Panda
Notes for Discussion

Torre Wenaus

BNL

ADC Retreat, Napoli

Feb 3, 2011

DB Performance/Scalability - Archive

- Major (biggest?) current issue
- Major performance problems in archival DB, have had to shut off functionality (restored after tuning if possible)
- Trajectory to resolving problems and restoring performance and scalability not really visible
- Working closely with DBAs but we take a clear message – look at other potential solutions also
- Cf. Vincent's talk yesterday – noSQL
- Currently looking at Cassandra, Google BigTable (no), Amazon SimpleDB
- Job data for 2010 extracted to Amazon S3 csv, file table in progress
- Send data to noSQL prototypes in parallel to Oracle while we evaluate; continue to work with DBAs. Not running away from Oracle

DB Performance – Live DB

- Live DB (in-progress jobs and 3 day archive) less of a critical issue but a serious scalability concern also – deadly if we do hit serious problems
- Look at expanded use of memcached, as live job state repository, with periodic (~1Hz?) flushes to Oracle
- Panda server already has memcached service across the servers – can meet stateless requirement without relying solely on Oracle, preserving horizontal scalability
- Possibility of no Panda downtime during Oracle downtime
- Has to be tested and evaluated in a testbed
- An idea – not implemented

Scale via multiple server/DB instances?

- Consider revisiting >1 Panda server/DB, eg. at a Tier 1?
- Please no
- Been down that road, don’t go there
- No scalability gain, no real availability gain (if CERN dies we die anyway), big maintenance load
- Not necessary – better to scale central services horizontally (and solve the DB issue thoroughly)

Data access/scaling (apart from PD2P)

- Event picking service – now extended to support tape retrieval of files with events on a selection list
- For sparse selections. What about processing that hits all or most files in a tape resident sample? And maybe also for sparse selections if they become very popular?
- Data carousel… sliding window… freight train scheme?
    - Cycle through tape data, processing all queued jobs requiring currently staged data (next slide)
- Panda support for WAN direct access
    - Adapt memcached file catalog (*) as ‘cache memory’ for rebrokerage to maximize cache reuse
    - (*) implemented for WN-level file brokerage but unused so far…?

Efficient Tape-Resident Data Access

- If we need something like a data carousel…?
  - Need a ‘carousel engine’: special PandaMover-like? job queue regulating tape staging for efficient data matching to jobs?
  - Brokerage must be globally aware of all jobs hitting tape to aggregate those using staged data
    - But (production) jobs split between proddb and PandaDB – Bamboo can handle it but is this a motivator for a merge? (next slide)
  - Continue/complete the move of group production to the production system!
    - Essential in order to process any tape-resident group production in an efficient way
  - Is our task-to-job translation optimal? Rather than pre-defined jobs are there advantages to dynamically defining jobs in an automated way based on efficiency considerations like cached data availability?
    - Would this tie in with analysis interest in a higher-level task organization of user jobs?

ProdDB/PandaDB

- If it ain’t broke, don’t fix it. If it ain’t well motivated, don’t do it.
- Are some good motivations appearing, especially looking towards a long shutdown?
- The ‘global job view’ issue of carousel schemes
- DB scale concerns – clean up redundant info
- If we move away from utter dependence on Oracle, utter dependence still in ProdDB would be a tether
- Any DB reengineering eg. to introduce some noSQL elements should involve a design optimization cycle – could make sense to include ProdDB/PandaDB rationalization

Analysis Issues

- Rebrokerage complements PD2P – good to reduce time to rebrokerage and really try it! Monitoring a few days away
- Panda/DaTRI mediated merging of small files (temporarily) for replication – where are we with this?
    - Doesn't address small files on storage (files are unmerged on arrival) – do we need to?
- Use dispatch datasets for analysis dataset movement for greater efficiency? Containers makes it practical

Panda in the Cloud

- Very grateful to CERNVM folks for jumpstarting this – implemented ATLAS evgen job in a CERNVM-resident production pilot managed by their CoPilot, with Panda-side work from Paul and Tadashi
- Then a quiet spell until Yushu Yao (LBNL) took up Panda@Magellan (DOE's R&D cloud)
  - EC2-compatible interface (Eucalyptus)
  - He's leaving for a promotion, sadly (for us!) Looking for replacement
- Many possible approaches, everything on the table (next slide)
- Need dedicated stable effort, if you believe like me it's important!
  - Learn for ourselves the utility, performance, cost
  - Leverage free/available cloud resources

Panda in the Cloud Issues

- Some of the things on the table…
- Use Condor to submit pilots? Condor supports EC2 well. Same pilot factory as now.
- Or, use a VM-resident pilot daemon? Requires some pilot adaptation, but should be minor
- Use CERNVM's CoPilot infrastructure? eg. Authorization management
- What infrastructure will manage VM creation, lifetime based on queue content?
- VM image generic or preloaded w/some workload specifics?
- How to handle the storage? SRM in the cloud? (eck.) http! (rah.)
- WAN direct access particularly interesting here? Minimize cloud storage costs

Performance/Behavior Studies

- Active program in mining job history (with some Oracle struggle!) for studies of system performance and user behavior, e.g.
- Popularity study based on user counts – see Sergey's posted slides
- Job splitting – are users over-splitting jobs and saddling themselves with inefficient throughput and long latency?
- Much more we can learn from archival data
  - Performance differentials among workload types, sites, data access variants
  - User behavior to guide system optimization, user education, beneficial new features, …
- Depends on, and motivates, a robust high performance solution to the archival DB

Security

- A lesson long since learned: we can't believe anything anyone tells us on glexec deployment schedules!
- Will it ever be not only broadly deployed and perform sufficiently robustly and transparently that it'll be left on?
- What priority should we place on security measures in Panda to plug the security holes that using glexec would open?
  - Limited-lifetime token exchanged between Panda server/factory/pilot to secure against fake pilots stealing the user proxies that have to be provided to pilots to support glexec
- Good security coverage on the web interface side with new Panda monitor manpower
  - Fast response to incident last fall
  - So far our only (known) hackers are our friends ;-)

Various

- LFC consolidation – move output file registration to Panda server from pilot, discussed yesterday
- dq2 based pilot data mover – discussed yesterday. Eager to have it!
- ActiveMQ – follow the DDMers in this? Improvements over http esp. for high-rate messaging? eg. file-level callbacks
- Next-generation job recovery system in progress
- Adequacy of pilot testing tools? Use hammercloud for all?
- Panda @ off-grid Tier 3s: supported (adequately?) but no usage or great apparent interest – is there demand?