Data Access and Data Management

in grids

Jos van Wezel
Overview

- Background [KIT, GridKa]
- Practice [LHC, glite]
- Data storage systems [dCache a.o.]
- Data and meta data
Intro

- KIT = FZK + Univ. of Karlsruhe
- Steinbuch Centre for Computing
- GridKa: German T1 for LHC

About me
• Worked for GridKa 5 years
• Since 01/08 includes data storage at SCC, storage team of 12 people
• Data intensive computing, disk and tape systems, SAN, dCache, SRM
• European Large Scale Data Center: storage of scientific data
Storage of LHC data

- Disk Storage
  - Home-grown parallel files systems with own access protocols
    - dCache, CASTOR, DPM, xrootd
  - Tape Storage
    - Site dependent
    - Sites re-use what is available (TSM, HPSS, Enstor)
- Transfer
  - gridftp for WAN transfers (globus based including auth.)
  - SRM for protocol negotiation
Data (reduction) in HEP

- Detector data
  - RAW - tracks, hits: size ~2.5MB
    - needs reconstruction
  - RECO - detailed reconstructed info, particles
    - suitable for detector studies and reconstruction code development
  - AOD - most used objects for analysis
    - (Analysis Object Data)
    - this is what end scientists use to run their jobs.
  - Skims - selected channels for focused research groups and individual scientists
    - they are just pointers to AOD objects
  - Ntuples - suitable for download to laptop and to run ntuple analysis (apply selections, cuts)

- Calibration and condition data
  - must be available at every job
  - side band data flow using distributed databases
What is current in LHC data handling

- DATA reduction: experiment’s data is split into data tiers orthogonal to the usefulness of a particular activity or step in the workflow
  - grid-wide replicated disk-based storage of most useful for analysis data.
  - limited, scheduled access to the rest of data that is normally kept only on tape

- Grid jobs are sent to where data is
  - no WAN access to data by jobs
  - WAN transfer mostly bulk scheduled transfers

- Non-event data stored using completely different highly reliable and proven technology
  - Oracle streams, web portals, MySQL, apache, squid

- Metadata is very experiment specific and seldom anyone trusts canned solutions
  - almost everything from database to web portals are developed in house
  - Relational databases is a natural choice for metadata store
    - database development needs knowledge and development skills (read: time, money)
Storage in the grid

assume that each computing element (worker node) has local access to storage resources

data light applications
- jobs can be scheduled anywhere
- simple cp of redirected stdout/stderr will do
- shared home or work directory

data intensive applications
- jobs are scheduled where the data is
- jobs are scheduled to prepare access to local data
- storage preparation
  - space for data is reserved
  - data is transferred (to the site or to local scratch)
  - data is ‘pinned’ i.e. locked on a specific storage tier (disk, tape, dvd)
- interaction with storage via storage services are offered by SE
  - SRM
  - file transfer
SRM managing storage in the grid

- provide access to storage resource (via SURLs)
- web based protocol
- policy layer on top of the site policy layer (independent)
- mediates file transfers, relays to file transfer services
- file pinning
- space reservation
- optimise file placement
- transfer protocol negotiation
- handle abort, suspend and resume of transfers
- ACL and directory management for SURLs
SE - Grid Storage Element

SRM

DataTransfer (gridFTP)

ESRF

Physical storage

File repository / catalog

filename → SURL
filename → SURL
filename → SURL

Internet

Diamond

ANKA/SCC

Note: all these connections need public IPs
SRM does not cut it

SRM possibilities: in principle all fair and nice but:
- implementation partly driven by eclectic use cases
- implementation of (some of) the features is slow
- interoperability between implementations (intra and inter) is troublesome (mostly glite grids use SRM).
- Grid-enabled (SRM) storage is not successful
- a lot of problems using SRM/gridftp/globus technologies e.g. transfer problems
Data management middleware
SRM implementations

- dCache
- DPM - Disk Pool Manager
- CASTOR
- StoRM - Storage Resource Manager
- BeStMan

Provide

- SRM for storage management
- transport protocols
- unified namespace over various storage
- monitoring and accounting
DPM – disk pool manager

- developed at CERN
- for ‘smaller sites’
- configuration is stored in a database
- support of ACLs
- France == DPM (except for the French WLCG-T1)
- reportedly reliable and stable operation
StoRM – storage resource manager

- data and meta-data in local file system: local filename = site filename
- based originally on GPFS now adapters for Lustre or XFS
- full POSIX access from worker nodes
- policy based HSM interface to HPSS and TSM
- currently few installations
CASTOR

- CERN developed
- rather complicated
- everything is a LSF job
- all data lands on tape: optimised for tape
- CASTOR itself handles robotics and drives
- implementations at RAL, INFN and Taiwan
dCache

- DESY and FNAL development
- pool nodes manage storage
- separate data and metadata path
- poolmanager manages storage and directs traffic to pools
- back-end for MSS (tape)
- various protocols (gsiftp, dcap, srm, xrootd)
- widely established
- scales to large sites [ but huge resource footprint (cpu, admin) ]
Grid (glite) data management

- VO specific Data Management
- Compute Elements
- FileTransferService
- gfal
- SRM
- GridFTP
- NFS
- dCache
- Transfer MetaData
- FS/cache MetaData
- File system
- Tape system
- Physical disk
- Tape
Non SRM solutions

- **xrootd**
  - SLAC based
  - recently integrated in ROOT environment
  - probably also integration in CASTOR
  - no metadata controller
  - redirector selects asynchronously from 1 to 64 hosts
  - can be setup with hierarchical redirectors (managers)
  - Tape backend

- **IRods**
  - policy based storage system
  - experience and development also at CCIN2P3

- **Cluster file systems**
  - offer large multiple volume storage
  - gdfs (interface to tape)
  - lustre (more or less open source)

- **Gfarm**
  - reference implementation of the grid-filesystem
  - global filesystem, federates storage over WAN links
  - re-uses local (worker-node) disks
  - used in asia – pacific region

- **hot**
  - Cloud, dust, mist and nebula storage
Cloud computing and storage

- Simplify the marketing hype: I see 3 layers through the cloud
  - Layer 1: applications and content (end user interfaces) example: hotmail
  - Layer 2: platform and operating system (grids, interoperability, programming), facilitates layer 1, example: heroku.com, mosso.com (cloudfs),
  - Layer 3: infrastructure (virtual hosting, virtual storage, large scale storage), facilitates layer 2, example: amazon, gogrid.com, linode.com
- Concentrate on the Layer 3: it will probably give you at least a reliable infrastructure
- Cloud infrastructure offerings: S3, GoGrid, Microsoft SSDS
- Forget cloud storage
- This is as much as I can say today, ask me again in a year
Revisit

- Raw data
  - transport via gridftp
  - storage with DPM, dCache
  - disk and tape systems

- Meta – data
  - transport/distribution via experiment specific software
  - use of proven technology
  - LFC
Meta data

- Data replication, relocation, migration are all based on meta-data
- How to find the proper data and not use filenames like these:
  `/mount/datatape/fdr08_run2/RAW/fdr08_run2.0052301.physics_Jet.daq.RAW.o3/fdr08_run2.0052301.physics_Jet.daq.RAW.o3._lb0004._0001.1`

- Currently there is limited meta-data handling support
  - Limited means: unaware of specific requirements
  - Basically just file names (inode info)

- WLCG experiments have cooked their own solutions
  - AMI (ATLAS), RefDB (CMS), ALien (ALICE)
- OGSA-DAI – GGF standard for Grid data(base) access
- **AMGA** – ARDA metadata grid application
  - Several interfaces and front-ends
  - Can replace LFC and for other relational MD handling
  - Strong security (in use for BioMed, ATLAS/LHCb Ganga)
  - File system like arrangement of MD

- Starting from scratch?
  - Probably no way around some development
  - Tools to develop workflow exist (see previous talks)
  - Follow the data
Summary / Conclusions

- Reduce work data sizes
  - match size to requirement of the step in the workflow
- Increase file sizes
  - reduces the effect of large handling overhead in transfers from/to sites and tape
- Bulk data transfer/handling methods exist and can be used
  - use with caution
  - FTS and SRM can be done without
  - See Derek's talk for examples from CMS
- Data storage
  - New Internet flowers are still too fragile
  - For long time storage you need tape. Count long in number of accesses per year.
- Meta-data transfer/handling methods are not generic
  - must be developed
  - will probably need lots of thought and development
3. expected GridFTP performance: 50% of line capacity
7. File transfer service needed: yes. Suggest Phedex if still alive
9. Small files in the grid: don’t do that. Try to assemble, tar, block, zip
11. Intranet to internet traffic: open gridftp data port range for known sites via acl
13. Access to available data: setup gridftp rr DNS
14. see 7
17. Is LFC sufficient: not sure, but look at AMGA
18. naming scheme: yes, although purists hate this
19. Tape needed: archival of data on live disk is costly
Storage Requirements
what you should know before you write your proposal or attach to the grid

- High available (how high is high, well, the costs are high)
  - every 9 after 99. doubles the costs
  - maintenance windows possible
- High reliable (again how high is high)
  - can you sustain a reboot now and then
  - should be taken care of via software (failover, round robin dns etc)
- Persistancy how long to keep the data

- High data rates (again .. ist getting boring)
  - from WNs to storage
  - from storage to archive
- Interface to the storage
  - API
  - use open and accepted standards
  - compatible to existing Grid storage (e.g. Glite)
- X.509 end to end security and VO Access control
- access pattern to and from WNs and repository
  - size of files
  - size of reads and writes
  - proportion read to writes
Some Hardware Globals

- For large filesystems (>0.5TB) forget EXT use:
  - XFS: comes with SL5
  - ZFS: Solaris only (would the grid exist without Linux?)
  - GPFS: rocks
  - Lustre: rocks mostly

- Watch out for silent data corruption
  - use checksums

- Use RAID6
  - reduce rebuild risk
  - reduce time to failure
  - reduce silent data corruption
  - waiting for T10 DIF (SCSI Data Integrity Field)
Storage costs

- high speed disk storage
  - TB size
  - disk units of 400 GB (SAS)
  - intra-cluster
  - Infiniband or SAN
  - 1000 – 2000 €/TB + 1200 €/TB
  - power/cooling (0.15 €/kWh)

- bulk disk storage
  - PB size
  - disk units of 1 TB (SATA)
  - inter-cluster
  - Ethernet
  - 600 – 1500 €/TB + 600 €/TB
  - power/cooling

- tape storage
  - PB ( - EB) size
  - cartridges of 1 TB
  - Ethernet or SAN
  - 60 – 120 €/TB + 1 €/TB power/cooling
  - costs depend on
  - size of the procurement
  - startup or established vendor
  - maintenance model

- cloud storage
  - GB size
  - wide area
  - 1000 – 3000 €/TB
  - (http://www.aw20.co.uk/storagecosts.cfm)

Read my lips: 1kB disk space is 1000 and not 1024 bytes