Collecting Client-Side Monitoring Metrics

Dirk Duellmann, CERN IT

Creating Federated Data Stores for the LHC,
22. Nov 11, Lyon

Open Questions

- Quantify effectiveness of data placement and multi-tier storage proposals
- When does download to WN pay off?
  - Core / IOops ratio is increasing (maybe throttled by available memory) - model must break down at some time.
  - does the failure rate of long standing connections still require the download with current protocols?
- Will caching at higher tiers pay off with the current / future job mix?
- Will a partial cache do better than a full-file access?
- Do we need data product (intra-file) popularity?

Current Tools

- Tools currently used for storage optimisation:
  - HammerCloud
    - (+) large scale repeatable tests with full experiment chain and result aggregation
    - (-) not easy to match real job access pattern
  - “Rene’s scripts”
    - (+) repeatable single client behaviour
    - (+) can be tuned easily from full data access to sparse read use cases
    - (-) which mix of the very different access patterns should we look at / optimize for?
- “Convenient” benchmarks have their use but can easily mislead us if they replace reality
  - Brian: rather define a configurable set of micro-benchmarks, which can be reweighed to match different VOs or different versions of the s/w of one VO
- Need to measure (continuously) the real storage access profile and adjust our load generators

How to profile real job mix?

- Several layers can provide useful information
  - Experiment framework (eg data products used)
  - ROOT I/O summaries (eg cache eff, sparseness)
  - SE specific protocol summaries
  - System level
- ROOT I/O summary looks like a good starting point for standard diagnostic metrics
  - sparseness (read/volume)
  - randomness (seek distance/volume)
  - effectiveness
    - of the in-process (TTreeCache) cache
    - async read-ahead, stored TTreeBlocks
  - stability (connection retrials)
- Add experiment info and target metrics
  - job type, event rate, data products used

How to obtain and aggregate this data?

- Past attempts to collect a data access profile from ROOT in Alice have been done
  - information is largely available in ROOT
  - the collection for all jobs has failed due to high data amount and hence load on aggregation server
- We do not need all information from all jobs
  - a small random subset obtained from eg every 100 to 1000th job would
    - be sufficient to obtain min/max/avrg info for the full population
    - be feasible to aggregate
      - eg via an extended job wrapper and ActiveMQ based infrastructure which is already in place for CPU related collection
      - or experiment production systems
- Propose to define a compact access profile record and collect and analyse the data eg monthly