Event Service status and news

Ale Di Girolamo, Wen Guan,
Vakho Tsulaia and Torre Wenaus
For the Event Service team

US ATLAS Facilities Meeting
UCSD, March 6, 2017

ATLAS Event Service (AES) today

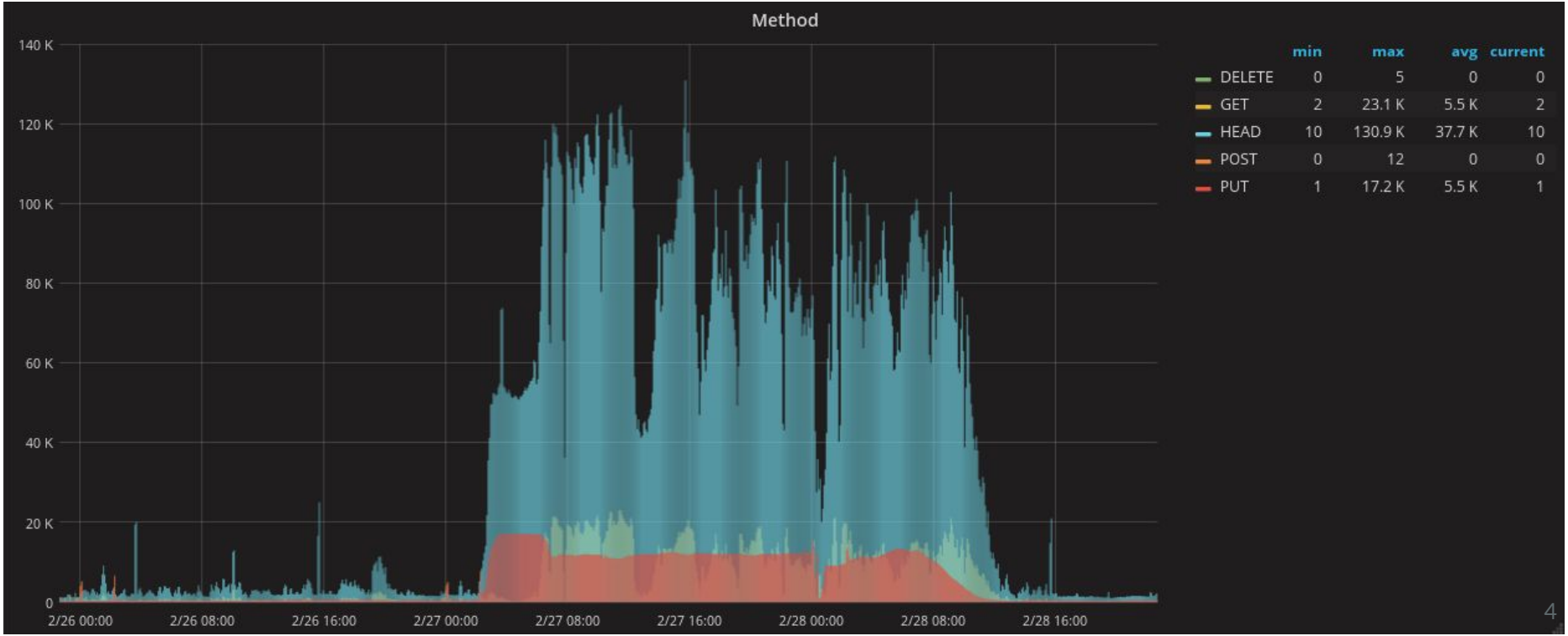
- We have been running G4 Simulation with Athena releases 19.2.X.Y and 20.3.X.Y. Starting to run with 21.0.X as well
- Event Service is being commissioned on the Grid
  - Few months ago we decided to focus on Grid resources to make the ES ROCK SOLID
  - This will allow us to better disentangle many small details that need to be fixed/polished
- ES (Yoda) on HPC in 2017
  - Has been running on SuperMUC
  - Has **not** been running at NERSC
    - Ongoing efforts: **NERSC-as-a-Grid-Site, Harvester commissioning**
- Accounting OK in terms of overall reporting of wall-clock/cpu time
  - To be checked the proper tagging of jobs
    - e.g. we saw reported some reco jobs which were simul - to be understood

Why Event Service on the Grid?

- The Grid presents opportunistic opportunities that could otherwise fall on the floor
  - e.g. draining prior to a shutdown
- When for some reason a site has no traditional work to process, give it AES work
  - **Never tell a pilot there's no work to do, utilize all the slots we harvest!**
- If we do more sophisticated management of how cores are allocated/used at the pilot level -- multipilots -- AES is the means to make this efficient
  - When cores complete their 'main' work, allocate them AES simu work so that all cores are utilized for the full slot lifetime
- Also: the Grid has real opportunistic, preemptable resources
  - e.g. preemptable opportunistic queues on non-ATLAS OSG sites
- Last but not least: we “know” the Grid much better than e.g. HPC:
  - Making the AES rock solid on the Grid should be “simpler” than on HPC -- it’s the first step

Object Store access

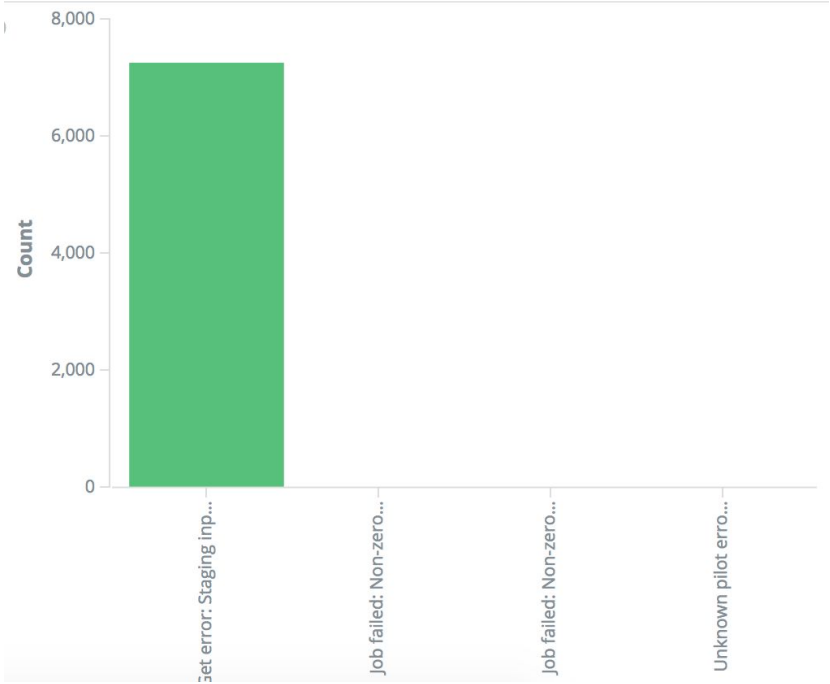
CERN OS Monitor (Dan Van Der Ster)



Object Store access (contd.)

- The OS access was dominated by **HEAD** operations
  - Coming from `get_bucket`, `get_key`, `get_metadata` and `set_metadata`
    - `get_bucket` and `get_key` can be improved
    - `get_metadata` and `set_metadata` **can be removed** (OS is a temporary storage, no need to have checksums)
- These optimizations have been implemented in the updated pilot **s3objectstore** site mover
- The new mover has been successfully tested locally
- The code has been merged with the master pilot branch
- Expected to be released soon (this week)

Failures of merge jobs



Number of failed merge jobs

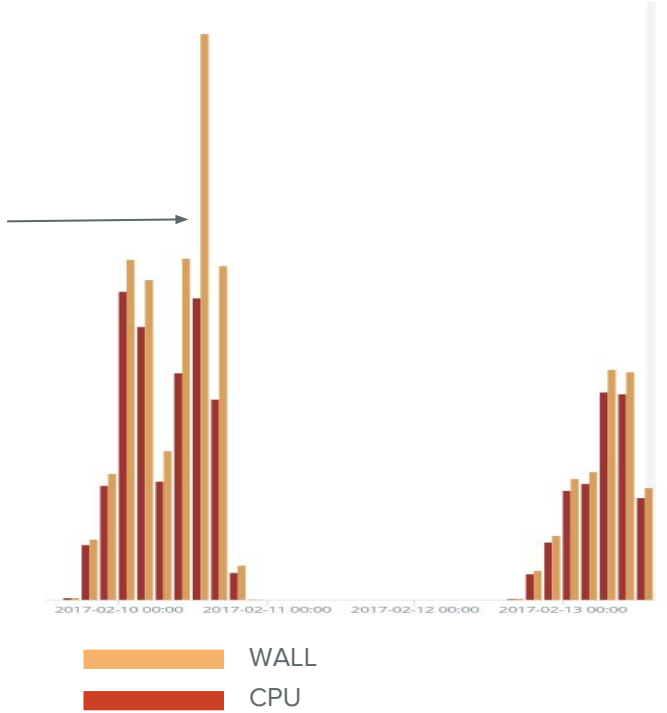
- >99.9% of merge jobs fail because of **stage-in problems**
- How OS is used by merge jobs:
  - Pilot gets an ES Merge job from PanDA
  - Pilot starts staging in files from the OS (one at a time)
  - **Failed stage-in gets retried 3 times. If failure persists, pilot fails the entire job and exits**

Failures of merge jobs (contd.)

- There are other reasons for failed merge jobs ...
  - E.g. **corrupted files** stored in the OS
- … but such cases **are extremely rare**
- While we should certainly come up with a smarter way of dealing with merge job failures (than the one described on the previous slide) ...
- … this is not a top priority task for the time being
- **The top priority task for the moment is to understand why we are seeing so many stage-in problems**
  - **And we are asking OS experts to help us out with this!**

CPU efficiency of the ES

- Example: recent Event Service task at CERN
  - **CPU time/WALL time = 76%**
  - Some of the jobs show very bad performance because of OS problems
- If no issues were encountered, the CPU efficiency can go **above 90%**



Handling of fine-grained outputs

- Event Service payload (AthenaMP) produces new output file for each event range
  - For G4 Sim: 1 event range = 1 event, thus the event range output is a 1-2MB HITS file
  - Results in large number of uploads and can saturate OS
- Alternative approach: reduce the number of transfers by tar-ing the outputs before uploading them to the OS
  - Initial implementation: tar all events into one tarball at the end of the job
  - **New approach**:
    - Periodically tar available events into one tar file, upload tar file and update event status
    - Trigger the tar-ing after regular time intervals (e.g. 10 min, can be made longer for single-core jobs)
    - Successfully tested for the ES and ES merge jobs

Evicted vs failed

- Evictions are expected when Event Service jobs run on opportunistic resources
- Currently monitoring shows evictions as failed jobs with lost heartbeats.
  ---> *Q: How can we distinguish normal evictions from real problems?*
- We want to understand if it's possible for the batch system, before evicting a job, to send to the pilot a signal different from SIGTERM
  - Yes: We can distinguish evicted jobs from crashes
  - No: We cannot do that
- Similar issue: we cannot distinguish the cases of killed jobs because of excessive memory usage
  - For the time being we don't know how to address this…
  - Pilot could send messages to PanDA server in case of excessive resource usage (to be discussed…)

Technical Validation

- We don't really need full Physics Validation for Event Service, but we need at least Technical Validation

- First validation task of the ES (20.3.7.5) finished
  - http://bigpanda.cern.ch/task/10611351/?mode=nodrop
  - Results are here: https://test-jgarcian.web.cern.ch/test-jgarcian/TechVal/NewES/
  - Jose: "*... the results are not too bad considering that the statistics between reference and test (ES) is not the same since some jobs failed.* "

- New validation task submitted together with a regular Grid task for reference
  - MC16a with rel 21.0.15
  - Event Service running on sites: BNL_LOCAL and CERN-P1

Running at full speed

- It has been noted that currently we don't have a continuous stream of the Event Service jobs. Event Service activity is spotty, which makes it non-optimal for some resources
- The ways of addressing this situation are being discussed within the Event Service team
  - We will have a pot of MC16 tasks which can run ES
  - We start with few sites, and then gradually involve more sites
  - This assumes that the **validation is done promptly and the validation results are positive**
- We want to make sure that each production software release can run Event Service jobs
  - This may not be true for some existing (old) production caches

Event Service & US ATLAS Facilities

- Do you (sites) have opportunistic/preemptable queue?
  - How many cores? What types (SCORE, MCORE)? How much memory/core?
  - We want a specific queue, because we **don't** want to mix for now "normal" resources with ES

- Do you want to participate in ad-hoc commissioning of … ?
  - Object Store interactions (from remote - far WN)
  - Monitoring: really difficult bit for us - Sites experts could have different wishes, would be nice to know them!
  - Accounting verification; Analytics studies; ...

If interested, please email to:
Alessandro.Di.Girolamo@cern.ch
wguan@cern.ch

Event Service: next steps

- Push for commissioning and validation on the Grid
- Resume Event Service operation on HPC
  - Switch to Harvester
  - Extend Event Service / Yoda to other HPC centers beyond NERSC (OLCF, ALCF)
- **Event Streaming Service**
  - V01 prototype already implemented but not yet tested
  - We'll have a focussed discussion on further developments during the S&C workshop next week
- Event Service beyond simulation
  - Derivation can be tried any time (once there is somebody available for doing this)
  - Should be relatively simple (at least from the payload perspective) to extend Event Service to those workflows that run transforms in **single step** (Fast Chain?)
- Implications/opportunities for the Event Service with AthenaMT
  - To be discussed **in Valencia**