LCG-2 Component & Service Evolution

Ian Bird

CERN

GDB

13th July 2004

Background & disclaimer

- Continue to develop LCG-2 service to deploy and validate basic underlying infrastructure services essential to have in place

- Cannot wait for new gLite developments – but ensure we are aligned
  - What we do now may/will be replaced but there is still much to learn and understand
  - What we propose is consistent with gLite developments
  - Underlying system-level issues (firewalls, security, network behaviour, error handling, …) need to be addressed now
  - Much is learned in the DC's – need to validate solutions to those problems
  - Intend to deploy/validate gLite solutions in parallel (on pre-production service)

❖ DISCLAIMER: what is presented here is what we recognise as missing or broken in LCG-2
  - Some solutions are suggested – but they are not the only possible solutions

Functional areas that need effort

- Data management
- Monitoring frameworks
- VO management tools
- Porting to non-RH73
- Operations and user support tools
- IP connectivity
- Interoperability
  - see next agenda item

❖ Our task is to find solutions and deploy them
  - Preferably existing solutions,
  - …but undertake modest development where needed

Data management

Reliable Data Transfer – management

- Implementation:
  - Currently investigating/testing 3 possibilities:
    - TMDB (from CMS) – together with EGEE and CMS
      - We could use “as-is”, EGEE want to adapt to new architecture
    - Stork (from VDT)
    - pyRFT (python implementation of Globus RFT)

- All of these could be used with little adaptation, allowing us to focus on system-level issues
  - Optimising performance, security issues, etc

- Effort:
  - 1-2 people in GD team, together with CMS and gLite
  - Work in testing has started, set up test framework to FNAL and Nikhef
    - Already being done in context of basic network infrastructure testing

File catalogues

- This is what we believe needs to be addressed

  – based on CMS/ATLAS/POOL experience: -

Key is to simplify, concentrate on functionality and performance

- Single central file catalogue providing:
  - GUID → PFN mappings – no attributes on PFNs
  - LFN → GUID mappings – no user-definable attributes (they are in metadata catalogue)
  - System attributes on GUID – file size, checksum, etc
  - Hierarchical LFN namespace
  - Multiple LFNs for a GUID – compatible implementation with EGEE & Alien
  - Bulk inserts of LFN→GUID→PFN
  - Bulk queries, and cursors for large queries
  - Transactions, Control of transaction exposed to user
- Metadata catalogue:
  - Assume most metadata is in experiment catalogues
  - For VO that need it – simple catalogue of “name-value” pair on GUID – separate from file catalogue

File catalogues – 2

- Other issues to be addressed:
  - Fix naming scheme (has been source of problems)
  - Cursors for efficient and consistent large queries
  - Collections – in file catalogue – seen as directories/symlinks (or as GUID)
  - GSI authentication …
  - … simple C clients (extend existing C clients)
  - Management tools – logging, accounting, browsing (web based)

- Availability
  - Replication –
    - Address through distributed database project
  - WAN interaction –
    - Several ideas (RRS, DB proxy from SAM)
    - Needed to provide connection re-use, timeouts, retries

File catalogues – 3

- Options:
  - Use existing Alien FC
    - Does not expose GUID
    - Brings in (a large part of) the Alien infrastructure
    - Not integrated with POOL
      - LHCb have not yet done this
  - Use Globus RLS
    - Grid3 and NorduGrid see reliability problems
    - Work ongoing to make it respond to CMS DC04 use-cases
    - Integrate with POOL and respond to main set of requirements ???
      - How close can it get? Timescale?
  - Adapt/rework the EDG RLS
    - Can re-use existing components
    - Complies with gLite model (ensure agree on interfaces)
    - Estimated work involved (prototype end August)

Lightweight disk pool manager

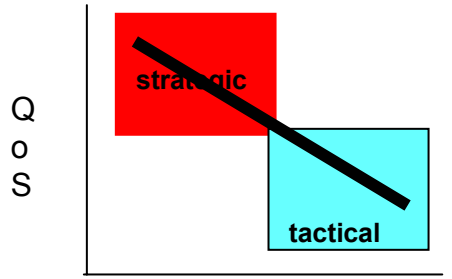
- Recent experience and current thinking gives following strategy for storage access:
  - LCG-2, EGEE, Grid3 all see a need for a lightweight dpm
  - SRM is common interface to storage; 3 cases:

1) Integration of large (tape) MSS (at Tier 1 etc) –
   - Responsibility of site to make the integration – this is the case
2) Large Tier 2's – sites with large disk pools (10's Terabytes, many fileservers), need a flexible system
   - dCache provides a good solution, but needs effort to integrate and manage
3) Sites with smaller disk pools, less available management effort
   - Need a lightweight (install, manage) solution

- We suggest that 3) is missing and is essential to move towards SEs with standard interfaces and behaviour

Disk pool manager – scope

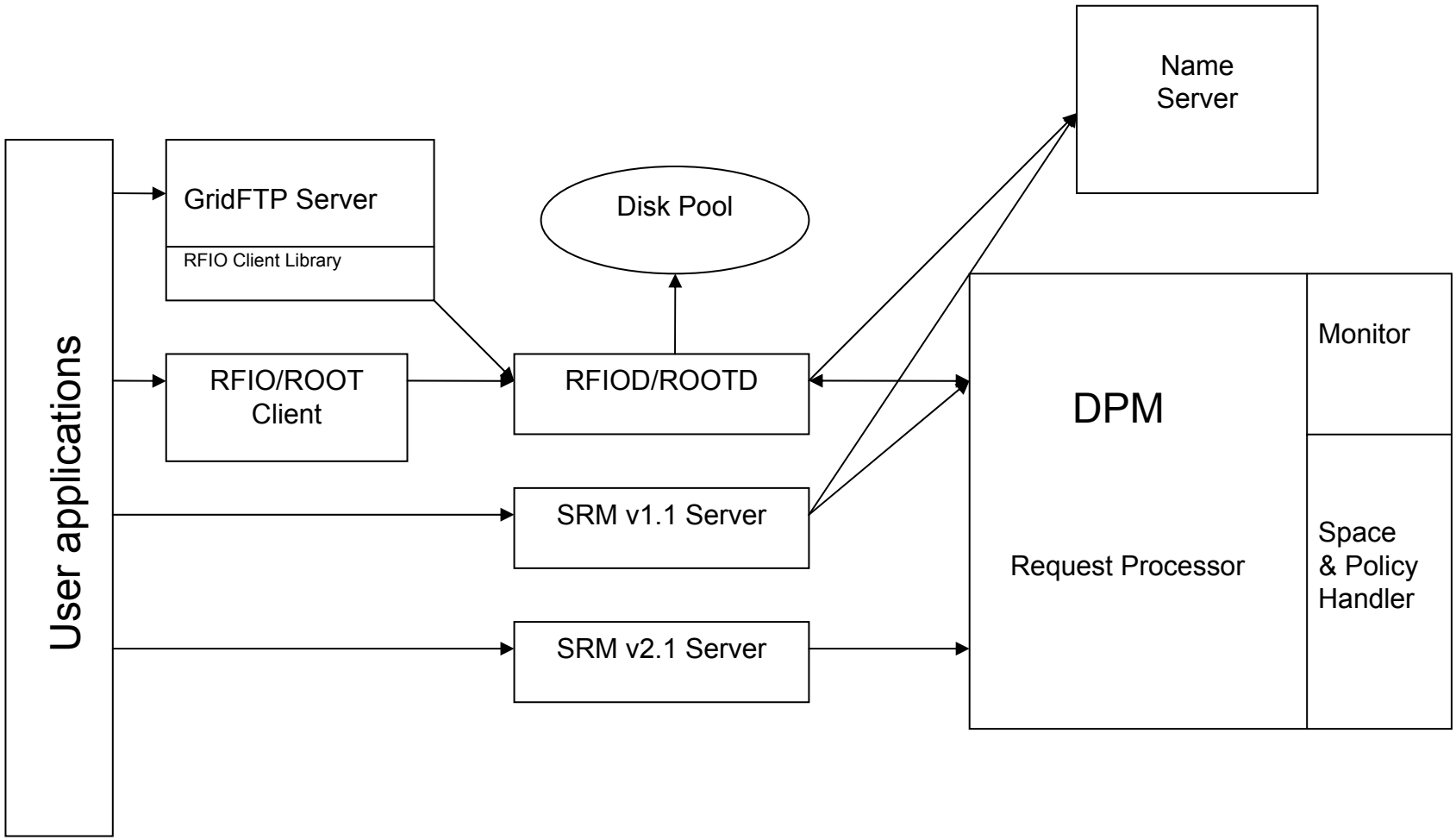
- Small Tier 2 sites
  - 1-10TB of storage, usually system-attached to nodes
  - No SAN architecture
  - No full-time support for storage solutions. Only a fraction of an FTE available to manage the system
- gLite specifies 2 types of SE:
  - Strategic and tactical
    - Tactical corresponds with the missing piece
- EDG “classic SE”
  - Gridftp server + published info
  - Must run on each storage node, each managed independently (cannot add space!)
  - No SRM interface (must use rm tools to hide different SEs)
- dCache
  - DCs → Complex to set up and manage,
    - prohibitive for small sites?



Disk pool manager – potential solutions

- Put more effort into dCache to make it simpler
  - It has taken 7 months to get this far – still do not have a general system that can be deployed easily
  - But is an important solution for large sites with large disk pools
- Look at other solutions
  - DRM: existing implementation not easy to adapt (Corba, …)
  - NEST:
  - …
- Build something new
  - Takes effort, but
  - Can re-use components
  - Aligned with EGEE/gLite plans – could we broaden this collaboration?

Disk pool manager – components



VO management tools

- Want to deploy VOMS
  - Still inconsistencies between LDAP and VOMS VO databases
    - Work in progress
- Need to agree on admin interface
  - Effort/direction in EGEE on VOMS management interface not clear
  - Propose to work with VOM-RS (collaborate with FNAL/US-CMS)
- Deploy incrementally
  - Grid map file built from VOMS
  - Integrate with local authorization for CE
  - SE?
- Long term issue – (for gLite etc)
  - Must have lightweight and simple scheme for creating/removing VOs

Porting to non-RH73

- Done for IA64 and CEL3 (almost Scientific Linux)
  - WN tested; still testing other components
  - Distributions will be available very soon
- Other work ongoing (TCD, QMUL)
  - For other OS
- Want to make WN installation as light as possible
  - Preferably as a simple (small!) tar file that can be installed quickly
    - Access to non-dedicated resources

Monitoring frameworks

- Identified a clear lack of monitoring tools
  - Intend to deploy R-GMA now
    - Permits experiments to use as mechanism to transmit job monitoring/bookkeeping info to central collector
    - Acts as a proxy if MON box at a site (if remote requires outbound IP)
  - Would like to understand also MonaLisa
    - Monitoring from LCG/EGEE level
    - Provide to applications
  - Continue to work with GridIce to make it more useable

Operations and user support tools

- Address needs of system managers, grid operations people, users
  - To better understand the state of the system and its services
  - To better debug problems with jobs, services, sites, etc.
- Much information is available
  - High level tools to pull it together and present it
  - Better use of logged information
  - Improve logging in job wrappers etc. – to aid in bookkeeping and debugging
  - Security audit
- Accounting
  - Is urgent
- Effort funded by EGEE will help address these
  - This work is in progress
- Experiment software installation

IP connectivity

- Important to make progress – providing needed functionality in a more secure way:
- Aspects:
  - Data access (including software), writing data to a remote site
    - All require Replica Manager *service* – there are several initiatives to be investigated as part of improving data management services
  - Publishing information about progress of jobs, general bookkeeping-like information
    - R-GMA – being deployed now – seems a good tool to address some of these issues
    - Already being used by several experiments in this context
    - We will build a generic framework
  - Remote DB access
    - Needs a general db proxy service – addressed by distributed DB project?

Summary

- Many functional areas need to be addressed
  - Some require significant effort
  - Perhaps not all can usefully be addressed in the LCG-2 lifetime
- Continue to add simple useful tools
  - Several provided during DC's
- Work on making the infrastructure more usable and manageable
  - Operations tools will be long-lived
  - Other tools may not work in gLite environment – but we need to understand requirements as input to gLite