LAL PROOF Cluster Update

Michel Jouvin
LAL, Orsay
jouvin@lal.in2p3.fr
http://grif.fr

LCG France, Lyon
22 Novembre 2010

Outline

- Reminders about context
  - LAL context
  - Reminder about needs and choices
  - LAL PROOT configuration
- Atlas end-user analysis specifities
- Results and problems
- Future directions
- Conclusions

Context

- LAL/Atlas request for an efficient local analysis facility
  - Atlas is the largest physics group at LAL: ~40 people
  - Concerns about competition to access a national facility
- GRIF operates a very large disk-based storage
  - ~1.5 PB mostly in 4 locations (over 6)
    - ½ dedicated to Atlas
    - LAL: ~25% of total GRIF storage
  - Full copy of AODs at GRIF
    - Mostly centrally managed
  - 10 Gb/s dedicated private network between GRIF sites
  - No backup but “easy” replication
- LAL internal non-grid storage ~50 TB
  - Atlas share: ~4 TB
  - Best-effort data management by users ~ no management…
  - Potential backup

Requirements...

- GRIF project includes T3 resources for local physicits
  - End-user analysis uses interactive tools
    - Short execution time: a few minutes
    - Grid/batch not appropriate: scheduling time vs. execution time
    - LAL grid CE very loaded
  - (partial) read of many files, few computation
    - Typically plot of an histogram of selected data
- Data: avoid duplication of what is already in grid side of the computing room...
  - Large data: transfer time vs. processing time
  - Data management sustanaibility on the long term
    - Requires tools... which already exists in the grid world
  - “Cheap” local file systems (e.g. NFS) have a limited scalability: may require a new storage infrastructure
    - HW + management cost

… Requirements

- ATLAS end-user analysis is ROOT-based
  - ROOT limitation: can use only 1 core
  - Multi-core usage/benefit requires PROOF
    - Computing cluster with 1 master and several workers
    - Each core is a worker
- PROOF has several constraints/requirements
  - PROOF implemented as an Xrootd plugin: data access protocol restricted to Xroot or Posix
  - PROOF driven by data: user analysis code **must** be supplied as a TSelector applied to input data

LAL Configuration

- Current PROOF cluster (workers) made of ATLAS non-grid/interactive machines at LAL
  - All LAL group servers are configured as gLite UI
  - PROOF master: 1 core on PROOF-dedicated machine
    - PROOF-lite « forbidden »... but difficult to enforce currently
  - Some machines partially dedicated to PROOF
    - Core subset configured as PROOF workers
  - Current configuration: 20 cores
    - To be extended soon, up to 100 cores
- PROOF Xroot-enabled storage: GRIF/LAL SE
  - DPM 1.7.4 + DPM/Xrootd plugin
    - Standard Xrootd on diskservers + plugin for namespace interface on DPM head node
  - Anonymous read: token-based auth configured but no Atlas token exists
    - No write through Xroot

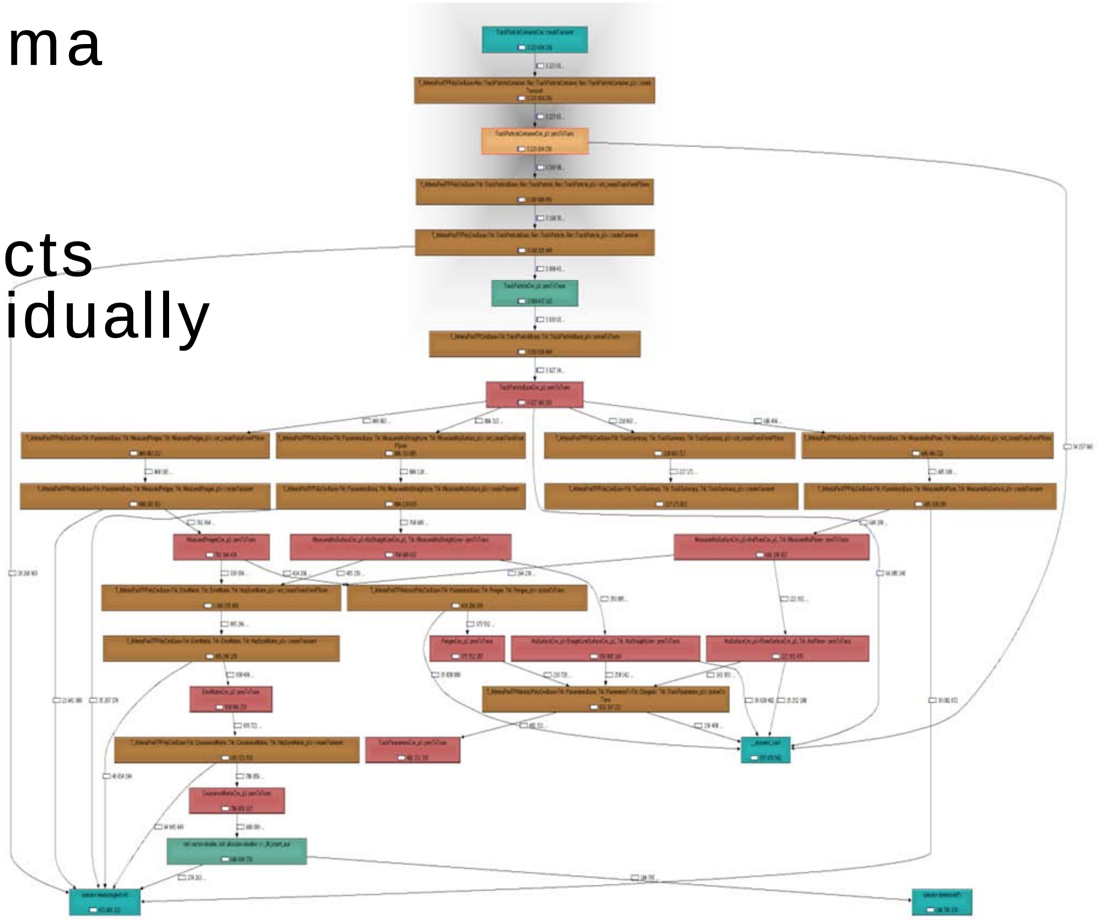
Atlas Analysis Specificities

- Early tests (Winter 2010): user complaints about bad performances when reading ROOT files
  - 5 MB/s... mais 100% CPU (RFIO ou Xroot): CPU bounded
    - No significan activity on disk server
    - Disk server : 10 Gb/s, UI: 1 Gb/s
  - Same performance with ROOT run on the disk server
  - 110 MB/s on interactive machine if file copied with rfcp
- Troubleshooting and optimization started by I. Vukotic
  - Work presented at CHEP 2010: PS20-5-090
  - Significant improvement since Marseille by internal reorganization/optimization of ROOT files
    - ROOT file structure can have a huge impact on read perfs
  - On-the-fly decompression doesn’t explain the bad perfs: mostly related to conversion from file structure to internal structure: generic + Atlas-specific converters
    - Work in progress in ROOT to optimize (2x) generic converters

Object Conversion

Borrowed from Ilija Vukotic’s presentation at CHEP 2010

- Transient objects are converted to persistent ones.
- To store it efficiently data from each sub-detector or algorithm passes a different (sometimes very complex) set of transformations.
- Converters of complex objects call converters for its members.
- It provides possibility for schema evolution
- Example: TracksCollection is composed of 20 different objects which can and do evolve individually



Ilija's Conclusions

Borrowed from Ilija Vukotic's presentation at CHEP 2010

- Data volume makes efficient reading of data extremely important
- Many possible ways and parameters to optimize data for faster input
- Different formats and use cases with sometimes conflicting requirements makes optimization more difficult
- Currently used file reordering significantly decreased job duration and stress on the disk systems
  - Will move to root optimized files
- DPM, Lustre, dCache
  - Need careful job specific tuning to reach optimal performance
  - Need a lot of improvements in order to efficiently support large scale IO required by analysis jobs

Results

- ROOT team/support very reactive and interested
- PROOF configuration integrated into standard Quattor QWG templates
  - Easy to setup a PROOF cluster with dedicated or non-dedicated resources
  - Very easy to add/remove workers (cores)
- PROOF stability is not perfect…
  - Mostly hidden from users with a cron job restarting Xrootd daemon if necessary
- Read performance problem in Atlas is dominated by ROOT-file internal structure and persistent/transient object conversion
  - HW cannot really help…
  - Work in progress
  - ROOT TTree cache has a limited (but non zero!) impact
    - Should help with concurrency as it improves load on disk servers

Open Issues

- DPM/Xrootd bad open performance
  - Consequence of asynchronous call to DPM namespace
    - Need to wait before checking status of open() but Xrootd timer in an integer number of seconds... thus 1 open = 1s
  - DPM will support synchronous call to the namespace at the end of 2010
    - Also required for NFS 4.1 support which is high priority on DPM roadmap
- PROOF sub-optimal packetizing phase increases impact of open time when processing a large number of files
  - During packetization phase (dataset verification phase), PROOF master has to open each file **twice**
  - Each file opened sequentially (one after the other) instead of // open of all files
    - To be improved in a next release
    - May have a dramatic effect on performance with 100s or 1000s of files

Future Directions (LAL)

- Convince users to look at PROOF
  - Most Atlas users have no PROOF knowledge and reluntant to “loose” time to look at it…
  - But this is the only way for efficient Atlas analysis
  - Next step: babysit 3 advance users with existing apps based on TSelector
    - Increase PROOF know-how at LAL
    - Better understand limitations of current configuration
- Assess PROOF/DPM performance improvements for large analysis
  - Current real use cases with 10-50 files
  - Working on 1000s files to prepare for the future
- Benchmark PROOF scalability
  - Optimal number of cores per user: overhead in split merge phase and master/worker communications
  - ROOT working on multi-core support without PROOF…

Future Directions (GRIF)

- Replicate LAL configuration at other GRIF sites
  - Other (Atlas) GRIF sites with the same needs
  - Some tests with a dedicated storage but storage duplication seen as a cost+manpower issue
- Assess efficient access to all GRIF data from any GRIF PROOF cluster
  - Should not be a problem as long as anlysis is CPU-bounded
- Extension to other VOs
  - Mainly ALICE: IRFU + IPNO
    - IPNO currently runs an ALICE-dedicated pure Xrootd server
  - CMS PROOF usage unclear
  - LHCb not interested: analysis tools based on Gaudi instead of ROOT
    - Gaudi has multi-core capabilities without PROOF

Conclusions

- LAL committed to provide an efficient analysis facility to local physicists with an optimized operational cost
  - Avoiding data duplication seen as the key issue
- PROOF is required in Atlas context for an efficient analysis
  - Due to high CPU usage for persistent/transient object conversion
  - May benefit from ROOT and Atlas optimization work
    - File structure and converter implementations
- DPM as a PROOF storage backend is functiunal despite an open performance issue with the current version
  - Improvement expected from next DPM and PROOF version in the coming months
- Setup/management of a PROOF cluster is easy with Quattor QWG templates
  - Should encourage sites to start one: no need for dedicated resources

Useful Links

- Caching strategy in ROOT and future evolutions (R. Brun, WLCG DAM Jamboree, Amsterdam 6/2010)
  - http://indico.cern.ch/getFile.py/access?contribId=22&sessionId=1&resId=1&materialId=slides&confId=92416
- Optimization and Performance Measurements of ROOT-based Data Formats in the ATLAS Experiment (I. Vukotic, CHEP 2010, Taipe)
  - http://117.103.105.177/MaKaC/materialDisplay.py?contribId=129&sessionId=42&materialId=slides&confId=3