

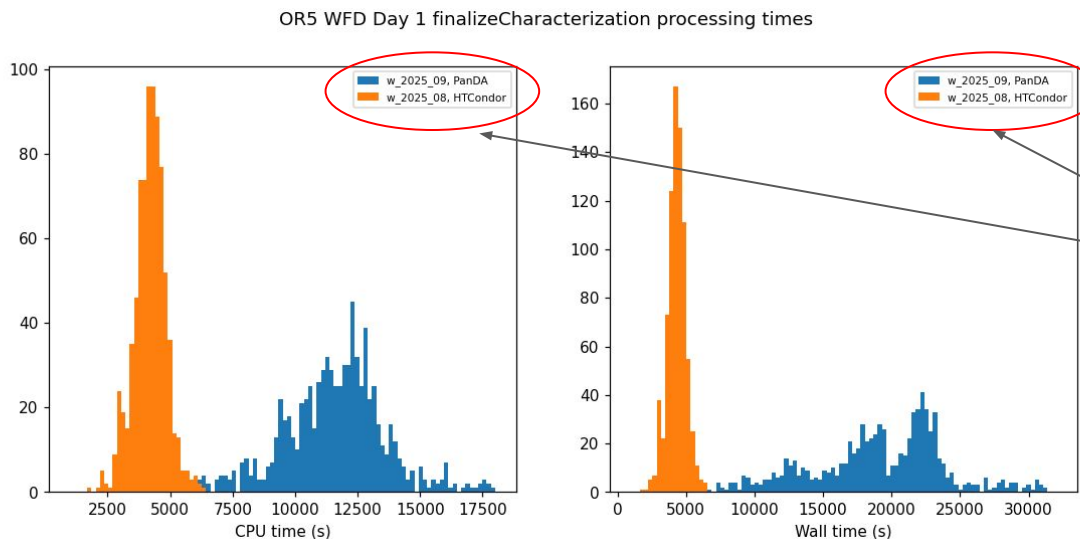
Investigation of Slowdown in RubiN OR5 testing

2025-04-10

Edward Karavakis, Wen Guan, Zhaoyu Yang on behalf of the PanDA team

OR5 test

- Rubin has a recent “Operations Rehearsal” testing (OR5). See Brian’s slides at last [PanDA Community Forum](#)
- In the OR5 tests, the same Rubin science workflows were tested with both HTCondor and PanDA.
- After the tests, an overall ~3x slowdown was reported to see in PanDA jobs.
 - Different Rubin software versions were used: **weekly 08** (HTCondor) vs **weekly 09** (PanDA)
 - Different datasets: **WFD** (HTCondor) vs **DDF** (PanDA)



The SW version and dataset matter a lot in the comparison (see next)

plot credit: Jim Chiang

Check on single job

- Firstly we examined the longest job from task *finalizeCharacterization*.
 - It finished in 1.5h in SLURM batch, but used 8.5h in the OR5 PanDA test.

```
dataId: "instrument='LSSTCam-imSim' AND skymap='lsst_cells_v1' AND visit=214468 AND band='i'"
task: finalizeCharacterization
```

```
PanDA runs 11380-11389, 11367, 11363, 11360, 11359
```

```
#w09 11380-11384
```

```
walltime = ["0:1:37:59", "0:1:29:08", "0:1:23:13", "0:1:36:51", "0:1:21:55"]
```

```
#w09+tickets 11385-11389
```

```
walltime = ["0:1:34:47", "0:1:29:01", "0:1:30:17", "0:1:27:52", "0:1:29:20"]
```

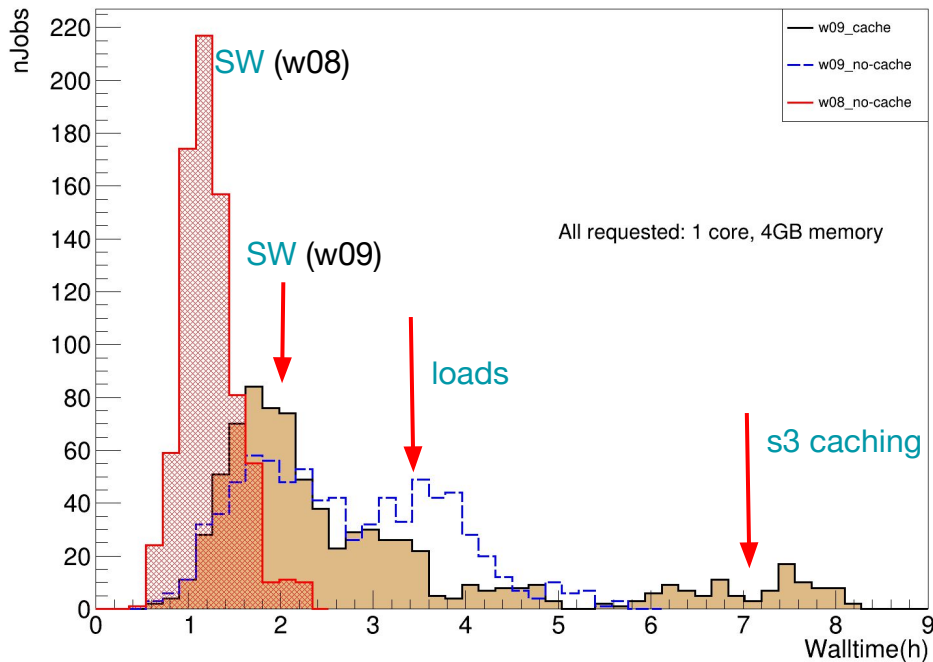
```
#w09+tickets&cache 11367, 11363, 11360, 11359
```

```
walltime = ["0:2:47:56", "0:2:52:49", "0:3:10:11", "0:3:15:07"]
```

- Observations for single job:
 - Using Rubin SW weekly w09, PanDA jobs have comparable walltimes to HTCondor (1.5h, 5 tests)
 - Using w09 and the ticket branches, PanDA jobs have comparable walltimes to HTCondor (1.5h, 5 tests)
 - The butler cache setting (more details later) doubles the job walltime (3h, 4 tests)
 - In all cases, we didn't reproduce the long 8.5h walltime
- caveat: OR5 HTCondor test in w09 used a different dataset (DDF). There are no tests that used the identical datasets and SW version in PanDA and HTCondor tests.

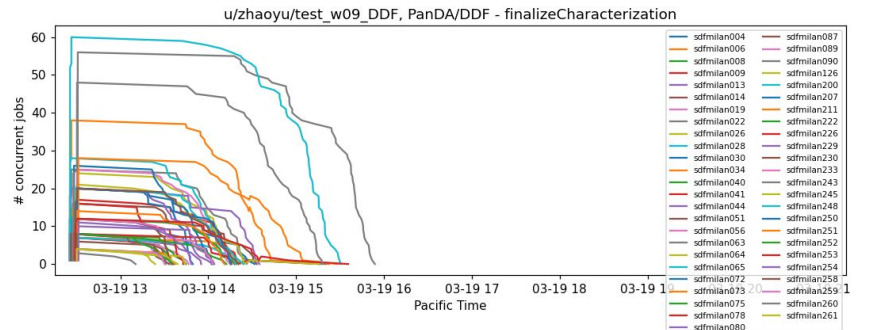
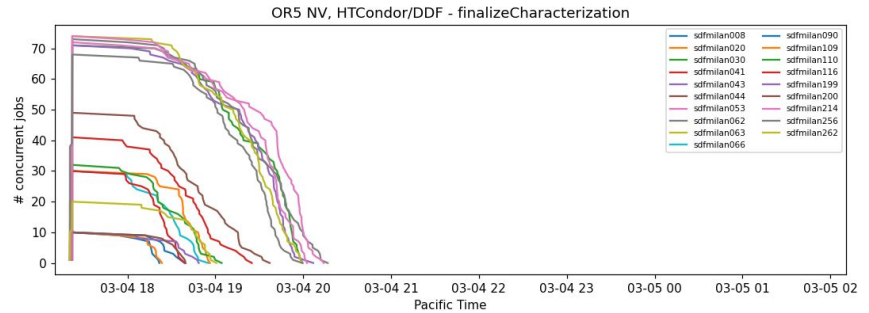
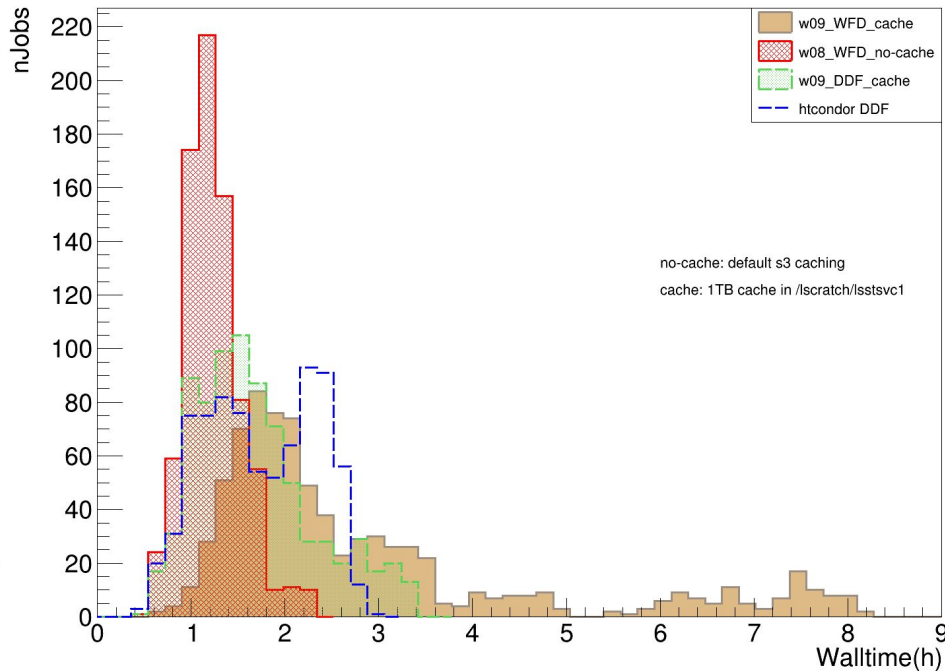
SW version

- Next we tested with 800 concurrent *finalizeCharacterization* jobs, which all finished within 2h in HTCondor, but has long tail in PanDA jobs all the way up to 8.5h.
 - When using the same SW **weekly 08** as HTCondor, and turning off the customized s3 caching, PanDA jobs have comparable walltime to the HTCondor.
 - The slowdown exists but not unique to a particular WMS. The walltime/cputime is mainly affected by the high loads, but s3 caching couples with it to form the long tail.



Loads

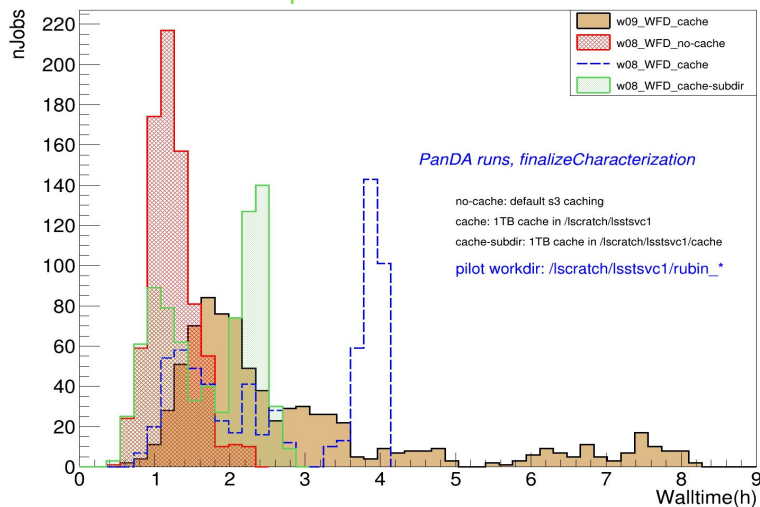
- At USDF, the batch system SLURM packs Rubin jobs in a fill-up mode:
 - Each node has 128 cores and 512 GB RAM. 8 cores are reserved for the storage management software.
 - SLURM packs Rubin jobs on a node as many as possible before moving to next one
 - For the 800 jobs(1 core, 4GB ram), quite often PanDA got only a few nodes with many having 120 payloads.
- When using the same SW version(**w09**) and same dataset (**DDF**), the loads impact PanDA and HTCondor similarly.



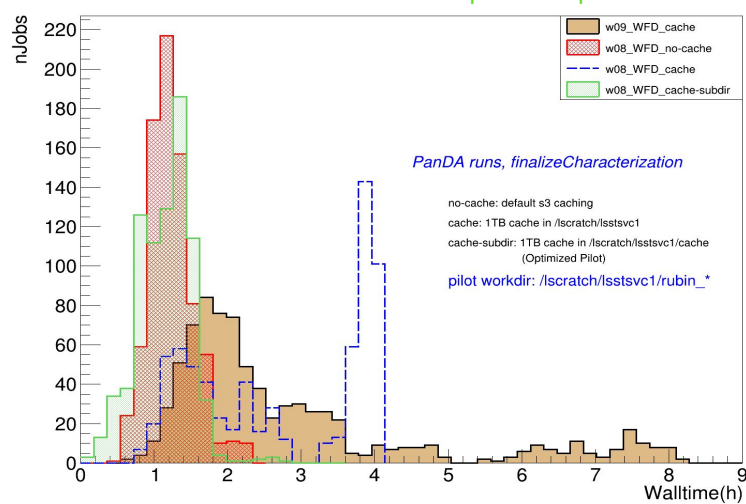
S3 caching

- Rubin data management is through the Data Butler that coordinates the Datastore and Registry
 - The s3 caching for the butler datastore is always used by default, but on the basis of each task with limited number of datasets. We've never seen such contrast of slowdown in Rubin's regular campaigns between PanDA and other batch jobs.
- In OR5 testing, it's the first time to use a shared s3 cache for all tasks/jobs. The cache area is set in \$TMPDIR, with a limit of 1TB.
- The s3 caching impacts PanDA more than HTCondor
 - PanDA pilot workdir is a subdirectory in \$TMPDIR
 - HTCondor workdir is on a different shared filesystem
 - There is a bug in the Rubin middleware, which left tens of thousands of empty directories in \$TMPDIR
 - A new test was done by setting the cache to a subdir in \$TMPDIR. It helps mitigate the caching effect.

production PanDA



dev PanDA with optimized pilot



Summary

- The investigation started with the relative slowdown of PanDA jobs in the Rubin OR5 tests.
 - This is mainly due to the comparison between different software versions and datasets.
 - When using the same SW version and same dataset, PanDA jobs have similar walltime/cputime as HTCondor if using the default s3 caching mode.
- Slowdown caused by high loads is a common issue in both WMS. Packing jobs close to capacity on a high-core node slows down jobs in PanDA and HTCondor similarly.
- The 1TB s3 caching in \$TMPDIR impacts PanDA jobs more than HTCondor. That's caused by too many empty folders left by the s3 caching in the parent directory of PanDA pilot workdir, which are not cleaned after the jobs finish.
- The thorough investigation helps us understand the loads and s3 caching impacts on job processing time. This can guide us to improve the job scheduling and monitoring in case of high loads on a node, and also how to use the s3 caching more properly to benefit from it other than hampered by it.