Accelerating application I/O with UnifyFS

ECP Community BOF - DAVS
March 30, 2021

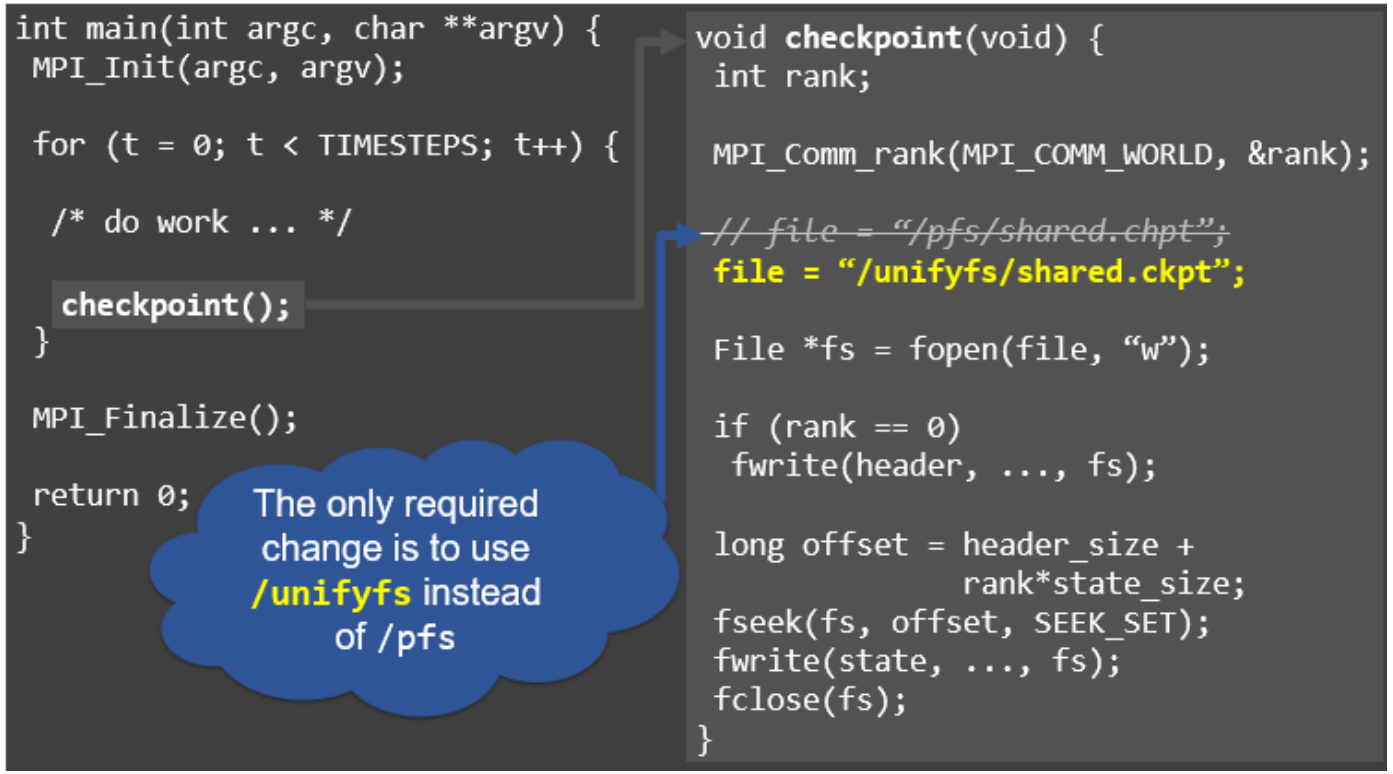
LLNL: **KATHRYN MOHROR** (PI), ADAM MOODY, CAMERON STANAVIGE, TONY HUTTER

ORNL: SARP ORAL (ORNL-PI), HYOGI SIM, FEIYI WANG, MIKE BRIM, SEUNG-HWAN LIM, JENNA DELOZIER

NCSA: CELSO MENDES, CRAIG STEFFEN

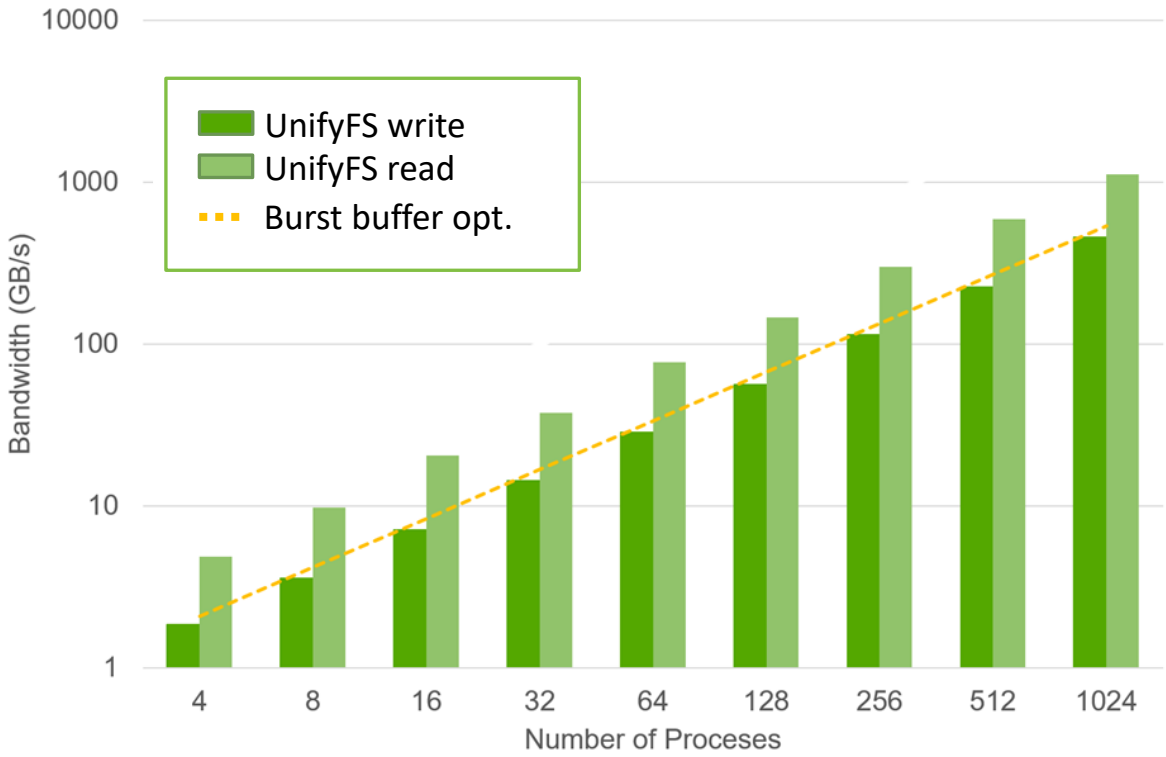
What is UnifyFS?

- Simply put, it's a file system for burst buffers
- Our goal is to make using burst buffers on exascale systems as *easy* as writing to the parallel file system and orders of magnitude *faster*



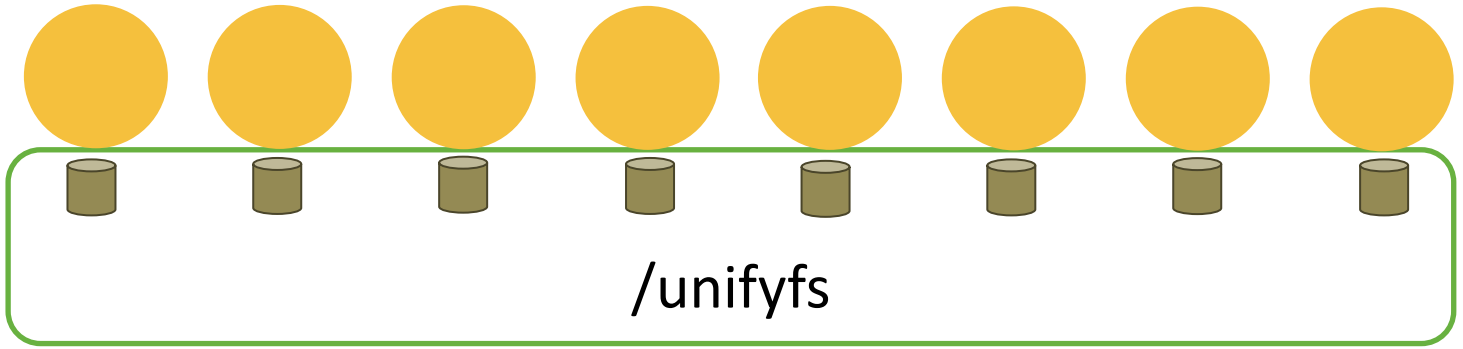
UnifyFS targets local burst buffers because they are fast and scalable

- UnifyFS (v0.9.1) scaling tests on Summit
  - All writes using burst buffer (no file data stored in memory for these runs)
  - Write performance is equal to the cumulative theoretical throughput of the node-local burst buffer
  - Read performance takes advantage of caching



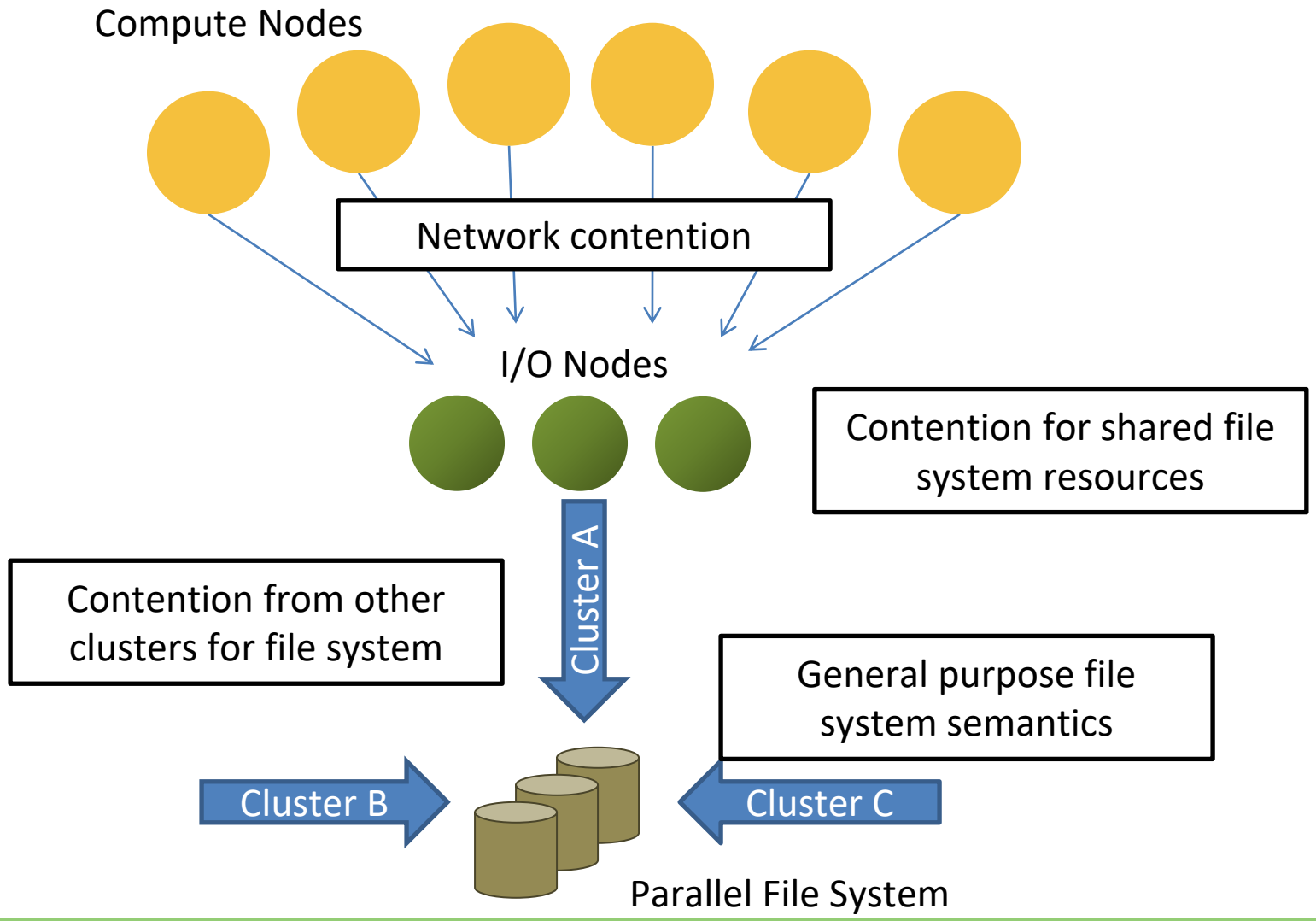
UnifyFS makes sharing files on node-local burst buffers easy and fast

- Sharing files on node-local burst buffers is not natively supported



- UnifyFS makes sharing files **easy**
  - UnifyFS presents a shared namespace across distributed storage
  - Used directly by applications or indirectly via higher level libraries like VeloC, MPI-IO, HDF5, PnetCDF, ADIOS, etc.
- UnifyFS is **fast**
  - Tailored for specific HPC workloads, e.g., checkpoint/restart, visualization output, loose coupling through files
  - Each UnifyFS instance exists only within a single job, no contention with other jobs on the system

Writing data to the parallel file system is expensive



UnifyFS is designed to work completely in user space for a single user's job

```
# Job Script Integration
unifyfs start –mount=/unifyfs
# run applications and tasks...
unifyfs terminate
```

Batch Job Manager

Dynamic File System for Each Job

HPC Application
(N-1, N-N Checkpointing, Collective I/O with Scientific I/O Libraries)

Checkpoint/Restart Lib: SCR, VeloC

Scientific I/O Lib: HDF5, MPI-IO, …

Standard POSIX I/O

UnifyFS

User level I/O interception

Fast data writes to node local storage

Data visible globally after "lamination"

Transfer API to move data to/from parallel file system

We need you!

- Our goal is to provide **easy, portable,** and **fast** support for burst buffers for ECP applications
- We need early users
- What features are most important to you
- Available on github: https://github.com/LLNL/UnifyFS
  - MIT license
- Documentation and user support
  - User Guide: http://unifyfs.readthedocs.io
  - unifyfs@exascaleproject.org

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

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.