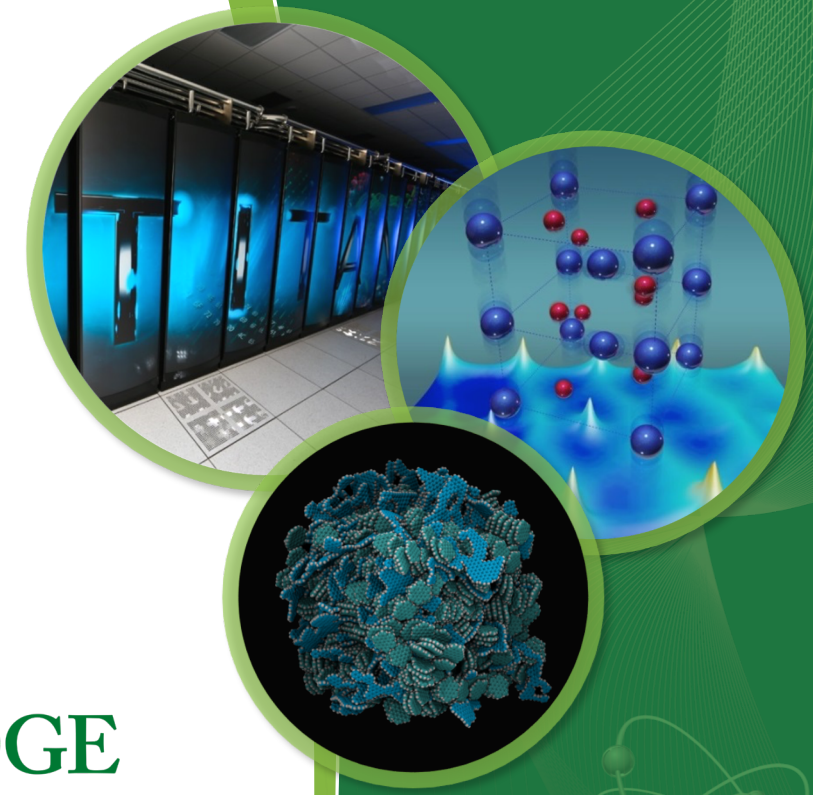
PMIx in unifyFS

Michael J. Brim, ORNL/OLCF

PMIx ECP Community BOF

March 31, 2021



Lawrence Livermore National Laboratory

OAK RIDGE National Laboratory

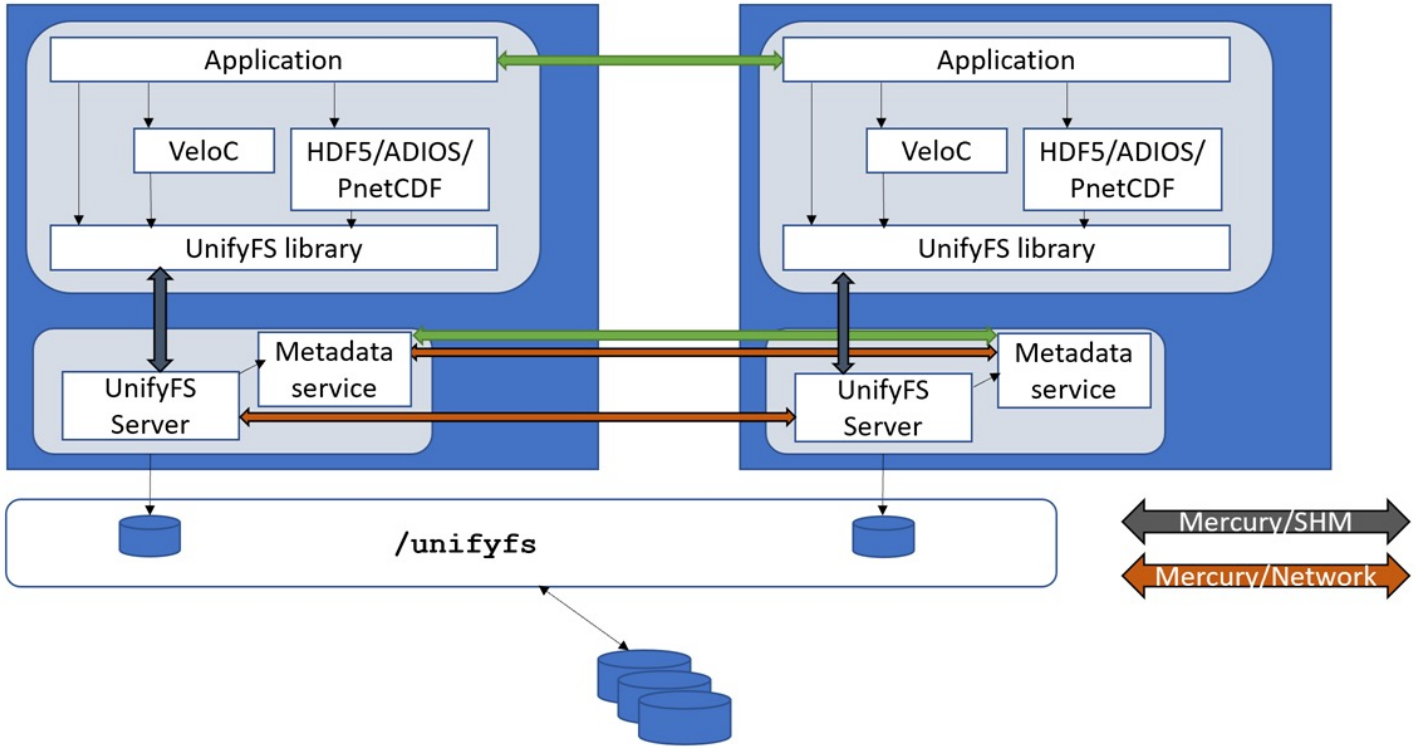
NCSA

What is UnifyFS?

https://unifyfs.readthedocs.io/en/latest/

https://github.com/LLNL/UnifyFS

- A user-level file system that *unifies* independent distributed burst buffer storage (e.g., node-local NVMe on Sierra/Summit, future near-node storage)
- Allows applications to treat burst buffer storage as just another parallel file system
  - N-to-1 shared files
  - N-to-N private files
- Customizable behavior for improved usability or performance



unifyFS

How do we use PMIx?

- For bootstrapping, of course
  - need to share server connection information (i.e., Mercury address strings)

- PMIx provides one implementation of our generic key-value store
  - others are PMIv2 and shared file system
  - see: https://github.com/LLNL/UnifyFS/blob/dev/common/src/unifyfs_keyval.c

- We use `PMIx_Publish()`, `PMIx_Fence()`, and `PMIx_Lookup()`
  1. Each server publishes two key-value pairs
  2. Use fence as a barrier synchronization across servers
  3. Each server looks up keys of all other servers

Current Status on OLCF Summit

- It works using IBM Job Step Manager ☺
  - `jsrun (Job Step Manager) 10.03.01.02rtm0 [Jan 21, 2020] built with PMIx 3.1.4`
- but bootstrap fails for ~30% of runs at 512 nodes ☹
  - Roughly half the nodes report failure:
    - `PMIx_Lookup("0.unifyfsd.pmi-rank") failed: UNREACHABLE`
  - What does "UNREACHABLE" error code really mean?
- so, we end up using shared file system as the keyval store ☹

unifyFS

Q: How should we be using PMIx?

- Anything obviously wrong with our approach?
  - Publish keys in `PMIX_RANGE_GLOBAL`
  - All-to-all key lookup with wait=1
    - Are we DDOSing PMIx servers?

- Does `PMIx_Fence()` guarantee published keys are visible?
  - documentation only mentions relation to `PMIx_Put()`

Acknowledgments

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.