ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CSCS
Swiss National Supercomputing Centre

Parallel I/O and the Metadata Wall

Sadaf R Alam, Hussein N El-Harake, Kristopher Howard,
Neil Stringfellow & Fabio Verzelloni

PDSW11, Seattle, Nov 13th 2011

Parallel I/O in a Supercomputing Centre

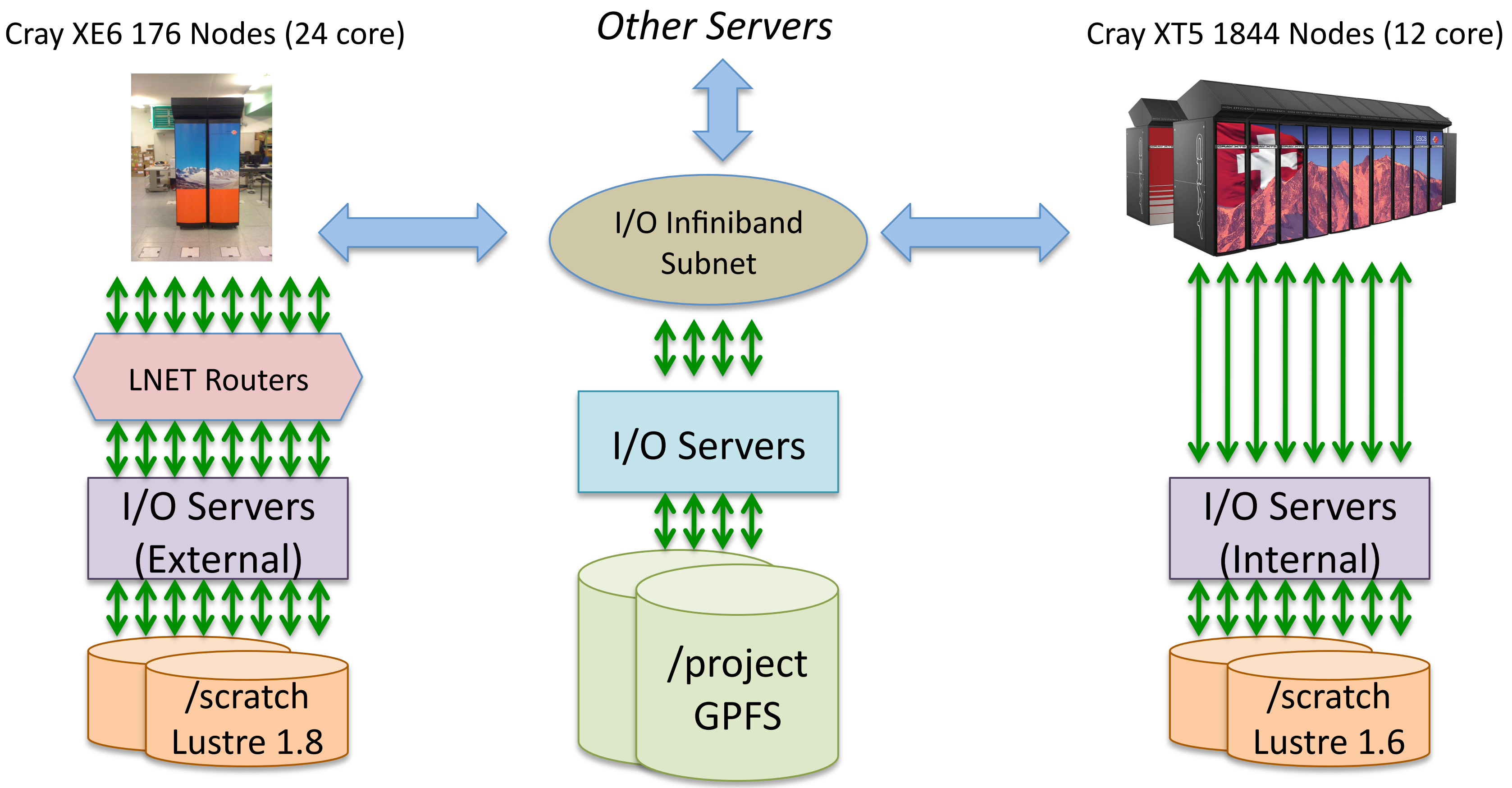
The Centre's View

- Interested in supporting "Big data"
- Concentrate on high bandwidth requirements
- Buying I/O infrastructure is often not the main priority in procurements
  - High capacity and bandwidth are often the main factors being considered
- Metadata not normally an issue in procurement

The Scientist's View

- Interested in the necessary I/O for their problem
- Time-to-solution of their problem is the important metric
- Scientist may choose to write to many files (one per process) or one large file
- Scientist does not want their job impacted by other users

Anatomy of a Supercomputing Centre’s I/O Setup



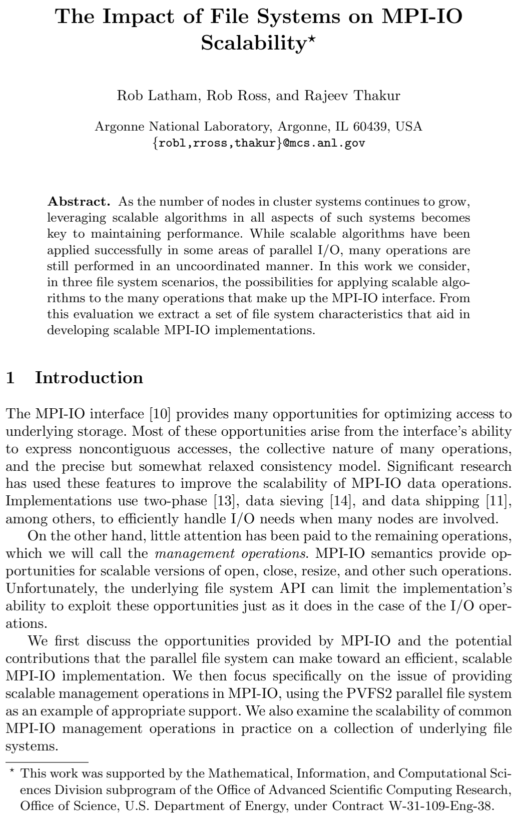
- Most of the I/O problems reported by users can be traced to metadata problems
- Often the metadata problems are caused by someone else but affect everybody

A Selection of Problems We've Seen

- Thousands of processors using a file to communicate
  - An old code from ~10 years ago that ran on a few processors was suddenly being used on thousands of cores
  - This caused severe performance degradation of the file system
- Jobs generating thousands of directories and files
  - Subsequently multiple jobs to delete them
    - Like a /scratch clean policy on steroids
- Scaling of MPI_File_open
  - Time for MPI_File_open scales linearly with number of MPI processes
  - At 1,000 processes MPI_File_open takes between 0.1 and 0.3 seconds on our file systems
    - *Imagine the same thing on 100,000+ processors*
  - Bandwidth gain in going from serial to parallel I/O can be negated by extra file open overhead

Parallel File System Support for MPI File Open

- Most parallel file systems use POSIX I/O semantics
- Individual processes have to open a file independently
- Problem was discussed by Latham et al.
- PVFS has support for MPI file open to only require one metadata request
- Many vendors only support Lustre and/or GPFS when selling a system
- Romio's "deferred file open" *could* help



The Impact of File Systems on MPI-IO Scalability*

Rob Latham, Rob Ross, and Rajeev Thakur

Argonne National Laboratory, Argonne, IL 60439, USA
{robl,rross,thakur}@mcs.anl.gov

Abstract. As the number of nodes in cluster systems continues to grow, leveraging scalable algorithms in all aspects of such systems becomes key to maintaining performance. While scalable algorithms have been applied successfully in some areas of parallel I/O, many operations are still performed in an uncoordinated manner. In this work we consider, in three file system scenarios, the possibilities for applying scalable algorithms to the many operations that make up the MPI-IO interface. From this evaluation we extract a set of file system characteristics that aid in developing scalable MPI-IO implementations.

1 Introduction

The MPI-IO interface [10] provides many opportunities for optimizing access to underlying storage. Most of these opportunities arise from the interface's ability to express noncontiguous accesses, the collective nature of many operations, and the precise but somewhat relaxed consistency model. Significant research has used these features to improve the scalability of MPI-IO data operations. Implementations use two-phase [13], data sieving [14], and data shipping [11], among others, to efficiently handle I/O needs when many nodes are involved.

On the other hand, little attention has been paid to the remaining operations, which we will call the *management operations*. MPI-IO semantics provide opportunities for scalable versions of open, close, resize, and other such operations. Unfortunately, the underlying file system API can limit the implementation's ability to exploit these opportunities just as it does in the case of the I/O operations.

We first discuss the opportunities provided by MPI-IO and the potential contributions that the parallel file system can make toward an efficient, scalable MPI-IO implementation. We then focus specifically on the issue of providing scalable management operations in MPI-IO, using the PVFS2 parallel file system as an example of appropriate support. We also examine the scalability of common MPI-IO management operations in practice on a collection of underlying file systems.

* This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-Eng-38.

Software or Hardware Problem ?

- The problems we see with metadata have to pass through software layers at application (MPI or POSIX I/O) and system (file systems) level
- The underlying hardware of the file systems can be a limiting factor
- Could improvements in hardware relieve our problems
  - In particular the use of SSDs for metadata targets
  - Enterprise SSD Vendors are promoting their products as potential metadata server targets

Experiments using `mdtest`

- The `mdtest` suite is “an MPI-coordinated metadata benchmark test that performs open/stat/close operations on files and directories and then reports the performance”
  - `mdtest` does not use MPI-I/O, but uses multiple MPI processes to carry out the file I/O operations
- We used the metarates benchmark suite for verification of results
- For tests of SSD hardware as metadata targets we sued a small test cluster

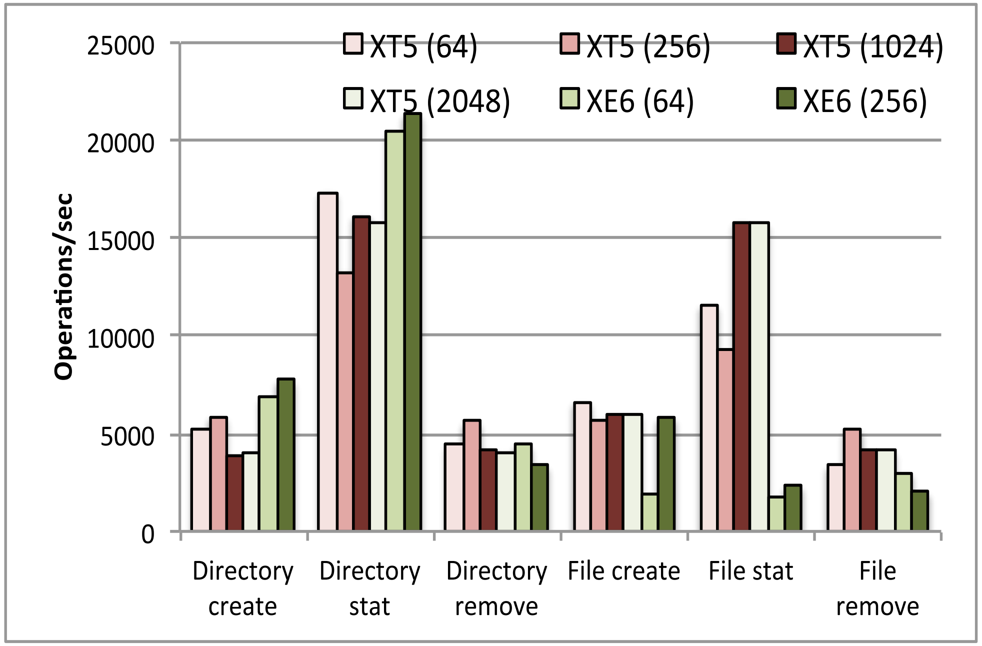
Experiments on Production Systems

- The goal of our study is to extrapolate the return of investment in hardware for parallel file systems
- We therefore measured metadata rates on two target systems
  - The Cray XT5 system has an internal Lustre file system, has a SeaStar II interconnect and uses Lustre 1.6.5
  - The Cray XE6 has an external Lustre, which is connected through routers, has a Gemini interconnect and uses Lustre 1.8.4
  - Both systems use SATA disks as the metadata targets
  - `mdtest` scaling measurements were carried out while keeping the number of total files and directories constant to 120K
  - The high-speed network can also contribute to the performance
    - These experiments were carried out on live production systems
    - We present the best results from a number of repeated tests

Current Production Platforms

- We ran `mdtest` on our current production platforms' Lustre file systems under production conditions (shared with other users)
- Except for file stat and create results, we observe a consistent performance behavior across the two systems
- Metadata performance does not continue scaling with the number of clients
  - There is a drop in performance for certain operations such as directory related operations, an indication of the metadata wall at scale
- Experiments with larger number of files at scale do not show performance scaling

Numbers in parentheses indicate numbers of MPI processes (clients)

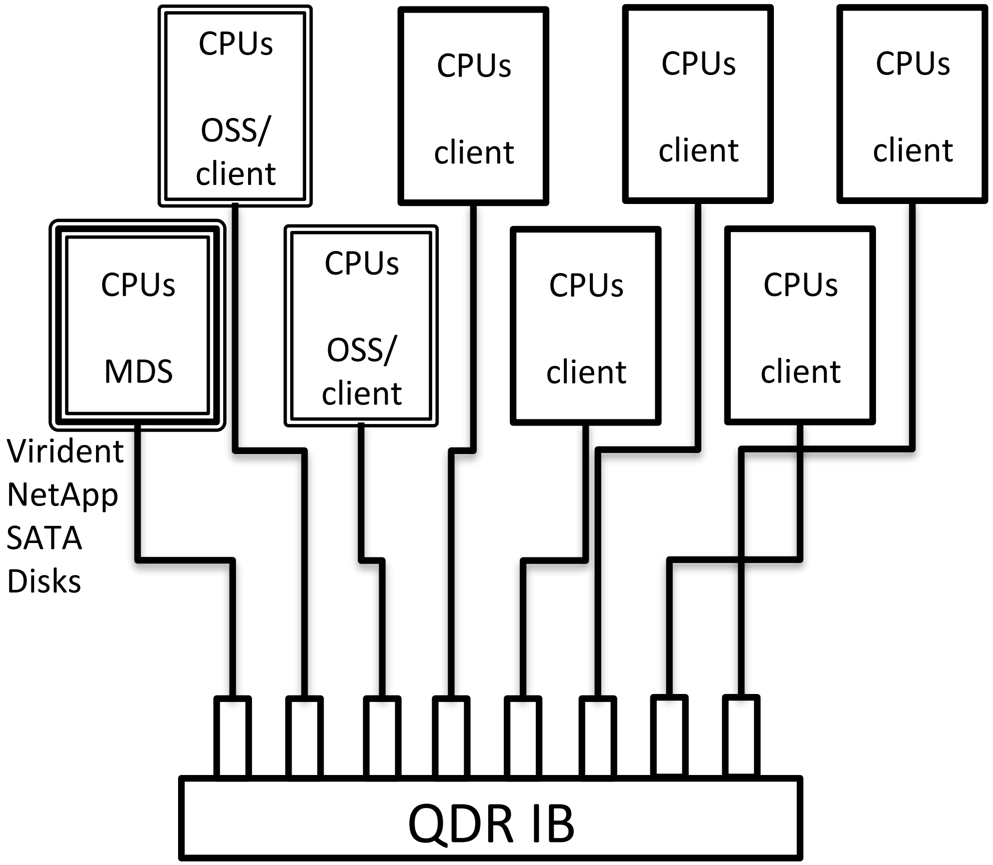


Experimental Hardware Evaluation Setup

- Servers and clients were from 7 dual-socket Intel Westmere nodes and 2 dual-socket AMD Magny-cours nodes
- Nodes were connected with a 36-port QDR switch
- Virident TachIOn SLC NAND
  - Theoretical peak of 300K IOPS for 4 Kbytes block sizes.
  - It should also deliver 1.44 GB/s (read) and 1.2 GB/s (write) performance.
- One couplet NetApp Pikes peak (E5412)
  - We used four SLC SSDs to create two RAID arrays
  - The controller is capable of ~120K IOPs for read and write operations using 4K block size.
- In addition to the above-mentioned devices, local SATA disks were also targeted for Lustre experiments.

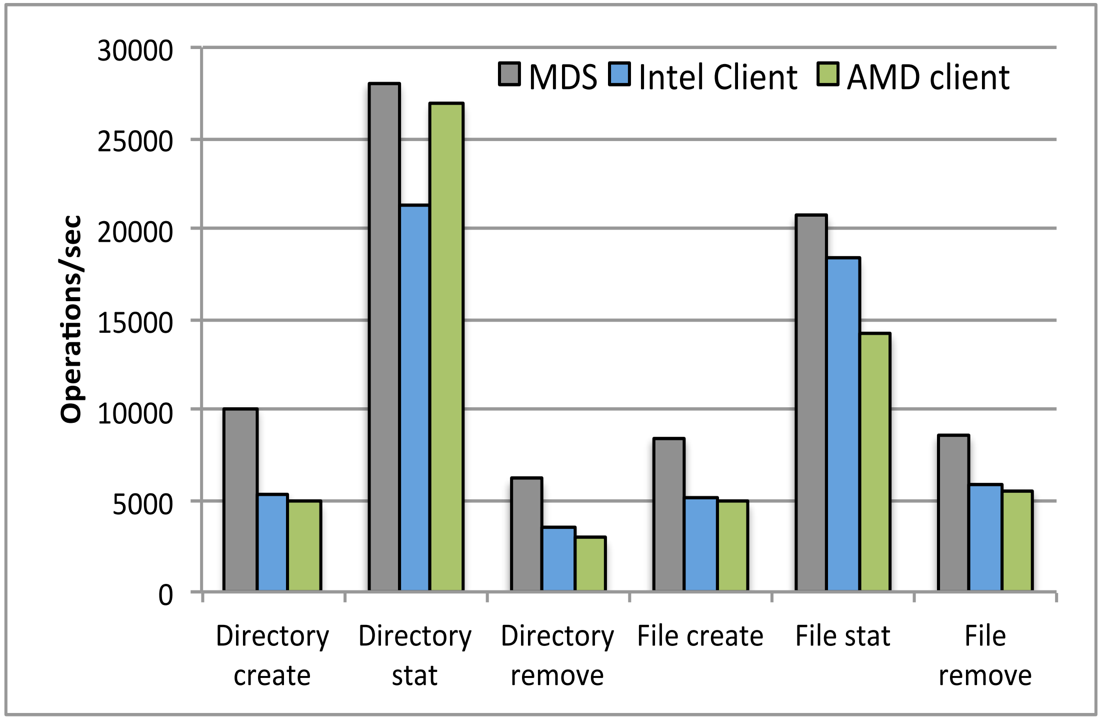
Lustre configuration for experiments

- We used Lustre 1.8.5 and configured 2 OSS and 1 MDS

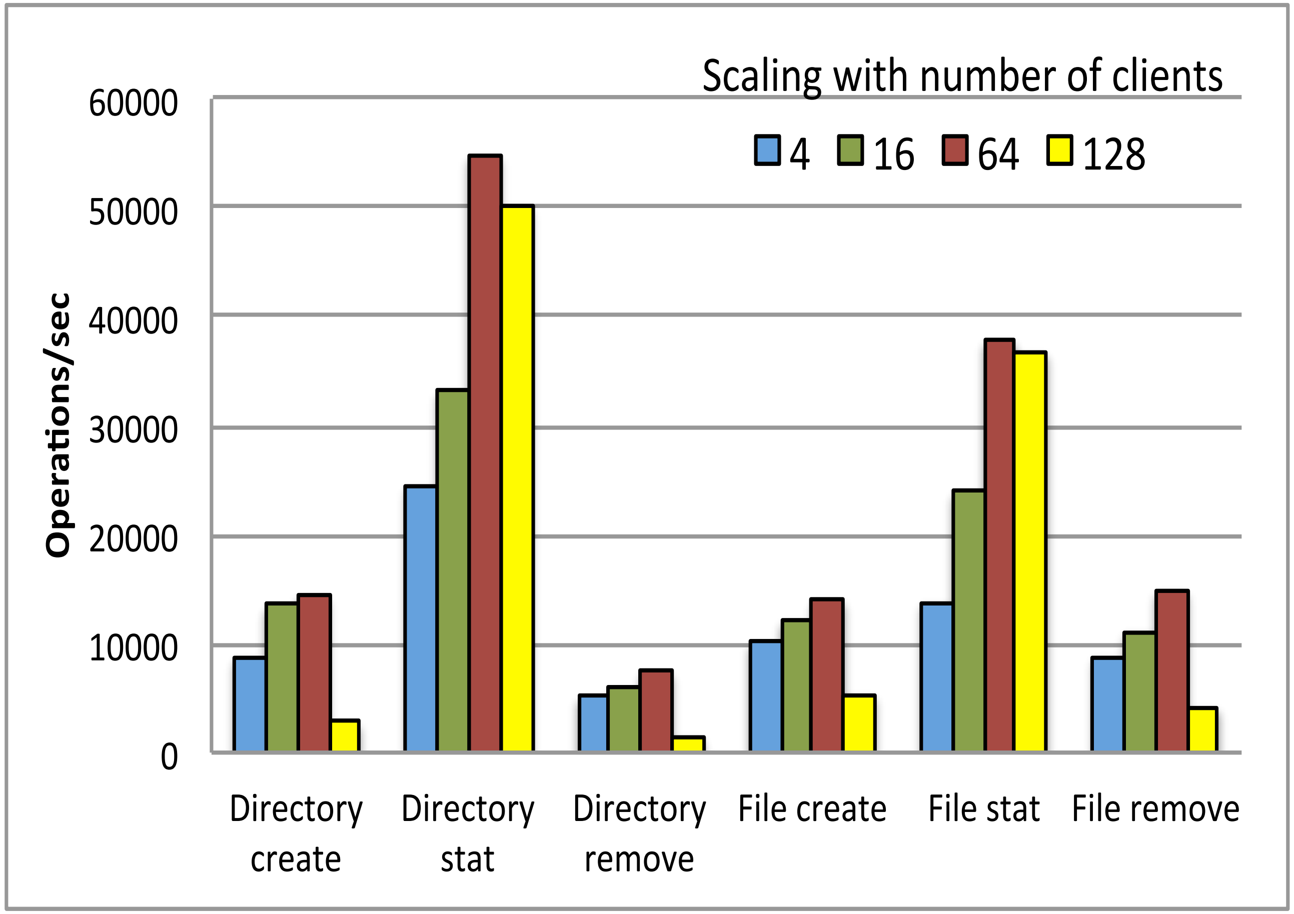


Processor and Network Sensitivity

- We measured the rates from different types of clients and from clients on the metadata server itself
- Tests were using Virident metadata targets and 300,000 file operations
- Clients running on the MDS did show better performance across the board
- As network latencies could contribute to the metadata throughput we did not include the MDS as a file IO client for Lustre experiments.



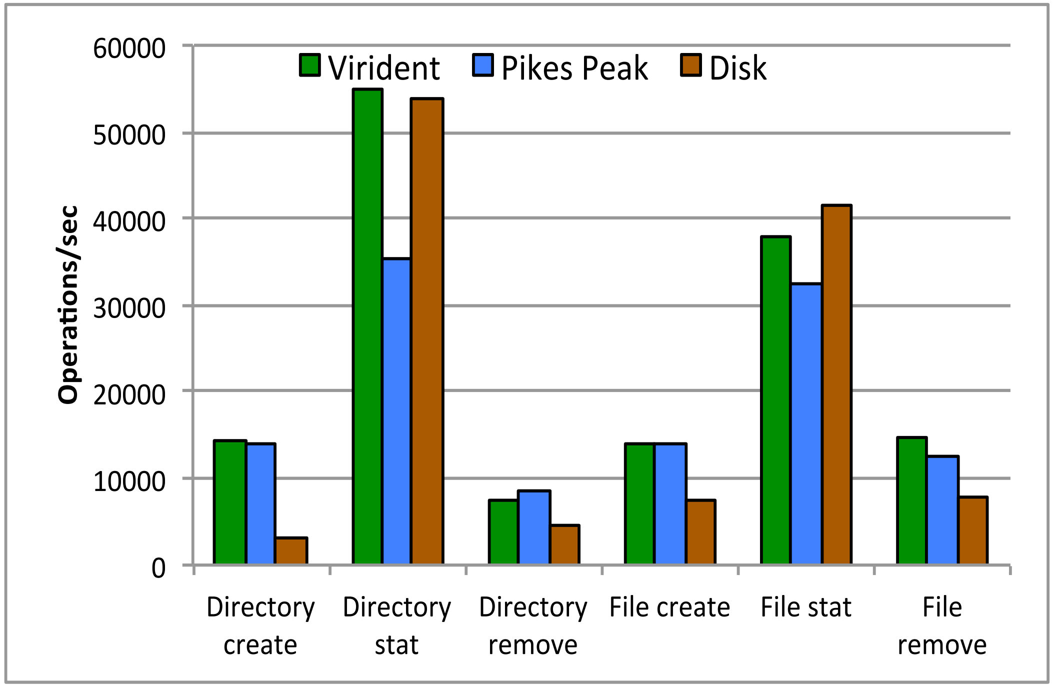
Sensitivity to numbers of clients



- 64 client delivered the best performance and was therefore the number used in the experiments, with 300,000 file operations
  - Results shown here are using the Virident SSDs

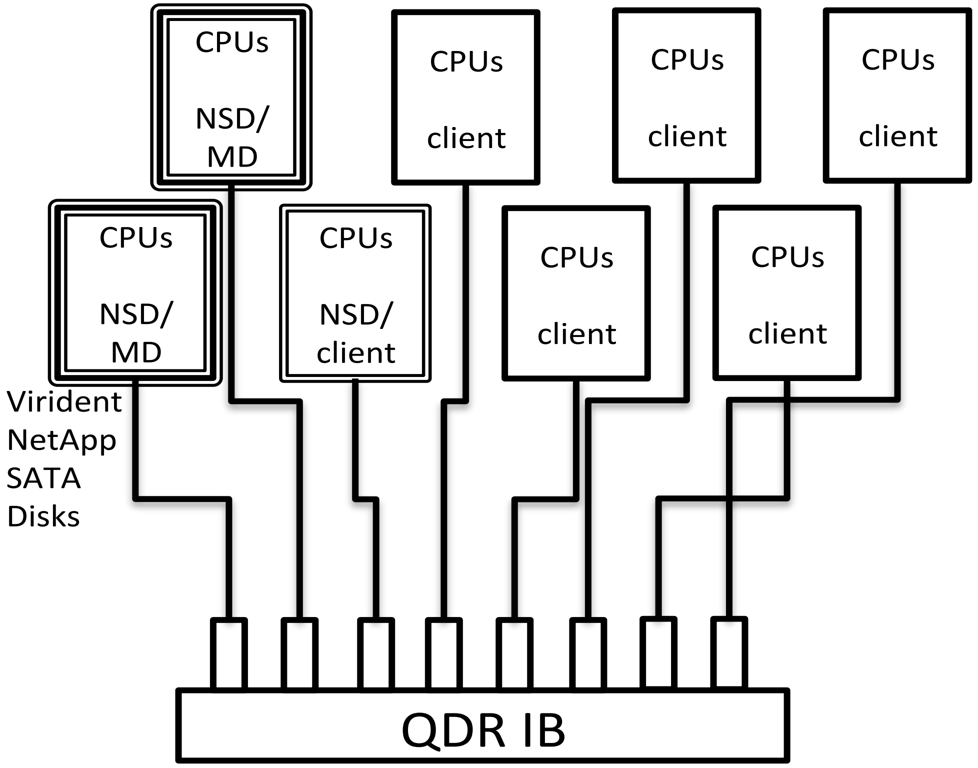
Lustre File System Results

- For stat operations there was little difference between SSD and disk
- For file create/remove there was up to 2X improvement
- For directory create there was around 4X improvement



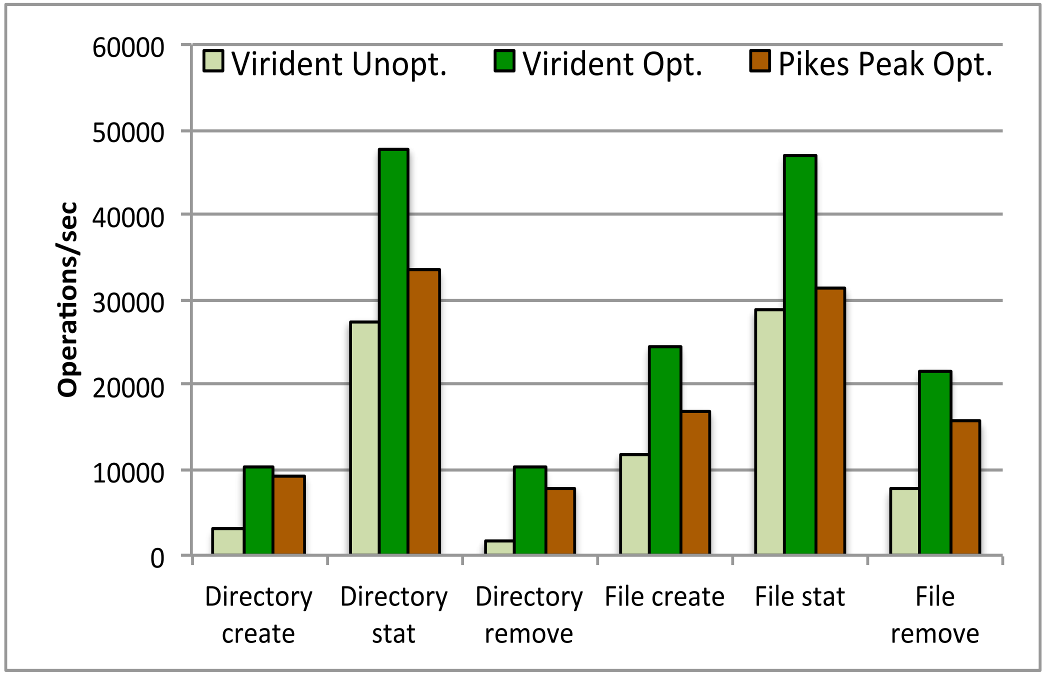
GPFS configuration for experiments

- We used GPFS 3.4.0-7
- As GPFS can be configured with multiple metadata targets we used 2 of these



GPFS File System Results

- For GPFS there was some extra effort required for file system tuning for the SSD controllers
- We were able to use multiple metadata servers with Virident



Some Observations

- The metadata results do not reflect the theoretical capabilities of the targeted hardware
  - Only a factor of 2 or 3 improvement over disk based metadata target systems were recorded
- For a given hardware, the operations/second for directory creation and removal are consistent across Lustre and GPFS
- The SSD hardware devices offer a potential for significant speedup for metadata operations which is not seen
  - The inherent software design limits of the parallel file systems is the likely inhibitor of this performance potential

Conclusions

- Previous work on strategies for improving metadata performance have been able to demonstrate performance improvements
  - These methods are not available in file systems such as Lustre and GPFS
    - These are frequently the only options promoted by vendors on large MPP systems and clusters that are installed at major HPC centers
- There are challenges in measurement and analysis of parallel file systems performance
  - There are several dependencies between the internal high-speed network and the storage area network
  - Caching at the disk controller level can influence performance measurements, which are not easily seen on the client side
- A factor of 2 to 4 improvement for some operations can be seen when targeting the SSD hardware for metadata
  - However this does not reflect the theoretical capabilities of SSD targets
- We have hit a “metadata wall” with major parallel file systems
  - Technological evolution in hardware alone may not be sufficient to address the issue