Review of problems seen in LCG-2: Minimum site requirements

Ian Bird

GDB

13th July 2004

Overview

- Review of issues in LCG-2 data challenges
  - From operations point of view
  - From experiments' point of view

- Appropriate levels of resources
  - Summary of experiments' needs
  - Request for resource levels to be made available

Operations issues – 1

- Site missing from BDII (6-8 sites)
    - Site GIIS down or provides wrong information
    - Known MDS problem – replace GIIS with BDII
- Job submission problems
    - PBS issue usually
        - Non-shared filesystem – wrong config of ssh keys
        - Shared filesystem – NFS issue (clock sync)?
    - Usually only a few nodes at a site with problem → BUT becomes a "black-hole"
- Replication problems
    - Site SE missing from info system – GRIS dies (MDS: use BDII)
    - Network/firewall problems
        - Wrong firewall config, or gridftp problem with multiple streams
        - (wrong BDII configured in RM – no longer an issue?)

Operations issues – 2

- Lack of operational tools to understand problems
  - Missing in middleware: interfaces for system management
- No accounting
  - We really need this urgently
- No statistics on usage/failures, etc.
  - Need to develop these tools
  - Need a much better top-down view of status and simple way to trace problems
- Many sites want to move away from OpenPBS
  - Bugs, want better scheduling
- Need better upgrade process
  - Hard to upgrade during production

Compute Element – Batch systems

- Batch systems – vs GLUE (or any fixed schema) vs CE vs RB
  - Batch systems like LSF very rich set of functionalities/sharing etc
  - Does not easily map to a (finite size) fixed schema
  - RB needs to be able to make use of published information
- Can't assume homogenous clusters
  - Globus model assumes homogenous clusters – very few are
  - Need separate CEs for each sub-cluster
- Can't see per VO free slots/ jobs running
  - Need separate CEs per VO
  - Need VOMS to really map to correct VO
  - BUT LSF/PBS cannot easily provide this in a shared farm (scheduling too complex)
- Missing (consistent) normalisation of CPU specs and queue lengths
  - We have published instructions on what sites must do
  - Has to be followed through

Resource Broker

- Use of ranking algorithms
  - Complex behaviour, not necessarily what is expected
  - But seems to behave correctly
- No bulk operations for submission/status
  - Missing functionality – really needed for big batch productions
- Speed of submission (1s response, 15s submission)
  - But does not die/choke/fail
  - Much faster now since can use BDII for ranking

+ bugs found and fixed

- Expiring and shared proxies
- File descriptor leak in C++ API
- Connection dropped – re-started all jobs
- Pointer to initial working directory

POOL/RLS Experience

(Dirk Düllmann 31/3 GDA meeting)

CMS Data Challenge showed clear problems wrt to the use of RLS

- Partially due to the normal “learning curve” on all sides in using a new systems
- Some reasons are
  - Not yet fully optimised service
  - Inefficient use of language bindings and query facilities
- POOL and RLS service people works closely with production teams to understand their issues
  - Which queries are needed?
  - How to structure the meta data?
  - Which catalog interface?
  - Which indices?

POOL/RLS Experience

(Dirk Düllmann 31/3 GDA meeting)

- But poor performance also due to known RLS design problems!
- File names and related meta data are used in one query
  - RLS split of mapping data from file meta data (LRC vs. RMC) results in rather poor performance for combined queries
  - Forces the applications (eg POOL) to perform large joins on the client side rather than fully exploit the database backend
- Many catalog operations are bulk operations
  - Current RLS interface is very low level and results in large overheads on bulk operations (too many network round-trips)
- Transaction support would greatly simplify the deployment
  - A partially successful bulk insert/update requires recovery “by hand”
- These are not really special requirements imposed by POOL
  - Still acceptable performance and scalability needs a catalog design which keeps the data which is used in one query close to each other
  - Try to work around some of this know issues on the POOL side

- …and…Java clients → clients based on C++ API

General issues

- Jobs “cancelled”/aborted for unknown reasons
  - see site configuration issues, and RB bugs fixed
- Lack of tools and information about failed jobs
  - Needed to involve site managers
  - GridIce monitoring is opaque
  - Tools are missing
- Lack of consistent storage grid interfaces
  - Hidden by RM, but …
- Lack of disk space on SEs
  - ➔ see resource requirements
- Unreliable data transport layer
  - Gridftp not robust
  - ➔ Need reliable data transfer service
- Large number and small size of files
  - Problem will only get worse – needs layer between tape and apps

Summary of resource needs

	ALICE	ATLAS	CMS	LHCb
SE GB/cpu	30	30-40	50	?
WN Disk GB/job	2.5	2.5	1	5
WN memory MB/job	600	600 (1GB for pileup at selected sites)	500	500
Longest job (@ ~2 GHz)	8h	24h	72h(exceptionally 1 week for Oscar?)	24h
SW installation space (GB)	0.5 GB in shared area	15GB	0.7 GB (prod) 20GB (analysis) in shared area	0.5 GB production 3 GB analysis

Comments

- SE GB/cpu:
  - Space needed on the local storage element in GB per cpu in the cluster. All experiments need similar amounts.
  - A comfortable limit would be between 1.5 and 2.5 TB per 50 CPU per experiment supported.
- WN disk GB/job:
  - Space needed on each worker node in GB for each simultaneous job. This is scratch space that should be available to each job.
  - With recent systems with large disks this should really be no issue.
- WN memory MB/job: RAM needed for each job.
  - To avoid swapping cluster nodes must have this amount of RAM available for each simultaneous job running on a machine, and sufficient swap space to go with it.
  - If the RAM is not available then the number of jobs that can be run on a machine should be limited appropriately.
- Longest job:
  - Length of the longest jobs measured in hours on a 2 GHz cpu.
  - Batch queues need to support jobs of this length *scaled by the site's slowest cpu*.
  - Thus, queues need to be able to support week-long jobs.
- SW installation space:
  - How much space in GB each experiment needs for its software installation.
  - This includes the installation of multiple software versions.
  - Usually shared filesystems

Requirements – for site to contribute to experiments' DC/production

- Storage element space: 30-50 GB per cpu
- Worker node disk: 5 GB per concurrent job
- Worker node RAM: at least 500 MB per concurrent job
  - More for ALICE and some ATLAS needs
- Batch queue lengths: > 72h @ 2 GHz equivalent
- Experiment software installation: 20 GB per experiment

- It is essential to ensure these resources are available urgently
  - Less will limit the usefulness of the site for LCG DC's and production
  - Experiments are likely to use only sites with sufficient resources
- Also ensure that information is advertised correctly
  - Respond to change requests ops team is asking for

Summary

- Need to (urgently) put resources in place
  - Cpu vs SE disk; scratch space; WN memory, queue lengths
- Storage issues:
  - Consistent interfaces, missing managed storage on SE
  - Large number of small files vs long jobs
- Unreliable data transfer
- RLS/file catalogues
- Lack of tools for
  - Operations support
  - Application debugging
- Model of RB/CE vs Batch systems, heterogeneous clusters
- Many bugs found and addressed