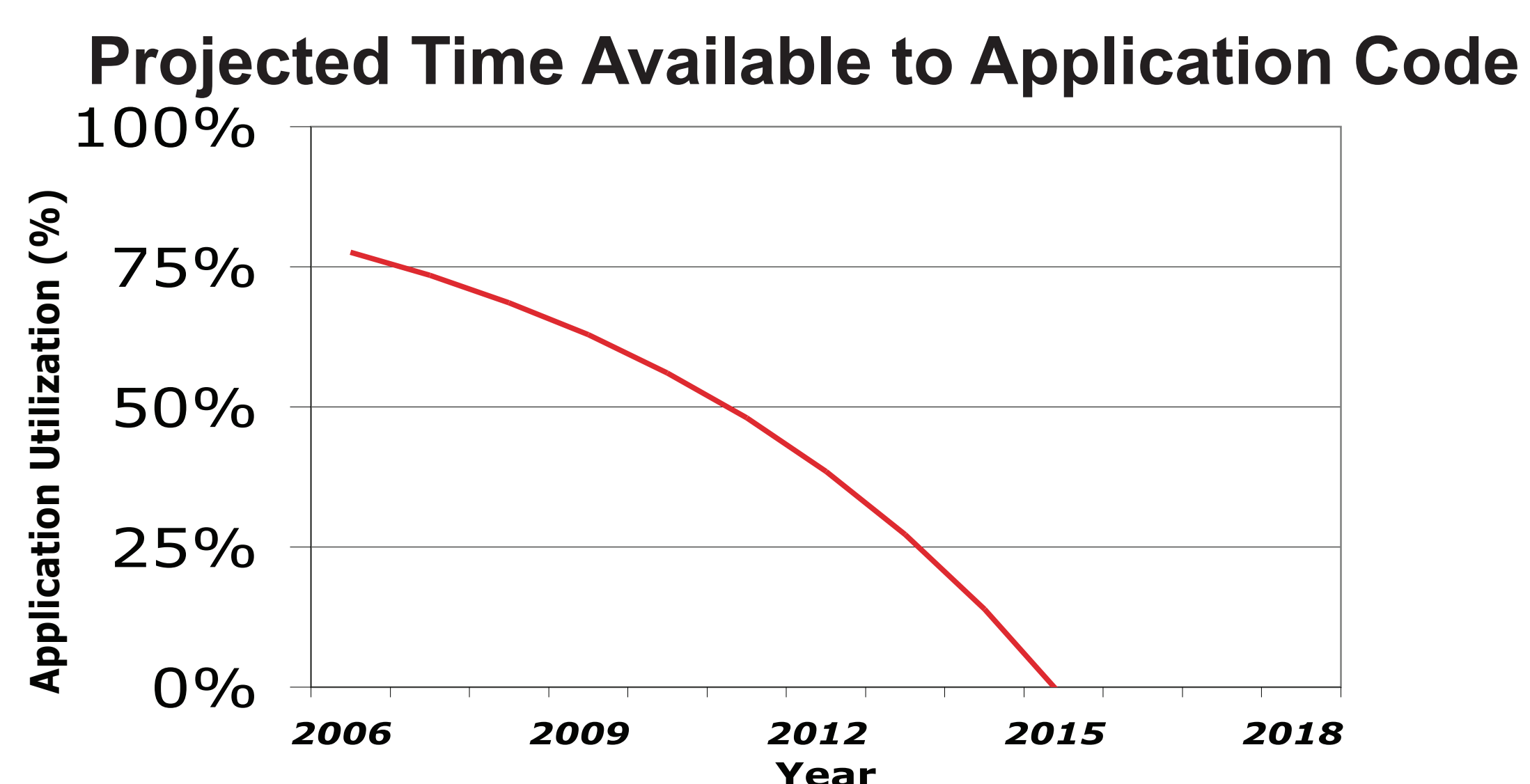


Log-structured Files for Fast Checkpointing

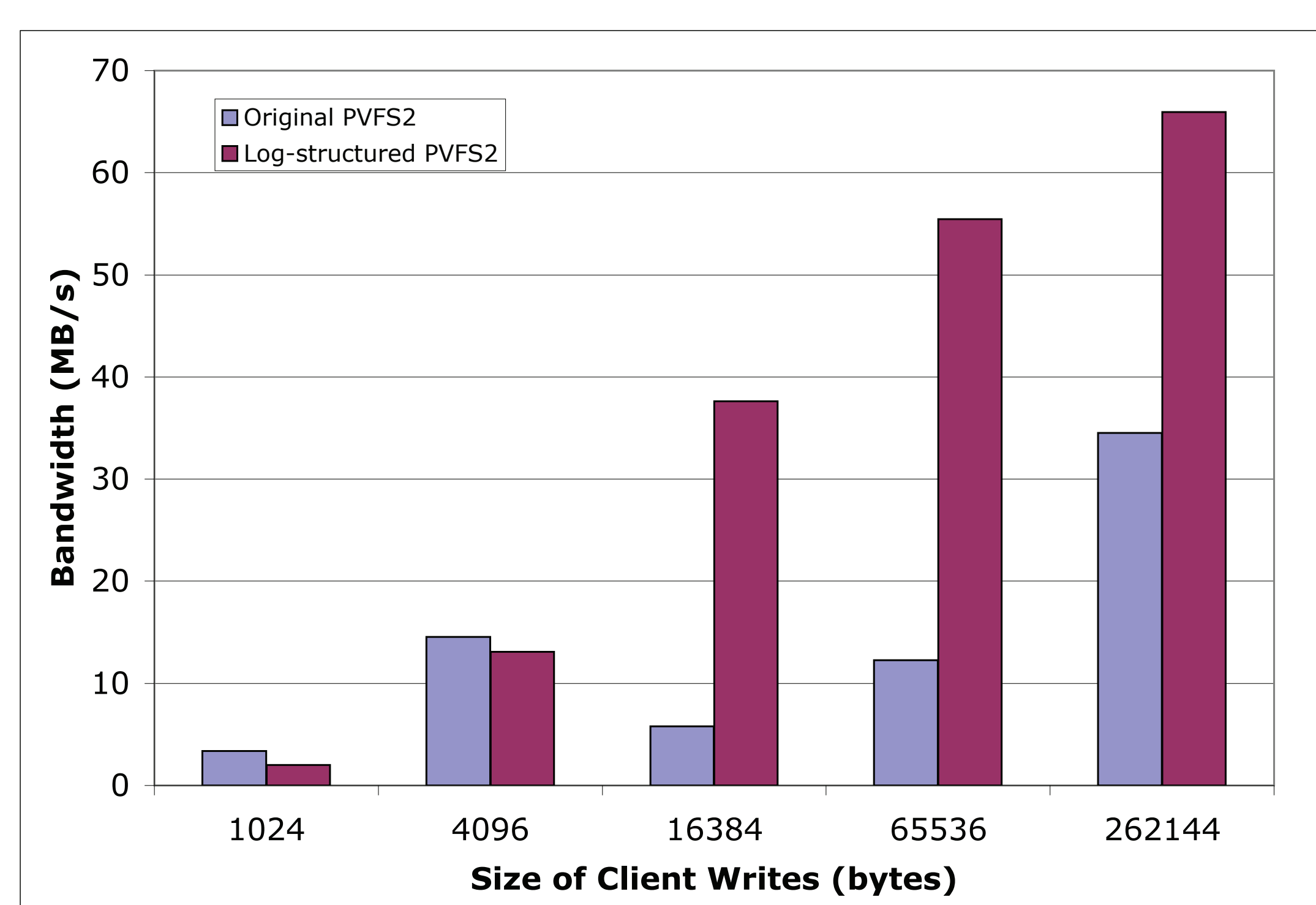
Milo Polte, Jiri Simsa, Wittawat Tantisiriroj, Shobhit Dayal, Mikhail Chainani,
Dilip Kumar Uppugandla, Garth Gibson

Motivation

- Approaching a checkpointing catastrophe in HPC:
 - Systems with more nodes, failures
 - Larger datasets
 - Longer running, more frequent checkpoints
- At current trends, checkpoints will prevent most useful application work within the next decade

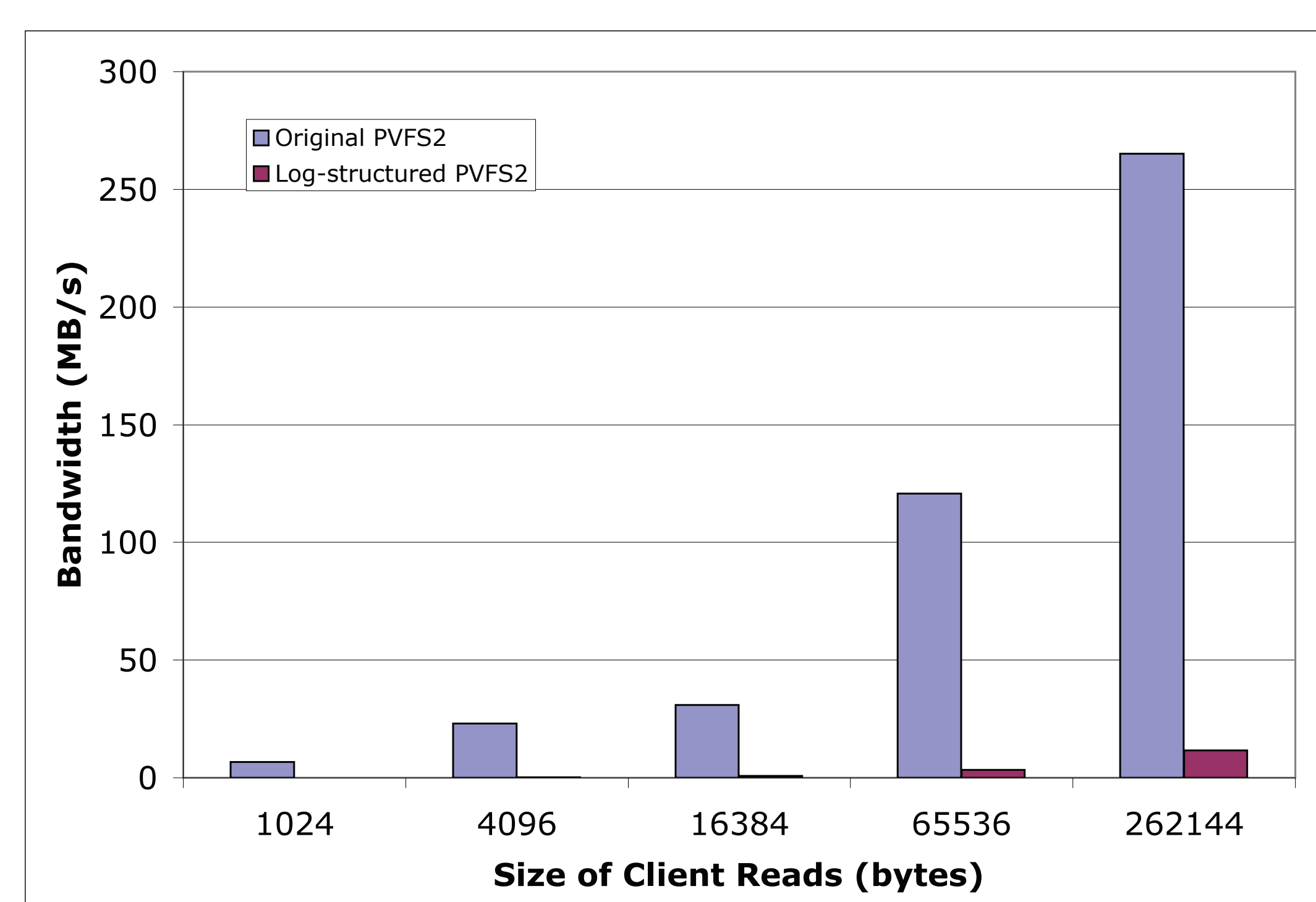


Write Performance



- Best benefit seen for 16k writes
- 1k writes penalized by header overhead
- 4k writes have alignment issues

Read Performance



- Performance poor as reads scan entire file for applicable writes

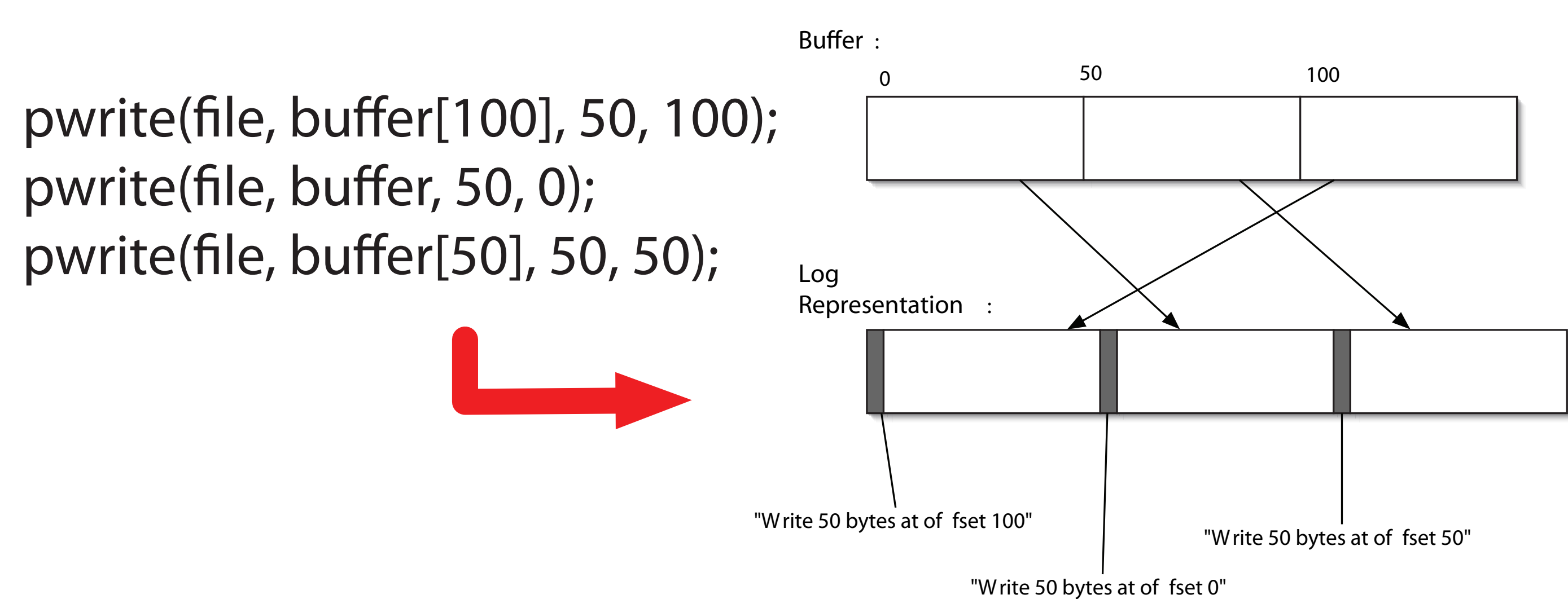
Carnegie Mellon



Checkpointing

- Technique for fault tolerance
- Compute nodes barrier sync, write state to storage
- No useful work until checkpoint complete
- Concurrent strided writers introduce more seeks
- Most checkpoints never used
 - 1996 log of jobs from LLNL had orders of magnitude fewer failed jobs than checkpoints

Log-Structured Writing



- Writes on disk in temporal order rather than logical
- Reduces time lost to write seeks, but can slow reading
- Not appropriate for all workloads, all files
- Appropriate for checkpoints (“write-once, read-maybe”)
- Our idea: Per-file log representations

Implementation

- Assigned as class project in Advanced Storage Systems
- Implemented in PVFS2, a parallel distributed filesystem
- Writes checkpoint files in a log structure
 - Each write is written to the end of the file with a header
 - Header contains logical location and size of the write
- Reads serviced naively by scanning file for all headers
 - Simple implementation
 - Checkpoints rarely read
 - Students not graded on read performance
- Evaluated with `mpi_io_test` using 10 clients

Current Status

- Student implementations show good write performance
- Promising potential for further work
- Read path improvements
 - Use footers for earlier terminated backwards scan
 - Flatten on first read
 - Separate index structure
- Generalize per-file representation technology
 - Per-file RAID
 - Optimized formats for scientific files

Parallel Data
Laboratory

