PLFS: A Checkpoint Filesystem for Parallel Applications

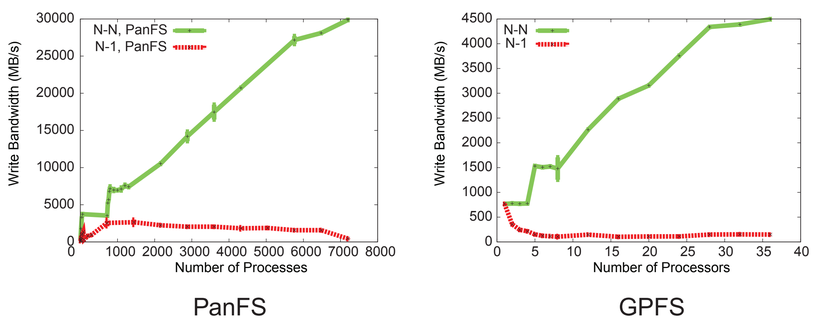
John Bent*, Garth Gibson†, Gary Grider*, Ben McClelland*, Paul Nowoczynski‡,
James Nunez*, Milo Polte†, Meghan Wingate*

**Los Alamos National Laboratory †Carnegie Mellon University ‡Pittsburgh Supercomputing*

Problem

- **Many important scientific applications create checkpoints using small, strided, concurrent writes to a shared file (N-1 checkpointing)**
- **Filesystems perform best on non-concurrent sequential workloads, such as N-N checkpointing**
- **Small-strided writes to a shared file often suffer from seeks and false sharing**
- **Unfortunately, we can't change the applications, but we can modify our filesystems**

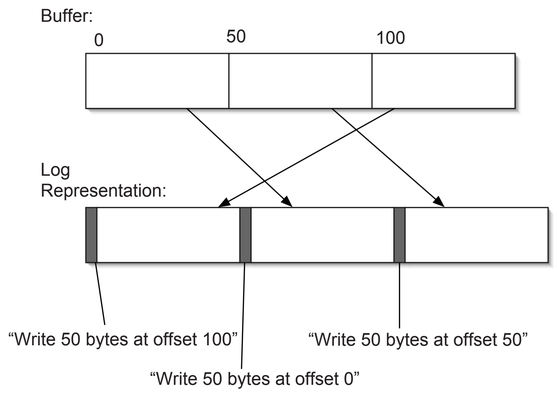
Checkpoint Bandwidth



Previous Work

- **"Log-structured Files for Fast Checkpointing"**
- **CMU students modified a parallel filesystem (PVFS2) to write shared files in a log representation**
  - **Required server modification**
  - **Only works with one filesystem**
  - **All writes are appended so no seeks, but clients still concurrently access a single file**

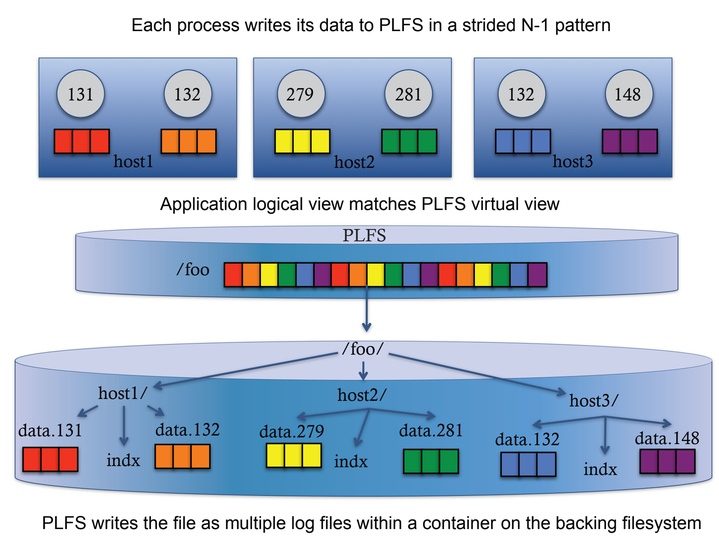
```
pwrite(file, buffer[100], 50, 100);
pwrite(file, buffer, 50, 0);
pwrite(file, buffer[50], 50, 50);
```



PLFS – Parallel Log-Structured File System

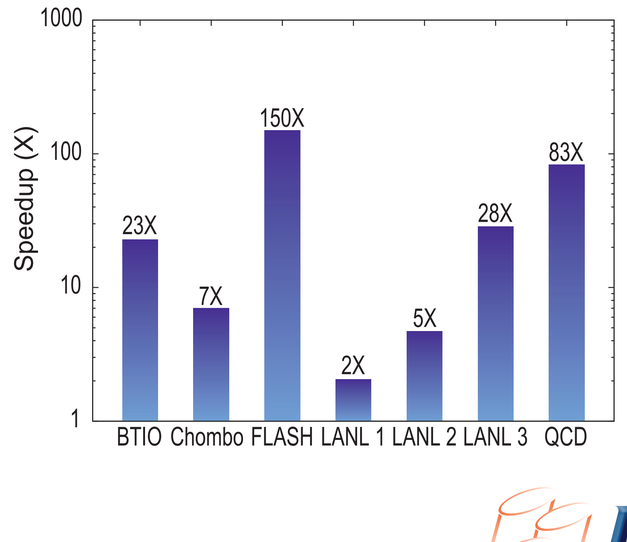
- **A project developing a filesystem level improvement to N-1 checkpointing, led by John Bent (LANL)**
- **FUSE based filesystem mounted on top of any existing parallel filesystem on clients**
- **Decouples a concurrent N-1 checkpoint into a non-concurrent N-N checkpoint**
- **Redirects strided writes from multiple processes accessing a single file to sequential writes to data logs and index files**

Layout of a PLFS Container



PLFS Speedup

- **2x – 150x speedups for important HPC applications at LANL scale!**



Elaborations and Future Work

- **PLFS can generate light-weight write map traces**
  - **http://institute.lanl.gov/plfs/maps/**
- **Adding additional functionality to other filesystems**
  - **Example: Concurrent write support for HDFS**
- **Further read path improvements**
  - **PLFS does not have a read penalty on checkpoint workloads**
  - **But what about analysis? Visualization?**
  - **Metadata servers for in-memory indices for read-write mode**

Carnegie Mellon
with information from LANL Technical Release LA-UR 09-02117

pdsi

IRHPIT INSTITUTE FOR RELIABLE HIGH PERFORMANCE INFORMATION TECHNOLOGY

Parallel Data Laboratory
R09