

Crossing the Chasm: Sneaking a Parallel File System into Hadoop

Wittawat Tantisiroj, Swapnil Patil, Garth Gibson

Overview

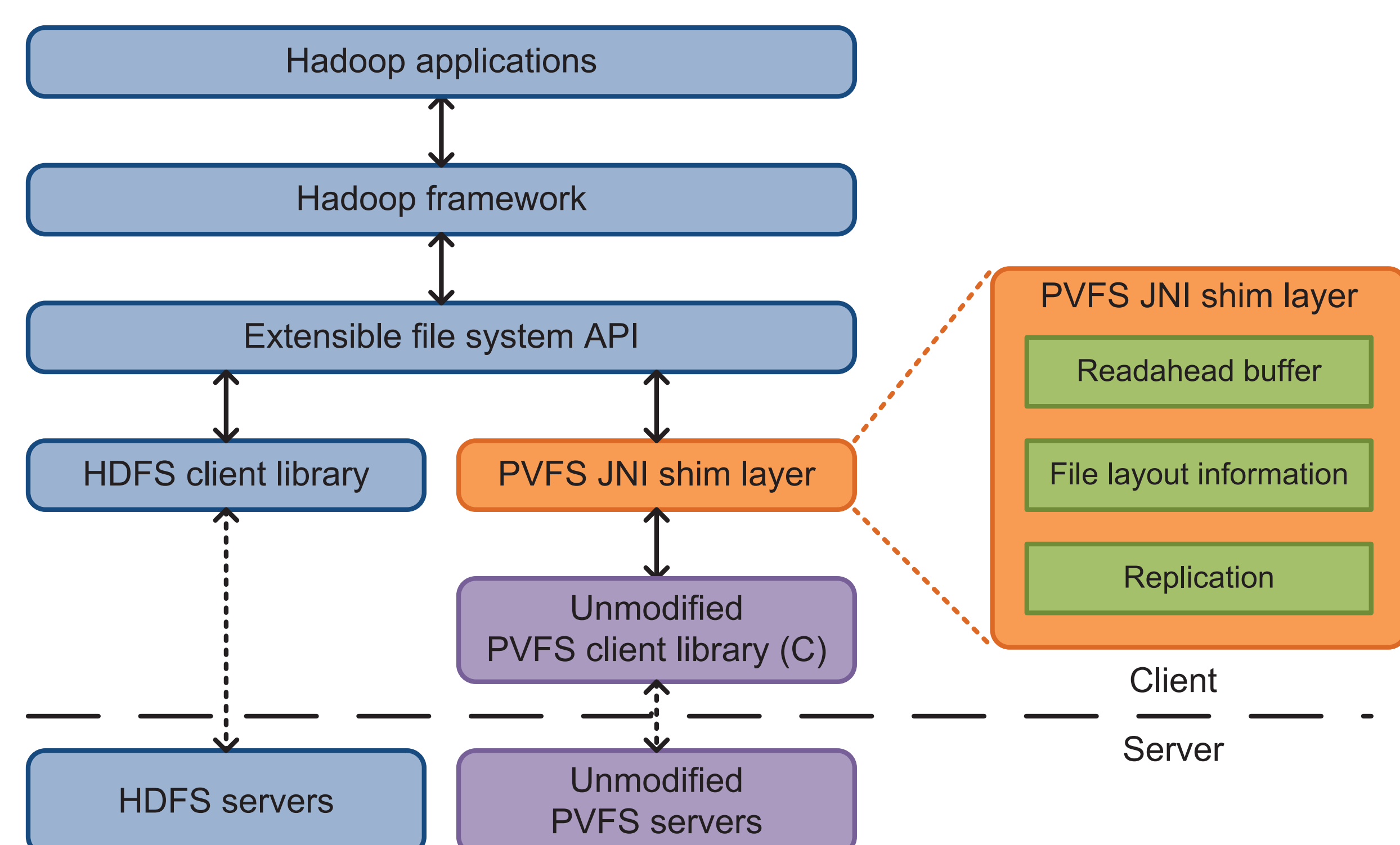
Internet Services

- Applications are becoming data-intensive
 - Large input data set (e.g. the entire web)
 - Distributed, parallel execution
- Distributed file system is a key component of the computing system
 - Purpose-built for anticipated workloads
 - New, diverse semantics (typically not support POSIX)
- Hadoop & Hadoop distributed file system (HDFS)
 - Distribute data across multiple nodes
 - Use triplication for reliability
 - Divide application into many small units of work
 - Use file layout information, which shows where data is located, to collocate computation and data

High performance computing (HPC)

- Equally large scale applications
 - Large input data set (e.g. astronomy data)
 - Distributed, parallel execution
 - Use parallel file systems for scalable I/O
- Parallel file system
 - Handle a wide variety of workload
 - Concurrent reads and writes
 - Small file support, scalable metadata
 - Typically support POSIX and VFS interface
 - Maturing and being standardized (pNFS)

PVFS plug-in under Hadoop stack

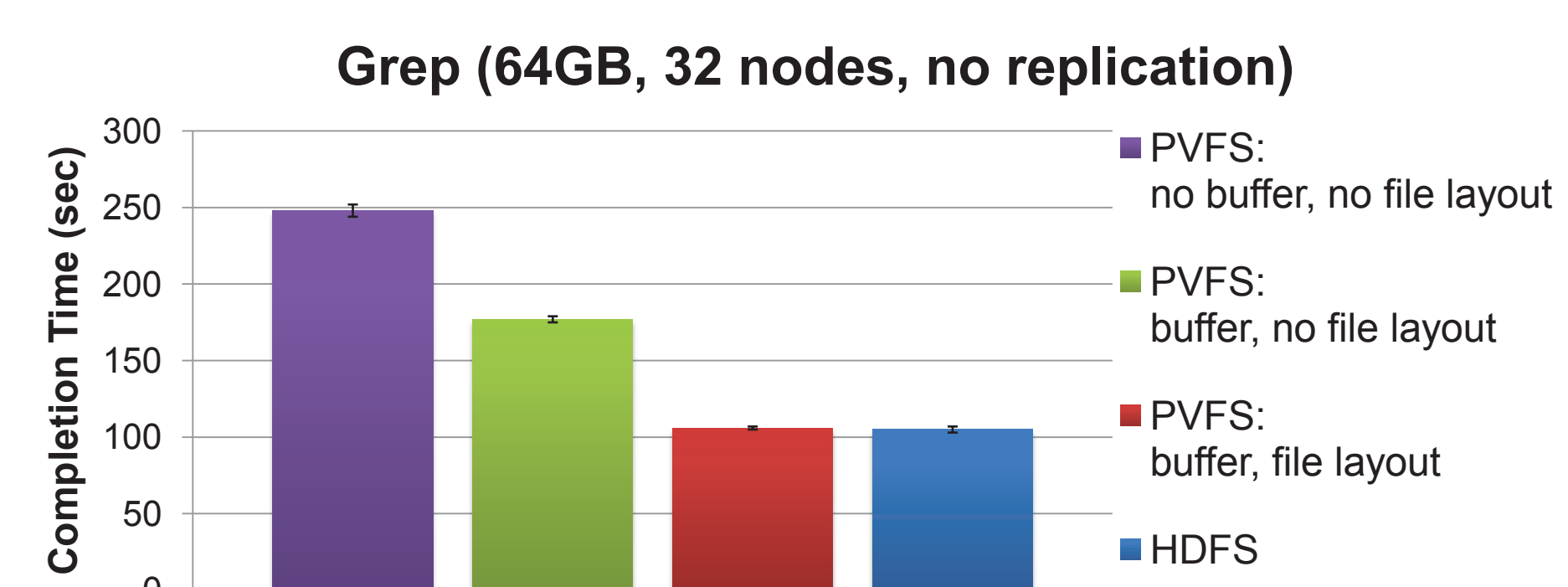


- Readahead buffer: reads 4MB from PVFS and replies in 4KB units to Hadoop
- File layout information: exposes file layout stored in PVFS as extended attributes
- Replication: triplicates write request to three PVFS files with disjoint layouts

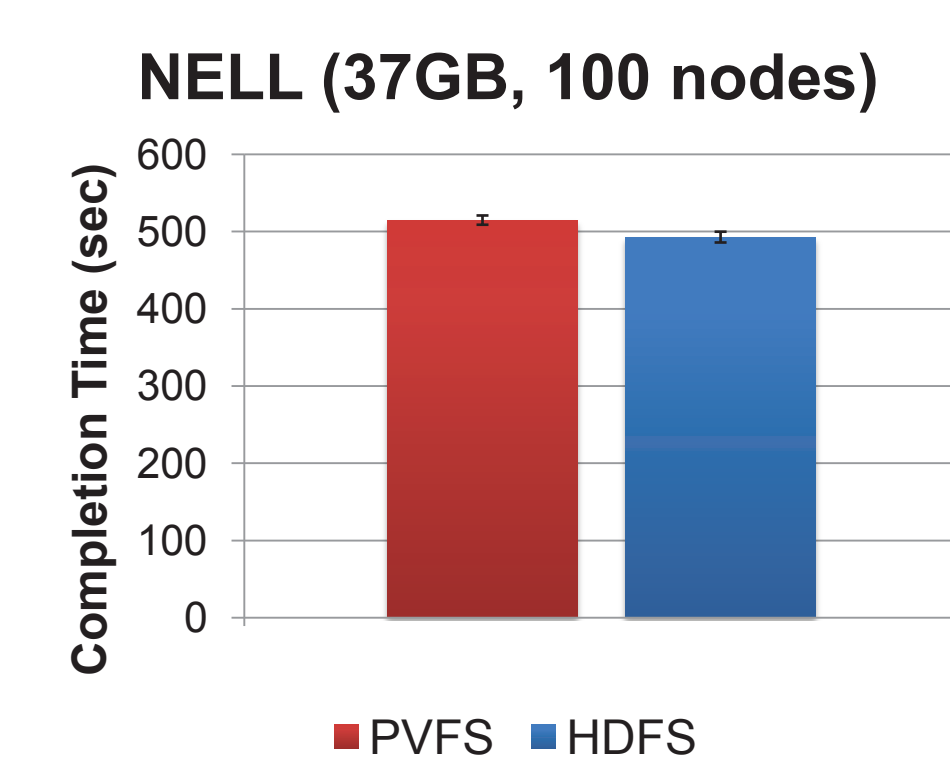
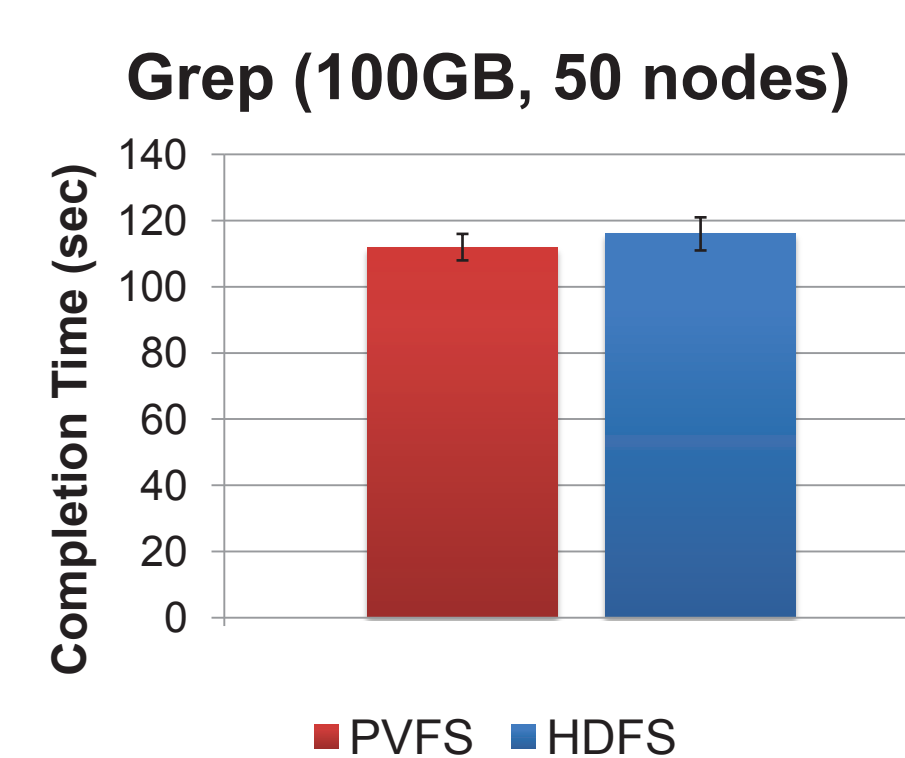
Experiment Setup

- Yahoo! M45 cluster (Xeon quad-core 1.86 GHz, 6GB Memory, 7200 rpm SATA 750 GB disks, Gigabit Ethernet)
- Benchmarks
 - Grep: Search for a rare pattern in a hundred million 100-byte records (100GB)
 - Sort: Sort a hundred million 100-byte records (100GB)
 - Never-Ending Language Learning (NELL): (from J. Betteridge) Count the number of selected phases in 37GB data-set

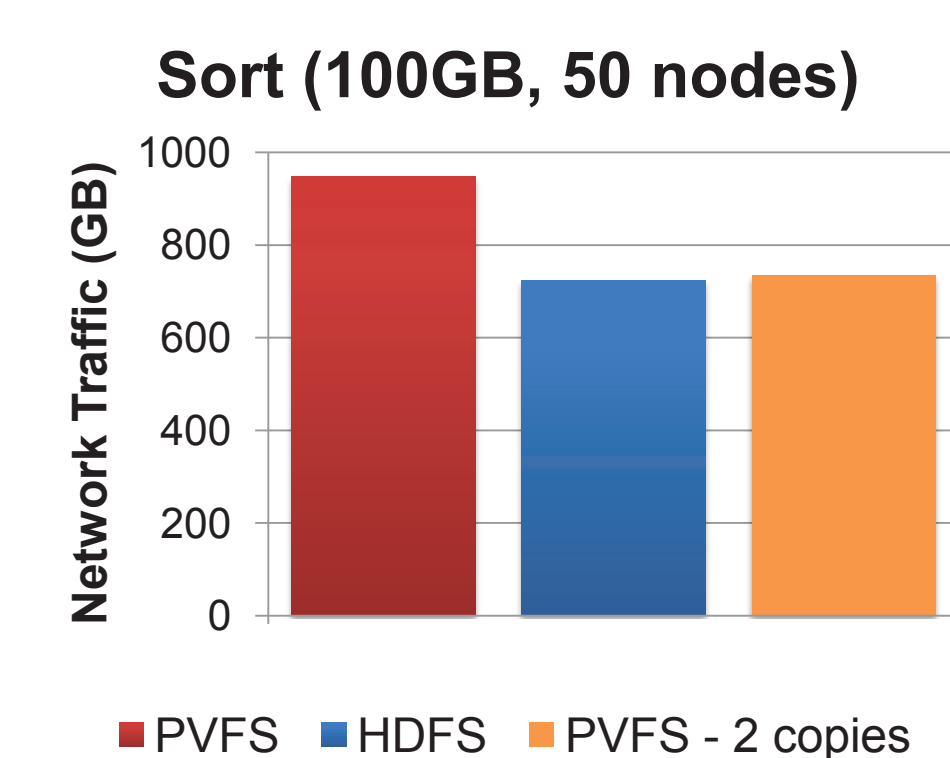
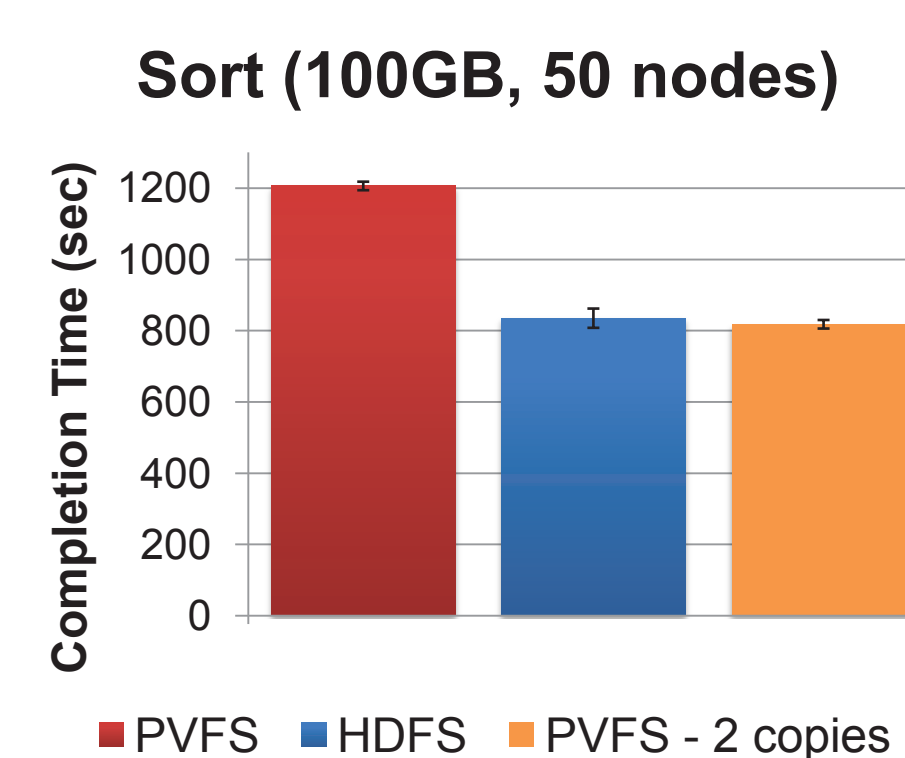
Experiment Results



- By using both readahead buffer and file layout information, PVFS performance is comparable to HDFS



- PVFS performance is comparable to HDFS for both read-intensive applications



- *sort* using HDFS is faster than running *sort* on PVFS because HDFS writes the first copy locally

Conclusion

- With few modification in a non-intrusive PVFS shim layer, PVFS delivers promising performance for Hadoop applications
- File layout information is essential for Hadoop to collocate computation and data

Acknowledgements

- Sams Lang, Rob Ross, Yahoo!, Julio Lopez, Justin Betteridge, Le Zhao, Jamie Callan, Shay Cohen, Noah Smith, U Kang and Christos Faloutsos

