OpenSFS Application I/O
kernel creation effort

Salishan 2015 Random Access Session

Ilene Carpenter, NREL
Application I/O kernel extraction task group

- What Application I/O kernels exist?
- What patterns are not represented?
- Develop application kernels to complement those that already exist, to allow evaluation of file system performance and scalability for specific application workloads.
What already exists?

The I/O associated with the class of applications that use MPI at large scale is well studied.

• Several previous efforts resulted in creation of application I/O benchmarks:
  o MADbench2, Chombo I/O, S3D-IO
  o Most of the I/O is associated with writing history files (results) or checkpoints.
  o Several layers of middleware are important for improving the I/O performance so (application + middleware) is the entity to benchmark.

• Many of the application I/O benchmark codes are not supported or available outside of the group that created them.
What patterns are not represented?

- I/O from applications that don’t use MPI IO, HDF5, netCDF, etc.
- I/O from non-traditional HPC applications for which source code is not available
- I/O from applications that use many files of different types, often with small I/O accesses, especially those that run many instances simultaneously on different data.
Examples:

• High throughput computational screening studies
• Parameter sweeps with serial or modestly parallel codes
• Genomic codes (ex. BLAST)
• Financial modeling codes
Challenges

• Often proprietary applications
• Uncoordinated POSIX I/O from different instances of the application running simultaneously.
Plan

• Develop application kernels to complement those that already exist, to allow evaluation of file system performance and scalability for specific application workloads.

• Use trace based methods to create surrogate applications.
Current status

• In 2014, came across an interesting paper by Babak Behzad (then a student of Marc Snir) about creation of surrogate applications that display the same I/O patterns as a traced application.

• Began discussions to see whether the tools he developed could be extended to cover any of the cases we’re interested in.

• Using Pynamic benchmark as a test case.
OpenSFS BWG


- Co-Chairs: Devesh Tiwari (tiwari@ornl.gov), and Andrew Uselton, (andrew.c.uselton@intel.com)

- GET INVOLVED!