Using Containers on HPCs for ATLAS Experiment

W. Yang[1]*, D. Benjamin[2], T. Childers[3], D. Lesny[4], D. Oleynik[5], S. Panitkin[6], V. Tsulaia[7], X. Zhao[6]

on behalf of the ATLAS Collaboration

[1] SLAC National Accelerator Laboratory, [2] Duke University, [3] Argonne National Laboratory, [4] University of Illinois Urbana-Champaign, [5] University of Texas Arlington, [6] Brookhaven National Laboratory, [7] Lawrence Berkeley National Laboratory, *Corresponding Author

Solving the ATLAS Software Distribution and IO Load Problems on HPCs

HPCs are often different from the Grid

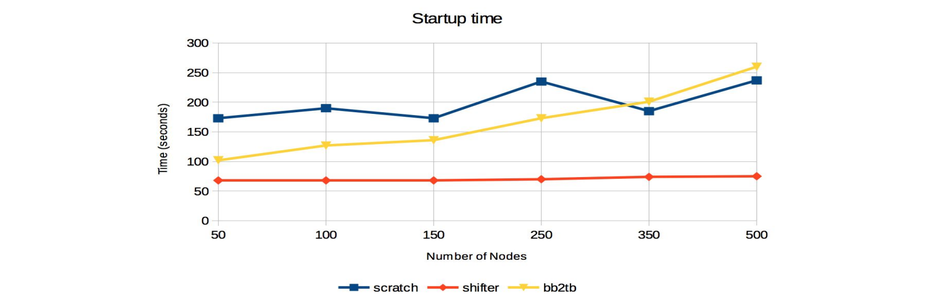
- Many do not have CVMFS
- TCP is not always available on batch nodes
- Many ATLAS jobs can start at the same time
  - HPCs are designed to run large parallel jobs
- Large high throughput shared file systems
  - But expensive and slow in file lookup & file locking (metadata IO)
- Put ATLAS software in shared file systems
  - **High labor costly** on installation and maintenance
  - **Does not scale well:** loading .py and .so by ATLAS jobs contribute to IO overload

ATLAS containers on Top500 HPCs (June 2018 List)
#7 Titan: Cray XK7
#10: Cori: Cray XC40
#21: Theta: Cray XC40
#22 MaroNostrum, Lenovo SD530

Container can solve these problems

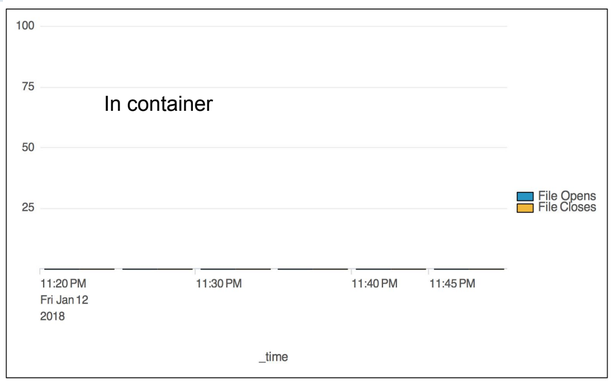
- Think of container as a read-only loopback file system
  - Container reside on the main shared filesystem
  - "Loopmount" at /cvmfs - no need to install software on HPC sites
  - Compute nodes do file lookup.
  - Shared file system only deliver data blocks.
  - Container is not quite a loopmount - but the effect is the same
- All inclusive container can be pretty big in size
  - ATLAS Grid runtime environment
  - Multiple ATLAS software releases, DB Releases, Generators, ...
  - O(100GB) or more
- Container size has nothing to do with loading speed
  - A 10TB hard drive will mount just as fast as a 100GB hard drive

NERSC Cori: Put ATLAS software in Shifter container, in Burst Buffer or on Lustre shared file system



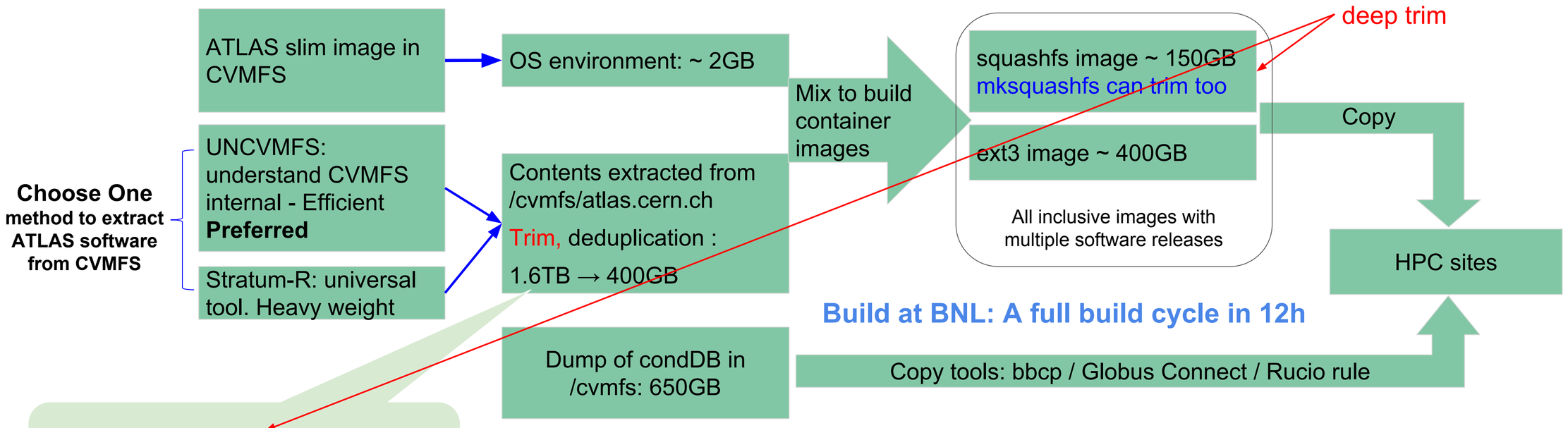
- The best scaling obtained with **Shifter**
- **BB** visibly outperforms **Lustre** for small number of nodes
- Very good scaling on **Cori Lustre** comparing to **Edison Lustre**

Typical Splunk profile for a single AthenaMP job on Titan



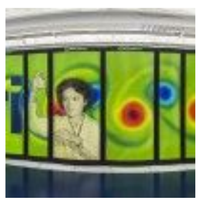
OLCF Titan: Use of containers with ATLAS software reduces load on metadata server on the Lustre shared file system. Single large file access instead of access to hundreds of small files.

Workflow of Creating All Inclusive Container Images for HPCs



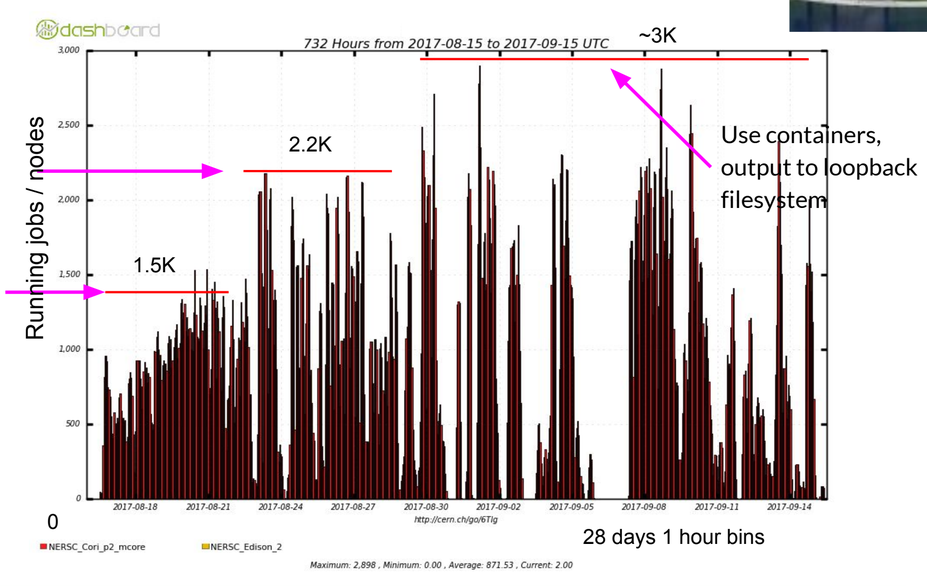
- Use rsync rules for fine grain trimming - both to reduce image size and to reduce IO
- An important step to build modifiable ext3 image, or deep trim image (e.g., single release images)
- First rsync took days to complete (small file IO dominated)
- future rsync make incremental changes - complete in a few hours

Container Enable ATLAS to Scale up at NERSC Cori 2

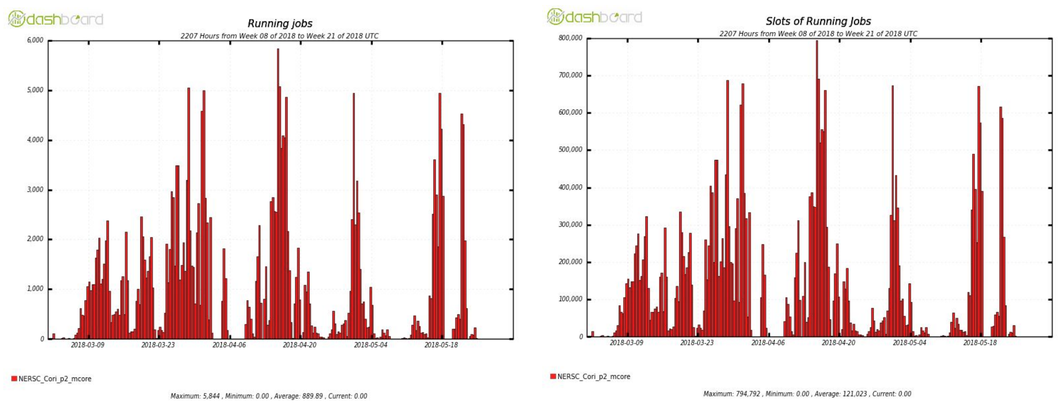


Use containers, output to Shared file system

Running in Shared File system only - at limit



Use containers, output to loopback filesystem



Each node run 136 simulation processes

1. using the Shared File System alone, no containers - beyond 1.5K nodes, job run time increase significantly.
2. using containers, output to Lustre shared file system
3. using containers, output to loopback file system

When packing twice as many simulation processes in one node …

- Run twice as many events in a node
  - A way to further improve CPU utilization due to many slow cores on Cori 2
- Container mitigates the increased metadata IO load

Summary

- All inclusive container has been used by ATLAS on HPCs
- Distribute ATLAS software to HPCs that do not support CVMFS.
- Scale up to 3000+ nodes on Cori 2
  - This is ⅓ of Cori 2
- Deployed an infrastructure for quick HPC container creation
  - Capable of filter out unneeded software and data in CVMFS

This work was supported through allocations on the US DOE Office of Science facilities at NERSC, ALCF and OLCF