Lesson learnt from CRAB

Stefano Lacaprara

Department of Physics
INFN and University of Padova

CMS Week, Computing session, 15 march 2005

Outline

Outline

PubDB development (slide by Alessandra Fanfani)

- Re-organization of the code:
  - Implement base functions to insert, delete and read each of the DB tables: for *administrator*, command-line and browser;
  - Re–factoring allows more flexible managing of code and DB;
- Update:
  - Better support for COBRA redirection variables

```
-VariableName=MyLocation -values=rfio:myhost:/mypath
```

  - Define default CE per each PubDB: can be overridden, if not default taken
  - Data Tier attribute supported
    - Relation between Collection and CollectionType;
    - CollectionType reflect Data-Tier as tacken from RefDB;
- Test is ongoing: release in 1-2 weeks
- Longer term plan in the context of DM/WM discussion

CRAB Status and Future

- CRAB 0_1_0 released last friday
- Main new functionalities:
  - Allow to ship also `src/Data`: strongly required by Higgs and other groups
  - Allow to write output directly to any `gsiftp` (aka `gridftp`) server such as `castor`
  - Resubmit on Grid Abort
  - Other ...
- Future
  - Monitoring still fragile and not friendly
  - Reengineering of core well advanced
  - Integration with Boss (see Claudio's talk): could solve many of the problems described later
  - Workplan for development and early test of new DM/WM system

Outline

The Good

- Actively developed to cope with (many) user requirements;
- Used by many real users;
- Actively used by many PRS end users $\mathcal{O}(10's)$, with little or no Grid knowledge;
- Already several physics presentation based on data accessed via CRAB
- Estimated grand total $\mathcal{O}(10^7)$ events (rough guess!)

Site access

- As expected, most of the problems come from deployment and site access
- Successfully used to access from any UI data at Tiers-1 (and some T2)
  - CNAF (Italy)
  - PIC (Spain)
  - CERN
  - FNAL (US)
  - FZK (Germany)
  - IN2P3 (France): not yet
  - RAL (UK): not yet
  - Tiers-2: Legnaro, Bari, Perugia (Italy)

Outline

Data accessibility

- Data accessibility and completeness is today guaranteed by site admin
- Fully CMS specific
- Some problems found by users in trying to access remote data tracked down to problem of files, publication, catalogs and such
- Actions
- Need more active data validation for the sites
- See Nicola's work on this issue!
- Full set of tools for validation available: define policy to actually use them more widely

PudDB deployment

- PubDB deployed at all sites hosting data;
- Not trivial to have a coherent system, even with fixed version of PubDB;
- Still too many things left in the hand of site admin;
- Actions
- More care in deploying with PudDB also set of tool to actually populate it
- Minimize (as far as possible) site admin intervention;

Site accessibility

- Site dynamically change state: mostly scheduled activity;
- Information is known to Grid BDII (fine!)
- How to propagate the news to users.
- How to negotiate site shutdown schedule if relevant...
- Actions
- Grid already have monitoring and test of site: more integration with CMS to propagate the info;
- Check if possible to setup a CMS dedicated site monitoring, as the “global Grid” one, but only with CMS site
- Add CMS specific accessibility test?

Resource monitoring

- How to know the load of each site: for user and also for management (eg further distribute data, ...)
- How many of the resources are used by CMS, and how many by others.
- Action: Setup a GridICE server using CMS BDII (need hw and manpower: probably at CNAF, negotiating . . .)

Dataset access monitoring

- Analyze data access pattern.
- Which data have been accessed by users?
- Which datasets need to be replicated?
- How efficiently are we accessing remote resources?
- Action: not easy today.
- Need “central” monitoring, not trivial to setup.
- In next LCG (and gLite) releases possible to put some “tag” in user jdl which will be publish in Logging & Bookkeeping service.
- Will see if usable for data access pattern monitor.
- Could spoils CMS specific site access problems (eg problems with incomplete catalogs, etc...) or problem with specific dataset

CMS SW Installation

- Lot of childhood problems here.
- Installation tool (`xcmsi`) available and working;
- Some problems with deployment in some site (eg IN2P3 failed so far to have ORCA installed, despite of big effort: data published but no sw to access them!)
- Two way of installing sw used:
  - Re–use installed sw (eg CERN, FNAL): must guarantee that installed (and removed) sw is advertised by the CE in Grid fashion and found where Grid users look for;
  - Install via Grid (“special” user `cmssgm`): still triggered “by hand”, must have automatic procedure.

- Actions:
- Define policy for sw installation
  - All site hosting data must have state of the art sw installed;
  - Installation start as soon as RPMS available for new releases;
  - Plus some old versions, removal policy should follow CMS general policy;
    - Push on all site publishing official data (Tier-1, but also Tier-2);
    - Pull (kind of register) for other interested site;
  - All site hosting data to be accessed by Grid must pass LCG test first, then CMS test (see after)

OS on the CE

- Actual situation rather confuse
- each CE can install almost any Linux flavour (most uses SL).
- Then the CE publish what has installed
- NO coherent naming convention (SL, SLC, Scientific, ScientificLinux, Scientific Linux, RedHat...)
- Common naming asked to LCG (and being agreed upon).
- That is not enough: CMS must “validate” each CE we want to use.
  - Validation as last step of SW installation (already foreseen by `cmsi`)
  - Validate OS flavour and version once, and thus validate all site publishing that OS ??
  - Validate all site in any case ??

User support

- Very time consuming: means CRAB actively used!
- Need several levels of support:
  - Pure Grid problems (certificate, problem with Grid services, sites,...);
  - CRAB support;
  - Data access support: problems with catalogs, missing files, problems with MSS, etc...
  - ORCA problems...
- Actions: crab_feedback list used to ask support: need a FAQ section
- Learning how to interact with Grid Support: some iteration with relevant people: general access point is GGUS (http://www.ggus.org/)

Catalogs issue

- Dealing with many catalogs on each site can become a nightmare very soon
- need to dramatically reduce the number of catalogs per site
- put everything which is available at a site in just one.
- Action: Phedex 2.1 is providing a mysql pool catalog, the very same used for data transefer.
- Testing . . .

Input sandboxes.

- Today sent via input sandbox:
  - Configuration files,
  - Job ancillary files,
  - User libraries and executable
- Size limit on InputSandBox $\mathcal{O}(10)$ *MB*
- Use SE for big input stuff: many problems.
  - Which SE?
  - Close to UI (not necessarily defined)
  - Close to CE, not known in advance
  - Probably second order optimization!
  - Must be sure to avoid name clashing (using what user want not some relic from past jobs)
  - Must cleanup everything at the end: when? data lifetime?
  - Should foresee a experiment specific service?

Output produced.

- User wants output on her computer or on a storage accessible from her computer (via posix or any usable protocol, eg RFIO)
- In general not interesting to have output on Grid
- Different for “production” use cases
- If output via output sandbox: user must ask when *Done*
- Query L&B every *x* seconds until job is *Done* scalability??
- Can user be notified when job is finished?
- If storage has the proper server installed (*e.g.* `gsiftp`) possible to just copy the output when done.
- What about ACL? Output written according proxy certificate ACL, which are different from storage ones
- `cms002` need to write on `/castor/cern.ch/user/s/slacapra/...`

Job clustering.

- Typical User job is splitted into several *subjobs* each accessing a fraction of total input data
- Subjobs are identical but for few bits
- Same Input Sandbox, same requirements, etc. . .
- Eventual common pre-job:
  - Stage-in (pinning) of input data from MSS
  - User sw compilation and linking
- Need job cluster (or bulk) seen as a single entity
- Allow bulk operations (submission, query, status, cancel, . . .)
- Also possible to get access to single sub jobs
- SubJob number available at WN level, used by job wrapper
- Several splitting logic possible
  - first iteration done at UI level
  - then at RB level, using Grid data location

Outline

Summary

- Many lesson learnt from CRAB usage;
- First lesson: people is using it
- Second lesson: real effort must be put in deployment
  the more problems are not to be addressed by CMS directly, the better