed the resources needed to make the transition to a permanent Analysis Operations task. In this presentation the results of the task force will be discussed as well as the CMS Analysis Operations plans for the start of data taking.

1. Introduction

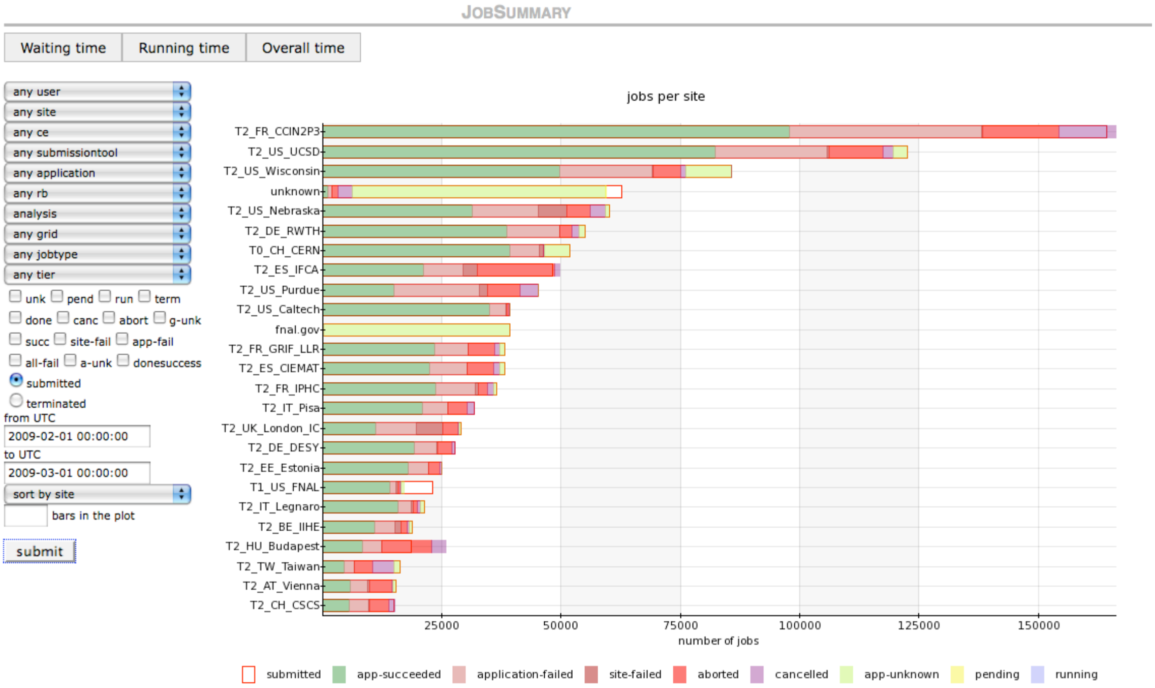
One of the major challenges facing the CMS experiment in the run-up to data taking is how to scale data analysis support to the needs of the future. In 2008 and the beginning of 2009 there were approximately 800 analysis users of varying levels of expertise, of which about 200 could be considered active analysis users. Later in 2009 and beyond, we expect that this number will grow by up to an order of magnitude, with users submitting to 50+ sites worldwide. Are the current methods and techniques of analysis support adequate to support this growing user community? Are the monitoring and software tools available to users and system administrators up to the task? Or do we need to reassess the way we do analysis support for a much larger community?

In the autumn of 2008 CMS established an Analysis Support Task Force to study these questions and to perform an experiment-wide survey of analysis users to understand the way in which people interact with the support infrastructure. The charge of the task force was to evaluate the available support tools, the user support techniques, and obtain the direct feedback of users with the goal of improving the success rate of jobs and evaluating the user experience when utilizing the distributed computing environment. The task force determined the tools needed to assess and reduce the number of non-zero exit code applications submitted to through the grid interfaces and worked with the CMS Dashboard [1] developers to obtain the necessary information to quickly and proactively identify issues with user jobs and data sets hosted at various sites. Results of the analysis group surveys were compiled. Plans for establishing “Reference Sites”, sites with enhanced real-time job monitoring for testing and debugging problems, were made in various geographic regions. The task force also assessed the resources needed to make the transition to a permanent Analysis Operations task.

2. Current Model of Analysis Support

The current model of analysis support is based on the following elements that are already in place:

- CRAB [2] - the CMS analysis software and its development community
- The CMS Dashboard - for job monitoring (job reporting at various stages of the job submission, execution and completion, with an archive)
- Education and Documentation - tutorials and workbooks.
- “Traditional” Debugging Support - via a mailing list with an informed user community of experts and developers.
- Data Placement – with PhEDEx [3,4], the CMS data transfer platform, according to the needs of physics groups, but also supply and demand.



JOBSUMMARY

any user
any site
any ce
any submissiontool
any application
any rb
analysis
any grid
any jobtype
any tier

□ unk □ pend □ run □ term
□ done □ canc □ abort □ g-unk
□ succ □ site-fail □ app-fail
□ all-fail □ a-unk □ donesuccess
● submitted
○ terminated

from UTC 2009-02-01 00:00:00

to UTC 2009-03-01 00:00:00

sort by site

bars in the plot

Fig. 1. The CMS Dashboard user interface, shown here for all analysis jobs submitted in February 2009, broken down by site where the job ran.

An example of the user interface for the CMS Dashboard is shown in Fig. 1. From this interface user and administrators can track the exit codes of analysis and other CMS jobs, broken down by site, user, software version, etc. over any time period. The Dashboard does not provide access to the user's log files, which is useful for debugging problems with a job. The feedback that the task force got from administrators, developers and users alike is that this would be a useful functionality to have, although not necessarily as part of the Dashboard monitoring. Various ideas have been proposed to provide access to standard output and error logs of jobs either at the end of jobs via the CRAB Server [4] or even during job execution, or by providing more detailed information to job monitoring software.

3. Evolution of Analysis Support

Several additional technical and management elements which we will review in the following sections of this note would be useful as Analysis Operations evolves to support a larger user community:

- Dedicated Analysis Operations debugging team to pro-actively find and debug problems
- Expert tracking of problems with a ticketing system (such as LCG savannah [5]) or other documentation such as the CMS twiki. This functionality is established but not extensively used for Analysis Operations yet.
- Reference sites for debugging user code
- Real-time job monitoring with real-time access to job log files
- Metrics to monitor and measure success
- Integration and Development – someone to interface between Analysis Operations and the various stakeholders in CMS analysis, such as the CRAB, PhEDEx and Dashboard software development communities, for example.

4. Metrics

4.1. Metrics Based on Current Dashboard Information

Throughout the task force working period, we compiled weekly data for "metrics" based on monitoring information from the CMS Dashboard. Metrics are useful to get a global picture of the success or shortcomings of Analysis Operations. Operational problems can be identified quickly by properly targeted metrics. The evolution of metrics over time can also be an indicator of whether Analysis Operations are successful at improving the success rates of jobs.

Metrics that were considered during the task force were based on the numbers of job submitted and completed in any given week, with a breakdown of the success rate (defined as a job with an exit code of zero) both by the number of jobs and weighted by CPU and wall clock (WC) time. While the metrics based on the number of jobs show overall success, the CPU and WC time metrics show the successful usage of computing resources. Details of these metrics are presented in Tables 1 and 2.

For example, during the first days of week 7, the task force members noticed that one of the metrics related to the percentage of jobs failing with data access failures (in Table 1, right column) was increasing rapidly. On February 12th the task force started a data consistency campaign at all Tier-2 sites[1]. This check verified that the contents of the disks at the Tier-2 sites were consistent with the PhEDEx [3,4] databases and DBS [7], the main file database that analysis jobs use. By the end of the week data access errors had been reduced to 1.4% of the total. However, from the subsequent

[1] The CMS computing model [6] has a tiered structure, with CERN being the Tier-0, eight regional Tier-1 sites including one at CERN, and >40 Tier-2 sites world-wide where most data analysis takes place.

evolution of the metric for data access failures, it is clear that regular checks are needed to sustain low error rates.

The failure of the staging out of user data at the end of CRAB jobs is a major source of error, as seen in Table 1. This problem is compounded by the fact that many jobs hang in the stage out step, blocking resources for other jobs while consuming no CPU time and thus reducing overall efficiency. Timeouts are needed for the stage out step, as well as alternate solutions when staging out to the primary destination storage element fails.

Week	Begin	End	Njobs Submitted	Njobs Terminated	All Fail	Njobs rc=0 Submitted	Succes Rate (%)	StageOut Failures (%)	Data Access Failures (%)
2	05-Jan-2009	12-Jan-2009	206434	206142	42323	11027	53%	17.1%	1.5%
3	12-Jan-2009	19-Jan-2009	185361	185325	30686	98715	53%	9.8%	1.9%
4	19-Jan-2009	26-Jan-2009	225622	225590	58591	125667	56%	8.1%	1.3%
5	26-Jan-2009	02-Feb-2009	243593	242583	49520	123148	51%	7.4%	2.4%
6	02-Feb-2009	09-Feb-2009	258319	258309	57569	145316	56%	9.0%	2.2%
7	09-Feb-2009	16-Feb-2009	323411	323006	42532	161346	50%	15.8%	1.4%
8	16-Feb-2009	23-Feb-2009	360766	359930	55717	201412	56%	10.1%	2.3%
9	23-Feb-2009	02-Mar-2009	383254	372725	106938	196655	53%	13.8%	3.0%
10	02-Mar-2009	09-Mar-2009	442165	423373	50543	251295	59%	32.5%	4.1%
TOTAL			2628925	2596983	494419	1413831	54%	13.7%	2.2%

Table 1. Metrics based on the number of jobs submitted during a given calendar week, with success rates. An analysis of failure rates due to stage out errors and data access errors are also given.

Week	CPU Submitted (ksec)	CPU rc=0 Submitted (ksec)	CPU Success Rate (%)	WC Submitted (ksec)	WC rc=0 Submitted (ksec)	WC Success Rate (%)	CPU/WC Efficiency (%)	CPU/WC Efficiency rc=0 (%)
2	258	188	73%	738	480	65%	35%	39%
3	427	306	72%	3227	3031	94%	13%	10%
4	432	368	85%	1155	749	65%	37%	49%
5	507	424	84%	992	813	82%	51%	52%
6	743	654	88%	1305	1046	80%	57%	63%
7	809	642	79%	1407	1047	74%	58%	61%
8	759	628	83%	1354	1090	81%	56%	58%
9	1046	796	76%	1805	1305	72%	58%	61%
10	1287	926	72%	2410	1754	73%	53%	53%
TOTAL	6269	4932	79%	14394	11314	79%	44%	44%

Table 2. Metrics based on the CPU and WC time weighted usage of jobs, with success rates and efficiencies of CPU/WC ratios.

CPU and wall clock success rates are about 80%, showing how effectively resources are actually being used. In Section 6 we discuss an investigation of the sources of inefficiency at one particular site.

Another useful example of a metric is looking at a single analysis task (same user, same code, same data set) that was run at different Tier-2 sites, as shown in Fig. 2. From this metric we can identify possible issues with data access speeds or stage out problems at particular sites. The task force worked with the Dashboard developers to be able to dump the Dashboard database and make different plots quickly. These techniques can be useful for a future Analysis Operations team to develop and view metrics quickly.

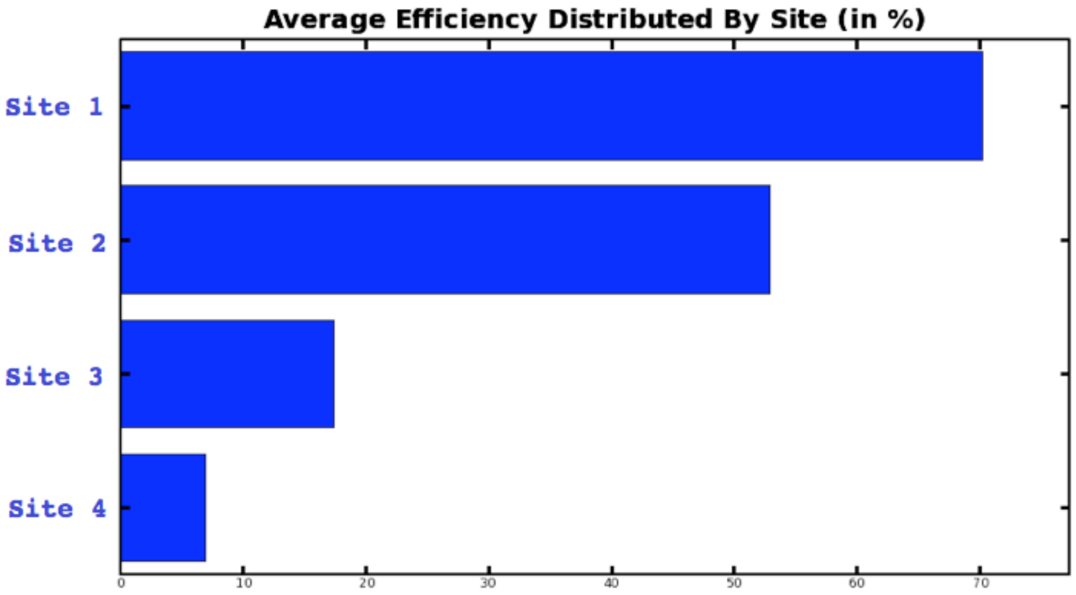
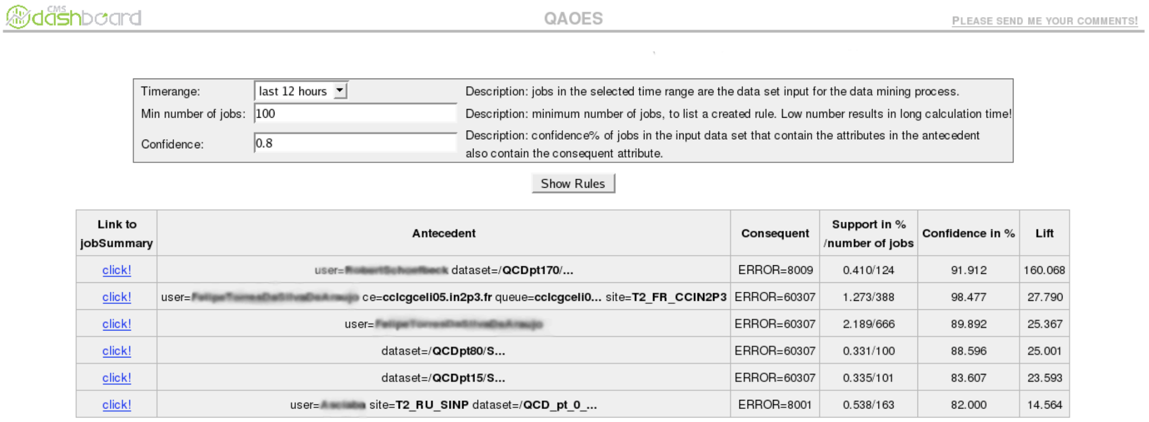


Fig 2. Average efficiency of a single task, measured as the CPU consumption divided by the wall clock time consumption of the jobs, for jobs run at four different Tier-2 sites. Efficiencies ranged between 7% and 70% for this single task.

4.2. Dashboard Developments - Association Rule Mining

An interesting development project from the CERN IT Division involves association rule mining [8] of the CMS Dashboard job exit codes. The web interface for this tool is shown in Fig. 3. The software considers combinations of "attributes" such as data set, user, site, computing element, etc. and looks for a significantly high probability of a "consequent", thus forming rules that indicate problems. Rules with a high confidence level may alert us to a problem as a site, or with a particular user's code, or with a data set, for example. Such automated tools may become useful in the future of Analysis Operations in quickly identifying problems.



Timerange:	last 12 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:	0.8	Description: confidence% of jobs in the input data set that contain the attributes in the antecedent also contain the consequent attribute.

Link to jobSummary	Antecedent	Consequent	Support in % /number of jobs	Confidence in %	Lift
click!	user=[redacted] dataset=/**QCDpt170/...**	ERROR=8009	0.410/124	91.912	160.068
click!	user=[redacted] ce=**cclcgceli05.in2p3.fr** queue=**cclcgceli0...** site=**T2_FR_CCIN2P3**	ERROR=60307	1.273/388	98.477	27.790
click!	user=[redacted]	ERROR=60307	2.189/666	89.892	25.367
click!	dataset=/**QCDpt80/S...**	ERROR=60307	0.331/100	88.596	25.001
click!	dataset=/**QCDpt15/S...**	ERROR=60307	0.335/101	83.607	23.593
click!	user=[redacted] site=**T2_RU_SINP** dataset=/**QCD_pt_0_...**	ERROR=8001	0.538/163	82.000	14.564

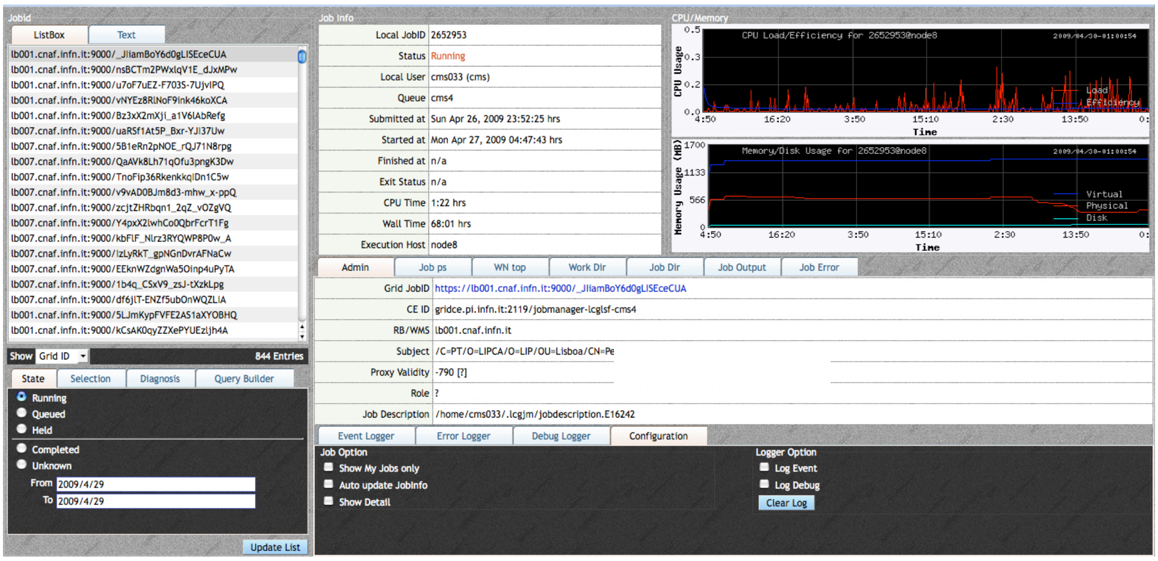
Fig 3. Web interface for Association Rule Mining in the CMS Dashboard.

5. Reference Sites and Real-time Job Monitoring

When an analysis job has a problem, it would be very useful to run that job over a validated data sample at a site where you can have real-time job monitoring, something which is generally not available especially with grid jobs. In this way it would be possible to quickly differentiate between user problems and site issues.

A reference site should therefore be a site with enhanced job monitoring capabilities to allow detailed debugging of analysis jobs even as they are running. The monitoring tools should be available to users via some web or grid interface. Reference sites should host standard data samples validated for particular versions of the CMS software. There should be a queue for quick job turnaround, and jobs should be short. The reference system (or systems) is a service that is not foreseen to have 100% reliability, but is provided as a service to the CRAB user community.

First steps were made in constructing such a sites by installing enhanced job monitoring software [9]. A snapshot of the web interface is shown in Fig. 4. In this interface job CPU and memory consumption are seen, along with various details about the job, with links to the process tables, job output, etc. Interfaces are being developed for the LSF, PBS and CONDOR batch systems.



lb001.cnaf.infn.it:9000/_JliamBoY6d0gLISEceCUA
lb001.cnaf.infn.it:9000/nsBCTm2PWxlqV1E_dJxMPw
lb001.cnaf.infn.it:9000/u7oF7uEZ-F703S-7UjvlPQ
lb001.cnaf.infn.it:9000/vNYEz8RlNoF9ink46koXCA
lb001.cnaf.infn.it:9000/Bz3xX2mXji_a1V6lAbRefg
lb007.cnaf.infn.it:9000/uaRSf1At5P_Bxr-YJI37Uw
lb007.cnaf.infn.it:9000/5B1eRn2pNOE_rQJ71N8rpg
lb007.cnaf.infn.it:9000/QaAVk8Lh71qOfu3pngK3Dw
lb007.cnaf.infn.it:9000/TnoFip36RkenkkqlDn1C5w
lb007.cnaf.infn.it:9000/v9vAD0BJm8d3-mhw_x-ppQ
lb007.cnaf.infn.it:9000/zcjtZHRbqn1_2qZ_vOZgVQ
lb007.cnaf.infn.it:9000/Y4pxX2lwhCo0QbrFcrT1Fg
lb007.cnaf.infn.it:9000/kbFlF_Nlrz3RYQWP8P0w_A
lb007.cnaf.infn.it:9000/IzLyRkT_gpNGnDvrAFNaCw
lb007.cnaf.infn.it:9000/EEknWZdgnWa5Oinp4uPyTA
lb007.cnaf.infn.it:9000/1b4q_CSxV9_zsJ-tXzkLpg
lb007.cnaf.infn.it:9000/df6jlT-ENZf5ubOnWQZLIA
lb001.cnaf.infn.it:9000/5LJmKypFVFE2AS1aXYOBHQ
lb001.cnaf.infn.it:9000/kCsAK0qyZZXePYUEzljh4A

● Running
○ Queued
○ Held
○ Completed
○ Unknown

From 2009/4/29
To 2009/4/29

Local JobID	2652953
Status	Running
Local User	cms033 (cms)
Queue	cms4
Submitted at	Sun Apr 26, 2009 23:52:25 hrs
Started at	Mon Apr 27, 2009 04:47:43 hrs
Finished at	n/a
Exit Status	n/a
CPU Time	1:22 hrs
Wall Time	68:01 hrs
Execution Host	node8

Grid JobID	https://lb001.cnaf.infn.it:9000/_JliamBoY6d0gLISEceCUA
CE ID	gridce.pi.infn.it:2119/jobmanager-lcglsf-cms4
RB/WMS	lb001.cnaf.infn.it
Subject	/C=PT/O=LIPCA/O=LIP/OU=Lisboa/CN=Pe
Proxy Validity	-790 [?]
Role	?
Job Description	/home/cms033/.lcgjm/jobdescription.E16242

Job Option
□ Show My Jobs only
□ Auto update JobInfo
□ Show Detail

Logger Option
□ Log Event
□ Log Debug

Fig 4. Screen shot of the Job Monitor web interface.

6. Examples of Analysis Operations

During the task force we did some Analysis Operations exercises to identify particular problems. One such exercise resulted from a study comparing the CPU and wall clock usage that were being reported to the CMS Dashboard with the accounting from the local batch system at a Tier-2 site to see if they were consistent.

In Figure 5 the wall clock time usage of CMS analysis jobs is shown, along with the usage of the CMS Monte Carlo production and users from other virtual organizations. The local accounting was found to be reporting consumption of resources about 10% higher than what was being reported to the CMS Dashboard, suggesting that about 10% of analysis activity is being conducted outside of the framework of CRAB. An example of such an analysis activity could be n-tuple analysis with privately developed code.

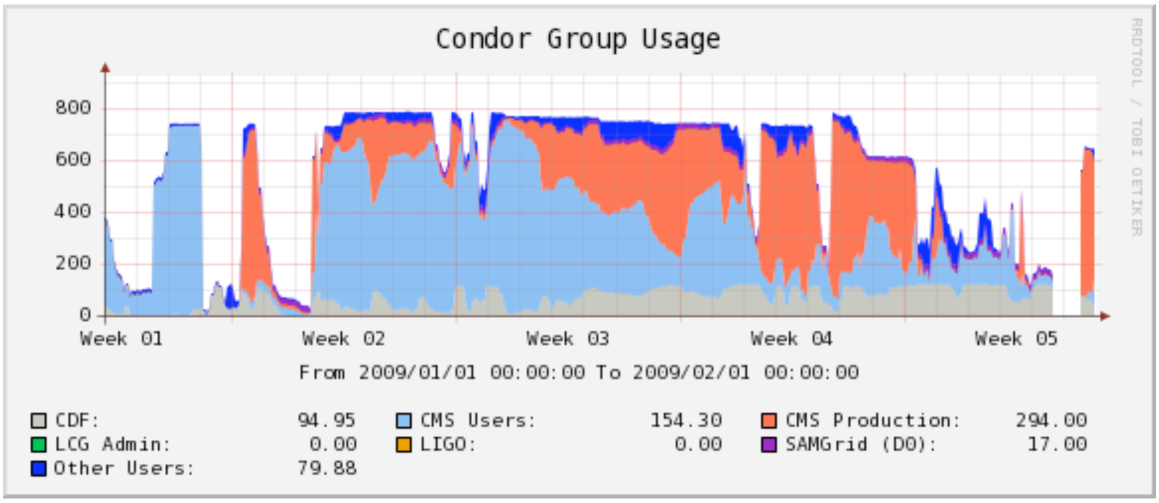


Fig. 5. Wall clock time usage at a particular Tier-2 site during January 2009. The light blue corresponds to CMS analysis users only.

An understanding of jobs that terminate with the status “Unknown” in the Dashboard also came from this exercise. “Unknown” means that the job did not return an exit code. At this site, we found out that jobs that run longer than 36 hours are killed by the local batch system, resulting in an “Unknown” exit code. In the month of January 2009, 2% of the jobs ran into this 36-hour time limit. In contrast, a typical analysis job runs for 40 minutes. Doing the math, one can calculate that the 2% of jobs that consumed 36 hours each used 52% of the resources given to CMS in that month. The 98% of jobs that used 40 minutes each consumed the other 48%.

An analysis of the jobs’ standard output and error files revealed that about 1/3 of the jobs were stuck in the final stage out of the user files, about 1/3 were just very long jobs, and the remaining third were jobs that were caught in a loop using lots of CPU but making no progress, stuck reading a particular data file and consuming no CPU, or otherwise problematic.

While a stage out timeout could take care of 1/3 of the problem of wasted resources at this site, the rest represent a serious issue. At present there is no matching between the resource needs of the jobs either in the client or in the CRAB Server, and the resource limitations at the sites. Even though such limitations of between 24 and 48 hours are common at sites, only a few sites actually publish such limits.

From this study the task force concluded that there are two areas where improvements can be made. The first is in the stage out of user files. When the stage out process hangs, the process needs to be killed or restarted. Some stage out processes are in fact successful while the return code is mis-reported as a failure, and vice-versa. Timeouts and alternative solutions for failures need to be implemented.

The second area for improvement could be matching CPU and WC time limitations at sites with the needs of CRAB jobs. This could be done within the CRAB Server as well. More worrying is that there are sites with no effective limitations on job lengths, and these sites could conceivably waste 100% of their resources if not monitored.

Clearly, this sort of analysis ought to be done not just for one site at one time, but rather become part of the routine metrics effort in Analysis Operations. Thus, one should not draw too strong conclusions from this single exercise.

7. Analysis User Survey

One of the central parts of the task force was the conducting of an experiment-wide analysis user survey. The survey contained questions on user experience and expectations in using CRAB, debugging problems, tutorials, etc. Of the ~800 analysis users in 2008-2009 (users who submitted at least one job) 226 people filled out the survey when it was conducted in February and March 2009. The average level of experience doing analysis in CMS was between 1-3 years, and almost 90% of analysis users reported using the CRAB software to carry out their analysis.

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. The main reason cited for not using the CRAB Servers is lack of familiarity with the tools. In general, the survey found that with newer features of the analysis infrastructure such as CRAB Server, or monitoring interfaces such as the CMS Dashboard, most users that are not using them do not because of unfamiliarity with the tools. This underscores the importance of education and documentation as new functionalities and management are introduced into Analysis Operations.

One of the major transitions in analysis work in the past year has been the migration of analysis to the Tier-2 centres and the stage out of user data back to the Tier-2 storage elements. For the staging out of user results at the end of analysis jobs, 44% of users surveyed reported using this stage out to the Tier-2 sites, with the rest still either staging out to the CASTOR storage system at CERN or using a mechanism for retrieving small output files back to the submission machine. The task force found that problems staging out users' output files is one of the areas where improvement was needed in the infrastructure of Analysis Operations.

Almost 80% of users reported that they would use a "Reference Site" with enhanced job debugging and monitoring capabilities to allow the diagnosis of problems with CRAB analysis jobs.

8. Transition to Analysis Operations

Most of the technical requirements to do effective Analysis Operations exist today or are in the development pipeline. Already existing are the Dashboard for job monitoring and archiving, an active community of experts for feedback, and education and documentation efforts for CRAB, CRAB Server, and tutorials and workbooks. Also in the pipeline are reference sites for fast, detailed job debugging with real-time job monitoring.

An Analysis Operations team can do more active management, improve efficiency and interface between the various core activities. The following sections outline what an Analysis Operations team might look like in the future:

8.1. Metrics Development

This activity would be to develop and report on various metrics to understand the system as a whole and to identify problem areas, at least on a weekly basis. It involves coordination with the developers of monitoring software.

8.2. Data Placement and Data Quality Operations

CMS allocates storage space at the Tier-2 sites for centrally controlled areas (30TB per site) as well as physics group controlled space (30TB per group). The process of allocating data to these areas needs to be done routinely. Data integrity and quality checks (data reading) and stage out (data writing) problems are also an important aspect of this part of Analysis Operations. Popular data sets can be replicated at multiple sites when efficiencies are observed via the metrics effort, for example.

8.3. CRAB User Support Operations

The current user support falls heavily on the CRAB developers themselves, which probably will not scale to an order of magnitude more users. A dedicated team of experts for pro-active job debugging and user support is needed. This is in addition to and in coordination with the traditional mailing list based user support community. A certain amount of this work (such as checking daily Analysis Operations metrics) possibly could be done by the people doing offline computing shifts.

8.4. Reference Site Commissioning

Although part of CRAB User support, the commissioning of the reference sites could be sufficiently time consuming especially early on in Analysis Operations that it warrants its own dedicated effort.

8.5. Integration

There are many stakeholders whose activities contribute to the success (or failure) of Analysis Operations. These include the physics groups, the software development communities for CRAB, PhEDEx, the Dashboard as well as other monitoring efforts, the sites where analysis takes place, education and documentation efforts, and computing management, just to name some. Communication and coordination between these various groups and with the Analysis Operations group itself is essential for success. A dedicated person to act as an interface between these various groups would make sure that changes and developments are done in coordination and with the full knowledge of the other groups involved.

8.6. CRAB Server Operations

As more and more analysis activity takes place via the CRAB Server, there will need to be operations support most likely locally at the site where the CRAB Server is operating. Reliability and stability of CRAB Servers could improve through sharing of information and cooperation as the infrastructure evolves.

9. Summary and Conclusions

The Analysis Support Task Force evaluated the current methods of user support and conducted an experiment-wide analysis user survey. While most of the technical requirements to do effective Analysis Operations exist today or are in the development pipeline, changes in the way we do analysis support are necessary to serve the growing user community. A plan for how an Analysis Operations team might be organized was presented.

References

[1] The CMS Dashboard web interface can be found at: http://dashboard.cern.ch/cms

[2] D. Spiga et al., “The CMS Remote Analysis Builder (CRAB),” HiPC 2007, Goa, India, December 2007, published in Lect. Notes. Comput. Sci. 4873:580-586, 2007.

[3] D. Bonacorsi et al., “PhEDEx High Throughput Data Transafer Management System,” CHEP06, Bombay, India, February 2006.

[4] R. Egeland et al., “Data Transfer Infrastructure for CMS Data Taking,” ACAT08, Erice, Italy, November 2008.

[5] Y. Perrin et al., “The LCG Savannah Development Portal”, CHEP04, Interlaken, Switzerland, September 2004.
See also the web interface at: https://savannah.cern.ch/

[6] “The CMS Computing Model”, CMS NOTE/2004-031.

[7] A. Afaq et al., “The CMS dataset Bookkeeping Service,” 2008 J. Phys.: Conf. Ser. 119 (2001) 072001.

[8] For a description of the project, see:
https://twiki.cern.ch/twiki/bin/view/ArdaGrid/AutomaticFaultDetection
The CMS implementation can be found at:
http://dashb-cms-mining-devel/dashboard/request.py/rules

[9] S. Sarkar et al., “A Grid Job Monitoring System”, CHEP09, Prague, Czech Republic, March 2009.
See also https://gridse.sns.it/jobmon/pisa, for example.