Computing technical program: Annual Review

Stefano Lacaprara

INFN Legnaro

CMS week, Computing session, 13 april 2005

Summary

CRAB CMS Remote Analysis Builder
user oriented tool to grid submission and handling of analysis jobs

BOSS Batch Object Submission System
Interface with scheduler (grid or not) with logging, bookkeeping and monitoring

Dashboard Provide single point of access to monitoring information for CMS jobs and resources.

CRAB

Goal of the project

provide an user friendly front-end for end user interaction with the grid for CMS, including interaction with data management, middleware, remote computing element, basic monitoring functionalities, etc.

Status

In production. Used since more than a year by end users.

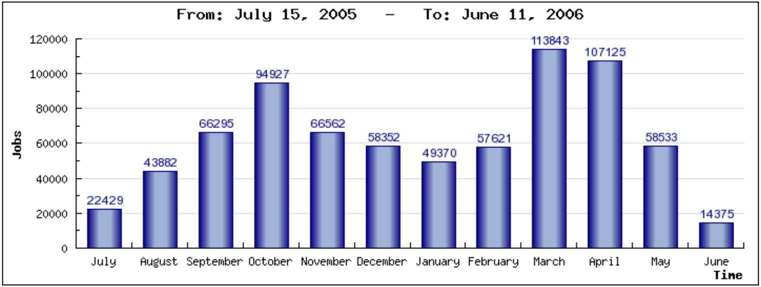
CRAB

- Support DBS/DLS Done Available both for old (ORCA) and new EDM (CMSSW). Fully implemented and available to users (not yet as default). Side remark: it takes a lot of time to switch between two implementations of a component from operation point of view.
- Support for CMSSW Done Complete workflow works, using DBS/DLS, site local config, trivial catalog etc . . . (to be released soon)
- Integration with BOSS CRAB makes full use of BOSS (version 3.6.4). Integration with BOSS4 is basically completed.
- gLite3.0 Standard job submission tested and working. Bulk submission tested by BOSS team, still some problem with 3.0, works well for 3.0+fix (to become 3.1, when?) Should therefore be available out-of-the-box for CRAB.

CRAB

- Interoperability with LCG - OSG Fully supported submission, via Resource Brocker to OSG sites (as well as LCG ones, of course). Transparent to final user, needed some work on CRAB side.
- Dashboard Full set of information about task, job, application, status, etc . . . sent to Dashboard, both from User Interface and from Worker Node
- Multiple job type (FAMOS) Supported: meant mainly as an exercise to understand how difficult is to introduce a different job type wrt ORCA. Very useful for CMSSW integration.

CRAB



reached 100′000 jobs/month: daily record is about 10′000 jobs.

CRAB

- Manpower 6 people involved, ~ 3 *fte*: adequate.
- User support: this is crucial activity and takes a significant fraction of developers time. Still lacking a CMS-wide user support even if more people (especially from sites) is now involved.
- Plan for 2006
  - Boss4 integration (basically ready)
  - gLite + bulk submission (should be more or less ready once boss4 is in operation)
  - rpm packaging
  - Support for advanced workflow as alignment one.
  - Automatization of the handling of user task (status, get output, resubmission in case of failure, post-production step such as merging, etc . . .)
- CSA06: basically ready. Functionalities present, expected rate ($\mathcal{O}(10^4)$ J/day) already achieved during "normal" operation.

BOSS

Goal of the project

Provide infrastructure for logging and bookkeeping of jobs and scheduler interface.

Status

BOSS v3 already used since long time in CRAB as well as in ProdAgent. New version (v4) integration in both projects in advanced status.

BOSS

- Task concept, set of similar jobs with small differences (such as input or parameters).
- A task can be described with iterators in a parametric way: match perfectly parametric jobs in gLite bulk submission.
- Support for linear chain of different application in a single job: for complex workflow.
- Interface to client is fully `xml` based.
- Monitoring information retieved from EDG/gLite Logging and Bookkeeping

BOSS

- fully interfaced to following schedulers: fork, LSF, CERN-LSF, EDG, GLITE, CondorG
- Support for gLite bulk submission with both parametric jobs or job collection, using either gLite CLI or python API.
- First set of python/C++ API
- Real Time monitoring (collecting info from worker node), using a third party server
- RT monitoring interfaced also to MonaLisa.

Dashboard

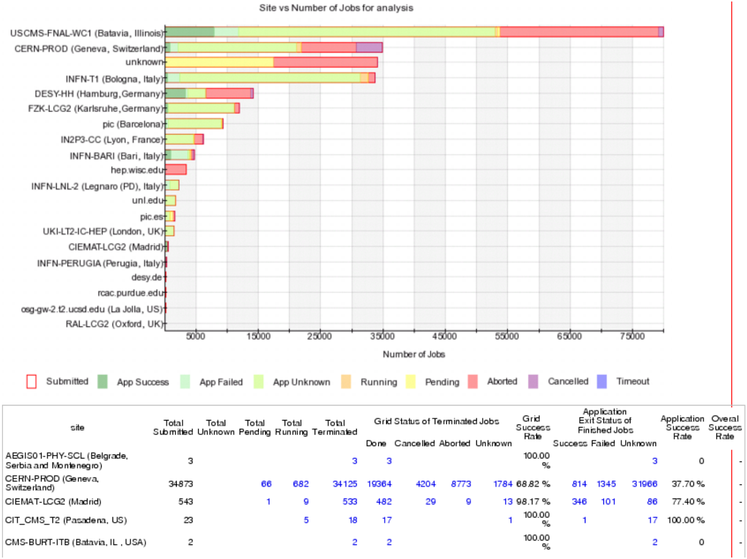
Goal of the project

Provide single point of access to monitoring information for CMS jobs and resources.

Status

In operation. Continuos feedback to improve the quality of information monitored.

Dashboard



site	Total Submitted	Total Unknown	Total Pending	Total Running	Total Terminated	Grid Status of Terminated Jobs				Grid Success Rate	Application Exit Status of Finished Jobs			Application Success Rate	Overal Success Rate
						Done	Cancelled	Aborted	Unknown		Success	Failed	Unknown		
AEGIS01-PHY-SCL (Belgrade, Serbia and Montenegro)	3				3	3				100.00 %			3	0	-
CERN-PROD (Geneva, Switzerland)	34873		66	682	34125	19364	4204	8773	1784	68.82 %	814	1345	31966	37.70 %	-
CIEMAT-LCG2 (Madrid)	543		1	9	533	482	29	9	13	98.17 %	346	101	86	77.40 %	-
CIT_CMS_T2 (Pasadena, US)	23			5	18	17			1	100.00 %	1		17	100.00 %	-
CMS-BURT-ITB (Batavia, IL , USA)	2				2	2				100.00 %			2	0	-

Dashboard

- Different DB backend: Oracle instead of Postgres
- New schema and optimized DB feeding part
- Flexibility to add new information source
- Currently 3 sources are used: MonAlisa, RGMA and IC real-time RB monitoring DB.
- Started to develop the feeder for BDII, to get info about sites.
- User web-interface not yet optimized: possible to perform complex query, but in a long scale most of the use cases should be covered by the precooked views.

Dashboard

- Collected info about task, job.
- Runtime processing info, including detailed breakdown of failures during wrapper processing, grid status, application status, available resources, etc . . .
- Measuring of the job processing rates
- Collecting information of usage of the resources and job processing latencies
- Following failures caused by different nature
- Already useful for user support team, site managers and individual users
- Performaces should scale to submission rates of SC6

Dashboard

- Manpower: 3FTE, adequate
- Plan:
  - Enable OSG jobs monitoring (done)
  - gLite interface
  - Precooked view (aka coffee view)
  - ProdAgent jobs (done)
  - Designing of the common python-based layer for DB access
  - Setting up the testing procedure and constructing of the site-sanity-check page
  - Better grid failure diagnostics
  - T0 monitoring (to be discussed)
- Basic functionality required for CSA06 is already in place. Coffee view would be a valuable plus

Conclusion

- CRAB In good shape, extensively used with success by many users to access data remotely.
- BOSS Good progress, well coupled to the needs of the client projects (CRAB and ProdAgent).
- Dashboard Collects and shows lot of informations about CMS jobs and resources: a key element for operation support.

Future

The key element is operation and user support.