Research track 2: Data transfer and access

Sergey Panitkin

(BNL)

Outline

- Introduction
- Pilots IO on DTNs
- Containers on Titan (in progress)
- Summary and Plans

IO issues on DTNs

- In November we got a notice from OLCF storage support (Jesse Hanley) about large number of open/close operations originated from the DTNs.
  - Different issue from payload generated Luster MDS problem observed in June
- Main reason - access to pilot log file
- At that time we were running with ~15 pilots
- We have been running with the same pilot version on the DTNs since March 2017

IO issues on DTNs: Access to pilot logs

- 5 minute sample of opens on dtn{35,36,37} gave ~66k, 125k, and 128k opens respectively
- Examples of the number of open() operations to the pilotlog files (5 min sampling):

6328 $PROJDIR/pilots_workdir1/Panda_Pilot_90065_1511867945/pilotlog.txt
10379 $PROJDIR/pilots_workdir2/Panda_Pilot_50511_1511869667/pilotlog.txt
8251 $PROJDIR/pilots_workdir3/Panda_Pilot_118163_1511874836/pilotlog.txt

- Many log files are opened simultaneously, since many pilots are running
- Open is followed by close
- Pilots can run for more than 2 hours writing ~400k log messages

Log messages

- Single logfile analysis (‘module’, N messages)
  - Pilot modules sorted by verbosity

```
[('pUtil.py', 87531), ('Monitor.py', 77607), ('PandaServerC', 65850), ('ATLASExperim',
34825), ('SiteMover.py', 30445), ('FileState.py', 26441), ('RunJobUtilit', 23432),
('Mover.py', 18384), ('GFAL2SiteMov', 15480), ('RunJobTitan.', 12985), ('JobLog.py',
7810), ('UpdateHandle', 5597), ('FileHandling', 5345), ('SiteInformat', 3582),
('ATLASSiteInf', 3564), ('Job.py', 3150), ('WatchDog.py', 3130), ('SiteMoverFar', 2376),
('RunJob.py', 1748), ('ErrorDiagnos', 700), ('Node.py', 256), ('JobRecovery.', 244),
('processes.py', 32), ('normal', 14), ('Experiment.p', 5), ('DBReleaseHan', 2),
('PilotTCPServ', 1)]
```

- 27 modules 430536 messages (delimiter lines ignored)

Log Messages

- Top 20 messages (‘module | message’, count):

```
[('pUtil.py | getSiteInformation: got experiment=ATLAS', 7330), ('PandaServerC | Batch system type
was not identified (will not be reported)', 4244), ('pUtil.py | HTTP connect using server:
https://aipanda031.cern.ch:25443/server/panda', 4244), ('ATLASExperim | Memory summary
dictionary not yet available', 4244), ('pUtil.py | toServer: cmd = updateJob', 4244), ('PandaServerC |
pilotId: xtestP001', 4244), ('PandaServerC | Did not find any reported high priority errors', 4244),
('PandaServerC | Stdout tail will not be sent (debug=False)', 4244), ('PandaServerC | Will send pilotID:
xtestP001', 4244), ('PandaServerC | Checking if new site movers workflow is enabled:
use_newmover=False', 4244), ('PandaServerC | getXML called', 4244), ('pUtil.py | Executing command:
curl --silent --show-error --connect-timeout 100 --max-time 120 --insecure --compressed --capath
/lustre/atlas/proj-shared/csc108/app_dir/pilot/grid_env/external/grid-security/certificates/ --cert
/ccs/home/doleynik/x509up_u10469 --cacert /ccs/home/doleynik/x509up_u10469 --key
/ccs/home/doleynik/x509up_u10469 --config /lustre/atlas2/csc108/proj-
shared/pilots_workdir1/Panda_Pilot_90065_1511867945/curl.config
https://aipanda031.cern.ch:25443/server/panda/updateJob', 4244), ('pUtil.py | Sending attemptNr=1
for cmd=updateJob', 4244), ('ATLASExperim | File does not exist either: /lustre/atlas2/csc108/proj-
shared/sandboxes/dtn35_1_01a961d3-ff7b-4216-a54f-
f68f3f51081b/memory_monitor_summary.json', 4244), ('PandaServerC | jobDispatcher acknowledged
with 0', 4244), ('FileHandling | Pilot error report does not exist: /lustre/atlas2/csc108/proj-
shared/sandboxes/dtn35_1_01a961d3-ff7b-4216-a54f-f68f3f51081b/pilot_error_report.json (should
only exist if there actually was an error)', 4244), ('ATLASExperim | summary_dictionary={}', 4244),
("PandaServerC | ret = (0, {'command': 'NULL', 'StatusCode': '0'}, 'command=NULL&StatusCode=0')",
4243), ("pUtil.py | Dispatcher response: [('command', 'NULL'), ('StatusCode', '0')]", 4243),
("PandaServerC | data = {'command': 'NULL', 'StatusCode': '0'}", 4243)]
```

Fixing Pilot1 logs

- The reason for this was identified as Pilot1 logging implementation

```
def appendToLog(txt):
""" append txt to file """

    try:
        f = open(pilotlogFilename, 'a’)
        f.write(txt)
       f.close()
    except Exception, e:
```

- Straightforward fix: Reduce the number of log messages by hand
- After Danila implemented the fix 5 min total went from 320k to 93k opens
- Pilot 2 and Harvester do not have this problem since both use Python logging module

Containers in ATLAS

- ATLAS started testing containers on Grid in 2017
  - Docker, Singularity
  - Typically requires Centos 7 installed on a site for full Singularity support
  - Site Singularity configuration plays large role
- Containers for HPC were tested at NERSC with Shifter and Singularity
- Containers are viewed as software distribution tool for HPC machines without CVMFS
  - Container with full (deduplicated) CVMFS tree ~600GB
  - Single release container ~50GB

Containers on Titan

- Singularity container platform became available for tests on Titan in 2017
- Accessible on batch worker nodes and interactive worker nodes
- Currently v2.4.0 module is available
- Some documentation and scripts are available in github
- Singularity on Titan imposes several requirements on user container images
  - No run-time mount points, all file system bindings have to defined in the image. Run time bindings (-b fs1:fs2) are not supported, since CNL kernel does not support overlayfs. (Singularity on Summitdev supports this option)
  - Placeholder for Titan specific setup script in the image (to be invoked at run time)
  - Linux userIDs in the image should coincide with Titan's userIDs

Container build for Titan I

- Singularity installed from scratch on my laptop, since root privileges are needed for container image building
  - MacBook Pro 2016 laptop with VirtualBox, Vagrant VM with Singularity 2.4, following Singularity documentation
  - Manual install of Singularity v2.4.2 in Vagrant VM later on. A lot of bug fixes in this version.
- Images with CentOS 6, 7 as base OS, loaded at build time from Docker Hub
  - Tried several different kernels, did not see much difference
  - Added a few system libraries required by ATLAS software
- “Post”-stage script for Titan specific mount points (from Adam Simpson’s Github)
- ATLAS release 21.0.15 installed using Pavlo’s scripts from Github
  - Special handling for installation of ATLAS DBRelease fix for 21.0.15
  - Installed customized DBRelease configuration files for the container
  - Some extra rpms for common tools required for ATLAS release install scripts (git, perl, wget, …)
- Several users added with proper Titan userIDs

Container build for Titan II

- Image build time ~2 hours on MacBook Pro
  - Max system load during build ~40%
- Container file sizes
  - Image file on top of Ext3 filesystem ~29GB
  - SquashFS based image file ~7GB
    - Support for SquashFS was introduced recently in Singularity v2.4
    - SquashFS supports compression
  - Same ATLAS release installed on Titan ~27GB
- For comparison some containers build by Wei Yang (SLAC)
  - “Fat” ATLAS container ~600GB
    - Full ATLAS (deduplicated) cvmfs
  - Container with rel. 21.0.15 and DBRelease ~ 50GB

ATLAS container tests on Titan

- Ext3 and Squash containers were copied to Lustre and NFS on Titan
- Tested with ATLAS production job
  - Short jobs with 16 events
- Jobs submitted manually to batch queue, f.e.
  - aprun -n 1 -N 1 -d 15 -r1 singularity exec /ccs/proj/csc108/AtlasReleases/containers/my_centos_6_docker_Titan_DBRelease_with_gcc_v2.simg ./run.sh
  - Release setup done at run time via run.sh
  - Job working directory is on Lustre
  - Root Input file with events on NFS or Lustre
- Timing from Athena logs
- Tried several container placement options including RAMdisk

ATLAS container tests on Titan: Some results

Type	Location	Size, GB	Setup time, s	Run time, s	Job ID
Direct Release	NFS	26.7	357	1610	3801346
SquashFS	NFS	7.2	742	4272	3800895
Ext3	NFS	29	766	4029	3801075
SquashFS	Lustre	7.2	746	4157	3807410
Ext3	Lustre	29	773	4023	3807409
SquashFS	RAM disk	7.2	722	4124	3801346

Setup time: from the transformation start to event loop start
Run time: from the transformation start to exit

Some initial observations:

- Simulations in containers run ~x2 longer than the simulation ran from disk installed release
  - Is this related to access to the container file?
- No big difference between run times for containers placed on NFS or Lustre (NFS is optimized for read)
- No big difference between Ext3 and SquashFS based containers
- Container started from RAM disk on worker node runs the same
  - Indication that the slowdown is not IO related?!

Solving the slow containers puzzle

- After discussing test results with Adam Simpson (OLCF) and trying several other possibilities I looked at dl-intercept feature
- This feature is active by default on Titan and is used to intercept loading of MPI related shared libraries within containers
- In the current Singularity setup at OLCF dl-intercept is on but can be switched off after Singularity module is loaded
  - unset SINGULARITYENV_LD_AUDIT

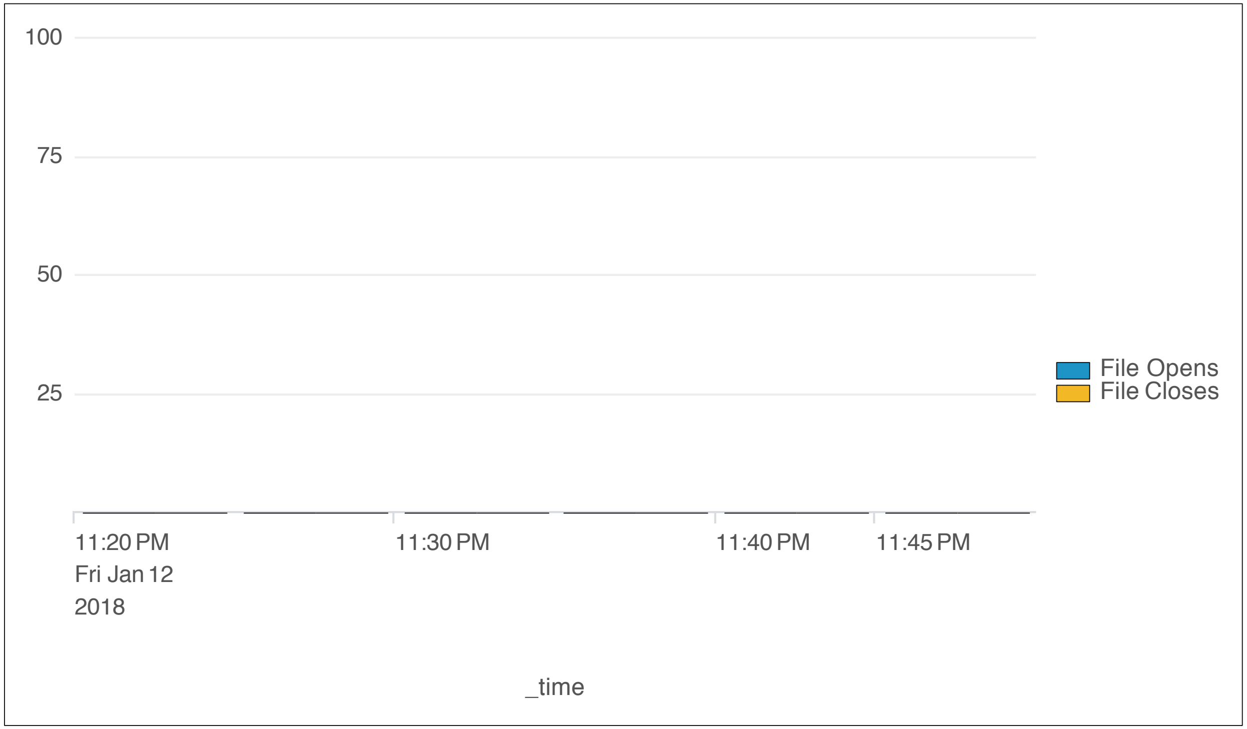
Running with DL AUDIT OFF

Type	Location	Size, GB	Setup time, s	Run time, s	Job ID
Direct Release	NFS	26.7	357	1610	3801346
SquashFS dl_audit ON	NFS	7.2	742	4272	3800895
SquashFS dl_audit OFF	NFS	7.2	221	1425	3822559
Ext3 dl_audit OFF	NFS	29	239	1491	3822317

- Simulations in containers run ~x3 faster when DL AUDIT is turned off
  - “unset SINGULARITY_DL_AUDIT”
- Containers run on par or even faster than release installed on NFS
- Not much difference in performance between SquashFS and Ext3 based containers
  - SquashFS based containers are much smaller (x4)

Container I/O. I

- Splunk profile for simulation in SquashFS container located on NFS
- Lustre file opens/closes

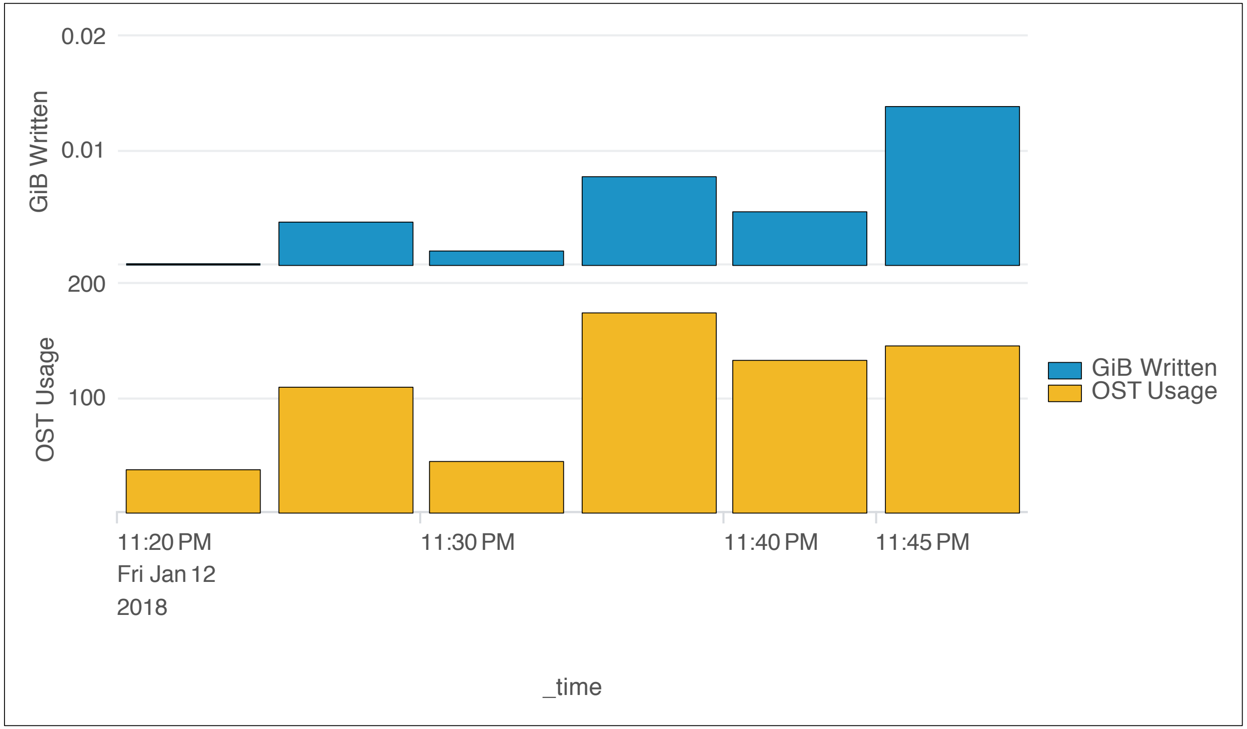


Job 3822559

Container I/O. II

- Splunk profile for simulation in SquashFS container located on NFS

Job Specific I/O Statistics: Write BW & OST Usage

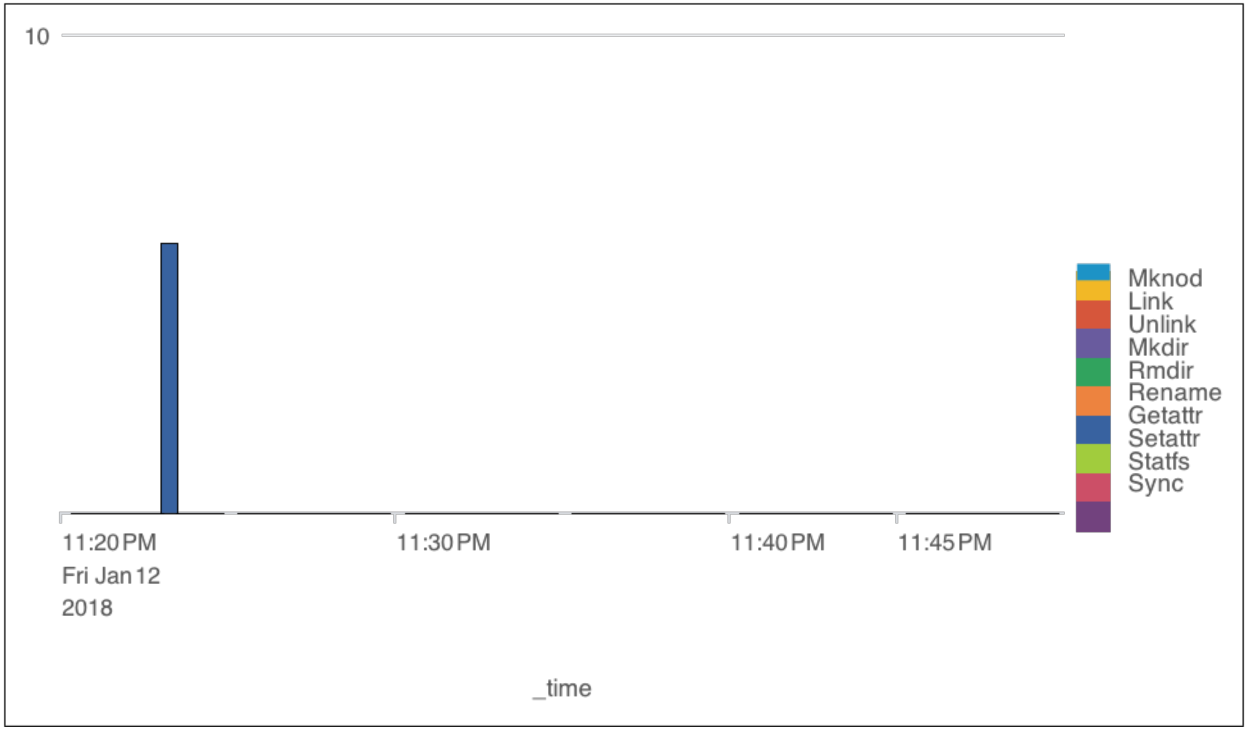


Job 3822559

Container I/O. III

- Splunk profile for simulation in SquashFS container located on NFS

Job Specific I/O Statistics: Other Metadata Operations



Job 3822559

Summary

- One issue with IO on DTNs was brought up in November
  - Large number of open/close for pilots log files
  - Analyzed and addressed
- Started work with Singularity containers for Titan
- ATLAS simulations in containers performed well (after the default Singularity option is turned off)
- Containers showed good IO properties with almost no load on Lustre MDS

Plans I

- Jan.- May. Containers on Titan for ATLAS
  - Scaling studies.
    - MPI wrapper for containers
    - Strong Scaling
  - Containers created by ATLAS (Wei Yang) at BNL
    - Need to be build and configured to reflect Titan specifics
    - Large size, hard to modify on a laptop
    - Started work with these, still do not work 100% on Titan, transformation crashes
      - Titan specifics, DB configuration
    - Hope to converge on working container (Jan?)
  - Work with Danila on using containers in ATLAS production on Titan
    - Integration with current Pilot setup (with Danila)
    - Containers with MPI wrapper
    - IO properties and timing
  - Containers integration with Harvester (with Danila and Pavlo)
- Containers with NGE (with Matteo)
    - ATLAS simulations are probably the easiest case

Plans II

- Mar. –June: Atlas payloads on Summit
  - Contingent on success of the ACSC project (ATLAS software compilation for PowerPC architecture). See talk by A. Undrus
  - ATLAS Geant simulations
    - Performance on Power9
    - Single node scaling (Power9 : 24 cores MT4, 2 CPU per node) – AthenaMT?
    - Multi-node scaling
  - IO issues (New storage hierarchy on Summit! 1.6TB NV memory per node, burst buffers,…)
    - Containers with MPI wrapper
    - IO properties and timing
  - Work with Danila on using containers in ATLAS production on Titan
    - Integration with Pilot
  - **Are there backfill opportunities on Summit?!**
  - Containers launched with Harvester (with Danila and Pavlo). Harvester on Summit.
  - Containers created by ATLAS (Wei Yang) at BNL.
  - Started work with these, still do not work on Titan
      - Titan specifics, DB configuration
    - Large size, hard to modify on a laptop
    - Need to be build and configured to Titan's specifications
  - GPU intensive applications
    - ML use cases. We have now ML people joint CSC108

- Feb. - May . Yoda on Titan
  - Software port, Configuration on Titan , running mode and test
  - Study IO properties