CMS Usage of batch systems

Stefano Belforte
INFN – Trieste
CMS Computing CO-Coordinator

Outline

- We are here to contribute to an ongoing process/discussion, not to disrupt it

- Short introduction to CMS computing model
- Requirements on batch systems (or more specifically on interface bwteeen grid and batch systems) are :
  - coming from obvious details of obvious functionality needs
  - no hope to be the full list
- I am here mostly to answer your questions

Onle slide summary

- A few groups, mapped to VOMS groups, with different relative share of CMS overall share
- Flexible CPU allocation to those
- Monitor and account with high granularity
- 3 or 4 different CPU time limits
- Access to experiment software distribution
- Posix access to local data
- The "trivial things" that make it work
  - Communication with grid CE works well
  - One beserk job does not harm others on same WN

CMS and the Grid

- At CERN: Tier0 and Calibration Analysis Facility
  - Data reconstruction and other organized activities
    - what reconstruction farms always used to be
  - High priority analysis of few critical samples
    - cahotic analysis of disk resident data
- At 7 large sites: Tier1
  - Data archival, re-reprocessing, secondary sample generation. Mostly organized activity
- At ~30 smaller sites: Tier2
  - Simulation (highly organized activity), and user analysis (very cahotic and random, fast response)
- All of this via batch jobs submitted via Grid CE's

Overlay CMS needs on batch system

- CMS jobs will always be of several "cathegories"
  - Which belongs to different user (groups)
  - Which have different priorities
  - Which have different work patterns
- Nothing unexpected
  - 2000+ collaborators in CMS: some structure needed
  - Common tasks, groups, subgroups… down to user
- One batch queue will not fit all
- Sites may need, or want to provide some structure
  - Grid must be able to exploit it

Priorities (shares)

- At site X, allocate CMS-CPU as:
  - 20% for MC production
  - 30% for Physics group A
  - 50% for User group B
- "users" described as VOMS groups/roles
- shares = minimum guranteed plus expansion, not fixed list of nodes
- This within: CMS has 60% of total
- But if B is the only customer, gets 100%
- These numbers (and groups) must change
  - With ~ 1day from request to enforcment

More on shares

- Shares need to be fair
  - All users in a (VOMS) group have equal access
  - Globally, on all of the grid
- Shares describe steady state behavior in equilibrium
  - Time to reach this should be ~1day
- Global fair share
  - Current prejudice we have is that local fair share at all sites will produce global fair share
- Global setting
  - 30% of all resources for Group C should be used by users Joe and Jane who do VeryUrgentWork

More flexibility

- There will be other "special" use cases
- Test jobs:
  - I need to debug my (Grid) script
  - Can not do in local mode
  - Need fast (min) turnaround
- Long/Short jobs
  - Job latency should be proportional to duration
- User's priority reordering
  - Yesterday I submitted 100K jobs
  - Today I have an urgent thing to do

How to do it ?

- CMS does not currently believe that the way to go is to have our own scheduler exploiting Condor-glide-in-like pilot jobs
  - Very difficult to do all that with a central scheduler, anyhow
- We think most of our share settings is born at the sites
- Think of GRID as being able to adapt-to/exploit-the flexibility implemented at the sites, rather then invasively enforce it
- This puts sites in control
  - Good: CMS Tier2's do like to have a saying in how their resources are used
  - Bad: they have to do something

So… a plan is being shaped

- Site should setup shares for ~3 CMS groups
  - E.g. by mapping VOMS groups to Unix GID
- Site should not recycle UID/GID fast
  - Or fair share will not work
- Site should allow jobs to have different time limits (within each group)
  - Short/medium/long (+ test)
- Site will have some way for shares and assignements to be changed
  - Phone, e-mail, wget, VOMS, passive, active…Sites must provide monitor/account to verify it works
- Some of this setup on US-CMS now, more tests on EGEE sites coming (under EGEE's TCG umbrella)

Things without a “plan” so far

- Test jobs
  - Debugging “grid scripts” calls for few minutes turnaround
  - CDF dedicates to this 5-th excution slot on WN
- Long jobs do not starve short jobs
  - e.g. CDF allows long jobs only to fill X% of farm
- Modifying shares “centrally”
  - changing VOMS/unix group mappings
  - changing batch system parameters
    - relative shares of CMS groups inside CMS share
- Accounting/Monitor by Group and User

Passing req's from CE to Batch system

- CPU limit: needed
- RAM size: really hope is never needed (1GB/process), but of course nice to have
- Local disk: some scratch space will be needed, a few GB, should not even need to ask.
- More important for a job to be able to know what is available and adapt. More coordination with middleware needed here.

- What did I forget ?

Onle slide summary

- A few groups, mapped to VOMS groups, with different relative share of CMS overall share
- Flexible CPU allocation to those
- Monitor and account with high granularity
- 3 or 4 different CPU time limits
- Access to experiment software distribution
- Posix access to local data
- The "trivial things" that make it work
  - Communication with grid CE works well
  - One beserk job does not harm others on same WN