Parallel I/O and the Metadata Wall

Sadaf R Alam, Hussein N El-Harake, Kristopher Howard, Neil Stringfellow & Fabio Verzelloni
Swiss National Supercomputing Centre
Via Cantonale 2, Manno, CH 6928, Switzerlnad
{alam,hussein,howardk,nstring,fverzell@cscs.ch}

ABSTRACT

Large HPC installations typically make use of parallel file systems that adhere to POSIX I/O conventions, and that implement a separation of data and metadata in order to maintain high performance. File systems such as GPFS and Lustre have evolved to enable an increase in data bandwidth that is primarily achieved by adding more disk drives behind an increasing number of disk controllers. Improvements in metadata performance cannot be achieved by just deploying a large volume of hardware, as the defining characteristics are the number of simultaneous operations that can be carried out and the latency of those operations. For highly scalable applications using parallel I/O libraries, the speed of metadata operations, such as opening a file on thousands of processes, has the potential to become the major bottleneck to improved I/O performance. This Metadata Wall has the ability to grow such that metadata operations can take much longer than the subsequent data operations, even on systems with very large amounts of I/O data bandwidth. We present results showing the performance of metadata operations with standard disk equipment and with solid state storage hardware, and extrapolate whether we expect the evolution in hardware alone will be sufficient to limit the effects of this I/O Metadata Wall. We also report challenges in making the metadata I/O measurements and subsequent analysis for parallel file systems.

Keywords

Parallel file systems, solid state disk, metadata benchmarking.

1. INTRODUCTION

Recent research and development for parallel file system technologies for Petascale supercomputing systems and global file systems for HPC centers have been largely focused on increasing bandwidth for parallel read and write operations for jobs that are running concurrently on the supercomputing platforms. Typically, the file I/O infrastructure on these platforms is shared between multiple jobs, unlike the compute resources which are often allocated in terms of multiple complete nodes dedicated to a single job. Under the constraints of this shared usage pattern it is therefore challenging to present canonical usage models of parallel file systems, especially requirements for metadata generation.

Performance of highly scaling applications that use MPI-I/O or a libraries such as HDF5 or NetCDF that rely on MPI-I/O for parallelism can be tuned to take advantage of the bandwidth available in modern file systems, but metadata operations can form a non-negligible part of the runtime, for example a file needs to be opened before any data can be written. It was shown [1] that the reliance on POSIX I/O semantics in parallel file systems such as Lustre and GPFS reduces scalability in metadata operations, in particular file open times grow linearly with the number of clients needing to participate in the file open. For these file systems a combination of strategies such as deferred file open and node aggregators that are employed in the ROMIO layer [2] of MPI libraries such as MPICH2 could alleviate these scalability problems, but the deferred open strategy needs to be implemented for each file system, and this is not currently the case for Lustre and GPFS file systems thereby leading to linear scaling in file open times for MPI-I/O based libraries. As average core counts on the largest machines continue to increase, and with many applications continuing to rely on message passing with MPI as the sole mode of parallelism to make use of these cores, the linear relationship between file open times and the number of MPI processes needing to open the file leads us to examine the possibility of delivering improved metadata performance through advances in hardware technology.

At the Swiss National Supercomputing Centre (CSCS) [3], we use Lustre [4] as a parallel scratch file system for our flagship, 20-cabinets Cray XT5 system, and this temporary storage file system has been tuned for jobs that have high bandwidth requirements when writing large blocks of data. In addition we have deployed a site-wide accessible, centralised storage facility with a current capacity of 1.4 PetaBytes and using GPFS [5]. This project file system serves as a longer term storage for a diverse range of computing, data analytics, and visualization platforms, including the flagship system, and is also used for staging data analytics and visualization jobs. Setup of CSCS storage resources is depicted in figure 1. On these systems, we observed contention for metadata (create, delete, search, etc.) operations impacting all users, especially on the Lustre file system, which has only a single metadata server. As we upgrade the Cray XT5 system to the Cray XE6 system, the increase in compute capabilities due to over two-folds increase in parallelism (12 cores to 32 cores AMD Opteron) is expected to increase both metadata and application data requirements.

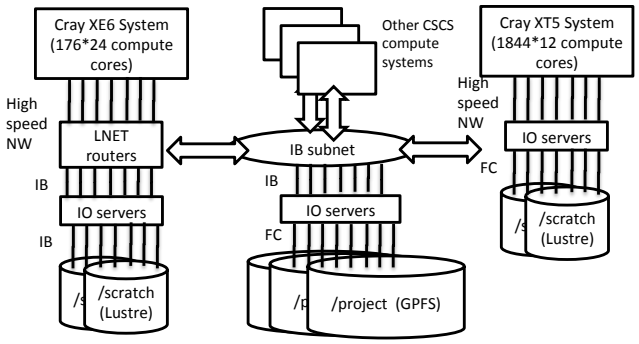

Figure 1: Setup for scratch (supercomputing platform) and site-wide project file systems at CSCS

In addition to increasing the compute capabilities by upgrading the multi-core resources, CSCS is adding additional resources for GPGPU cluster computing that are likely to increase load on the project file system. In the near future we will increase the compute capability of the flagship system that targets the project file system and we will also add two parallel GPU platforms, a Cray XK6 system and a commodity cluster. Although these

systems will each have their own local scratch, for multi-platform experiments where users typically target the project file system we expect an increase in metadata and data volumes and hence the potential for system or site-wide bottlenecks. We therefore undertook this study of characterizing metadata performance on emerging technologies, for example, solid state storage as metadata targets, which offer substantially higher bandwidths and potentially reduced latencies, especially the PCIe and Fibre Channel (FC) attached SSD devices from FusionIO[1], Virident[2], RamSan[3], NetApp[4], and others. Compared to traditional SATA based drives these PCIe and FC connected storage class memories (SCMs) offer significantly higher bandwidth and IOPS rates.

Performance characterization of SSD devices have been presented by many HPC sites and vendors but these are either focused on non-parallel file systems performance or do not distinguish between metadata and data throughput but only look at the sustained IOPS rates [6][7][8][9]. In this paper, we report on our attempts to measure the metadata IOPS for Lustre and GPFS on two PCIe SSD cards using a limited number of clients connected with a QDR InfiniBand (IB) interconnect and report on challenges for undertaking such measurements and subsequent analysis. Note that we have both IB and FC connected network storage for the Lustre and GPFS file systems at CSCS (figure 1).

We extrapolate whether we expect the evolution in hardware alone will be sufficient to limit the effects of this I/O Metadata Wall. With the number of IO clients for typical jobs exceeding thousands of tasks and with many applications requiring frequent checkpoint and restart file IO operations, we expect a considerable increase in the rate of metadata operation requests, a requirement that will be exacerbated for those systems that use GPU devices to increase the rate of computation. Already, there have been instances at CSCS where accidental creation of hundreds of thousands of files by a single user and their subsequent deletion caused severe disruption for all users and their running jobs. Hence, we are interested in evaluating the following scenarios:

- Simultaneous file creations by 80% of compute processes for reading and writing, i.e. directory and file creation rate of 100s to 1000s of thousands of files per second is required.
- Checkpoint and restart behavior, where files are quickly created and removed. File and directory removal rates of order of 10s to 100s of thousands per second are expected.

We report on our experiences where I/O clients are severely limited from achieving theoretical peak IOPS on different devices and also report on issues in measuring metadata IOPS in a consistent manner across two parallel file systems. Two micro-benchmarks, mdtest and metarates, were targeted with directory and file configurations that are representative of our file system usage [10][11]. However, we observed substantial variations across the two networked parallel file systems using an identical setup. We report on our benchmark training process and why tuning was needed to achieve a higher fraction of the SSD peak IOPS for Lustre and GPFS. We also identified that the performance potential of these devices depends highly on the workload, i.e. number of directories and files per directory and anticipate that software level optimization will be needed to address this issue. Although our investigation of GPFS with several metadata configuration options is still a work in progress, we have observed patterns where a combination of hardware and software parameters could yield optimal results. We note minor variations between the PCIe and FC connected metadata targets. Overall, both for Lustre and GPFS, we achieve higher performance using the SSD devices as compared to SATA disk drives for the metadata performance. We also identify challenges in improving metadata performance for Exascale parallel file systems (simulation data, checkpoint-restart, in-situ visualization, etc.) that hardware evolution alone may not be able to resolve.

The paper layout is as follows: in section 2, we describe our experimental setup and the set of micro-benchmarks that are targeted for measuring the metadata performance. In section 3, we report results for the two file systems using our targeted flash drives as well as the reference production systems. We then discuss and analyze our experimental results as well as shortcomings and limits of our current measurement approaches. We also highlight a need for developing standard measurement techniques and benchmarks for metadata operations for networked, parallel file systems. Finally, we summarize the study and list plan of work in the near future.

2. EXPERIMENTAL SETUP

2.1 Hardware

Our experimental configuration is composed of 7 dual-socket Intel Westmere servers and 2 dual-socket AMD Magny-cours servers, a 36-port QDR switch, and SSD devices that are targeted for metadata performance improvements. These include:

- Virident TachIOn SLC NAND, which has a capacity of 400 GB, has PCIe connectivity (8x and 4x), and theoretical peak of 300K IOPS for 4 Kbytes block sizes. It could also deliver 1.44 GB/s (read) and 1.2 GB/s (write) performance.
- One couplet NetApp Pikes peak (E5412) installed in a DE1600 enclosure. We used four SLC SSDs to create two RAID arrays that are exported through channel 0 from both singlets using FC8 ports. 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 are also targeted for Lustre experiments. The hardware configuration does not change for the experiments since the SSD devices are distributed on different servers. For different sets of experiments, Lustre and GPFS are built by mounting the targeted device. For example, we deployed a file system called /scratch with two Intel servers each with PCIe connected Virident cards as GPFS data and metadata targets and /pikes for the single NetApp device as a target.

For the compute systems (file servers and I/O clients), we have a collection of AMD and Intel based platforms. Our Intel systems have dual-socket 6-cores Westmere processors with 24 GB of DDR3-1333 memory. The AMD platforms are dual-socket 8-cores Magny-cours processors also with DDR3-1333 memory. The IB connection is through PCIe 2.0 8x, which offers up to 8 GT/s bandwidth. For experiments, we used OpenMPI that is available with the OFED stack.

[1] http://www.fusionio.com/products/

[2] http://www.virident.com/products/products.php

[3] http://www.ramsan.com/products

[4] http://www.netapp.com/us/products/protocols/san/san.html

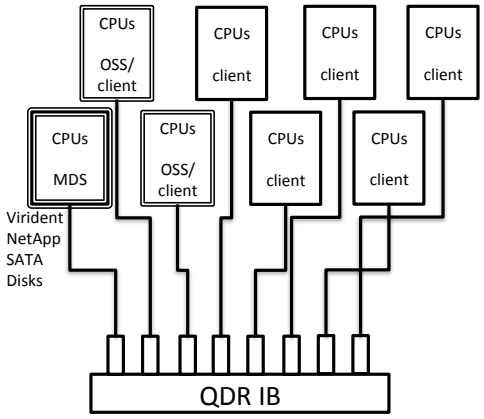

Figure 2: Configuration of Lustre file system with metadata (MDS) and data (OSS) targets

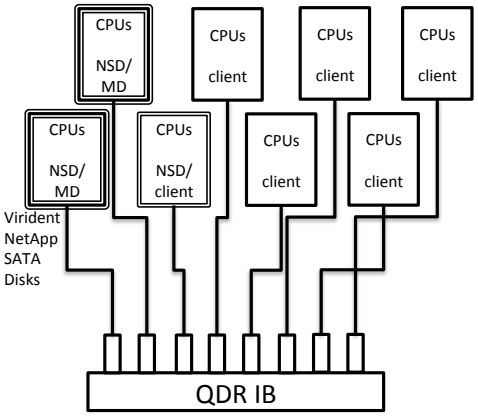

Figure 3: Configuration of GPFS with two metadata (MD) and storage servers (NSDs)

2.2 Parallel file systems: Lustre & GPFS

For experimental evaluation we built Lustre version 1.8.5. The Lustre parallel file system provides a separation of file system activities by having a metadata server (MDS) and metadata target (MDT) for all metadata operations, and a number of object servers (OSS) and targets (OST) for data operations. Lustre currently only allows a single metadata server which is often deployed on physically separate hardware from that used for the object servers that deal with data transfer operations. Whilst we only have one metadata server there are typically multiple object servers and targets in a parallel Lustre file system. For our experiments, we setup the metadata server targeting the three different types of storage: SATA disk drives, PCIe connected SSD and FC SSD. The setup of the Lustre parallel file system used in the experiments is shown in figure 2.

IBM's General Parallel File System (GPFS) is also a clustered file system like Lustre. Unlike Lustre, metadata can be distributed in GPFS and both data and metadata can be striped across multiple network shared disk (NSD) servers and targets. We use GPFS version 3.4.0-7 for our experiments. The GPFS configuration for our experiments is shown in figure 3. Note that, on aggregate, we have a higher bandwidth available for the metadata operations for the GPFS setup as we target multiple SSD devices available on different servers.

2.3 Metadata Micro-benchmarks

For metadata experiments, we wanted to explicitly measure the metadata performance and to have the flexibility of generating workload behaviors that are representative of access patterns, especially the ones that cause congestion and bottlenecks, on the networked, parallel file systems. Most commonly used benchmarks for parallel file systems performance evaluation combine data and metadata operations or can only be run from a single client or by using some scripting solutions [12]. Since nearly all of the parallel applications on CSCS systems have message passing with MPI as the parallel programming paradigm, we chose to use benchmarks that also run in parallel using MPI. The metadata benchmarks used for this study are:

- mdtest: this benchmark explicitly reports the rate of creation and deletion for directories and files as well as stat operations. There are several configuration options, allowing users to choose the number of directories, files per directories, depth of directories, etc. This test does not evaluate MPI-IO performance, instead MPI is only used for launching multiple processes and within each MPI task, POSIX file IO operations are performed.
- metarates: this is primarily a file metadata benchmark, where number of files per MPI task can be specified. This benchmark was mainly used to identify tuning opportunities and to validate mdtest results. Like mdtest, POSIX file IO is implemented in the benchmark.

All experiments have been performed three or more times and the highest performance results have been reported in the subsequent sections.

3. RESULTS AND ANALYSIS

3.1 Benchmark Training

Since our experimental setup is relatively small compared to the sizes and systems for which we are evaluating these technologies, we conducted several experiments to isolate the effects of network latencies and potential buffering at the client side, especially when we oversubscribe clients with a large number of MPI tasks. Figure 4 show a directory and file create operations rate for Lustre when mdtest is called with 16 MPI tasks from the server itself and from a client side, where the MDT is 2 Virident SSD cards. Altogether 134,400 files and directories are created for the experiments with 400 files per directory. Based on the empirical evidence that network latencies could contribute to the metadata throughput on the client side thereby resulting in misleading performance expectations, we decided against including the MDS as a file IO client for Lustre experiments.

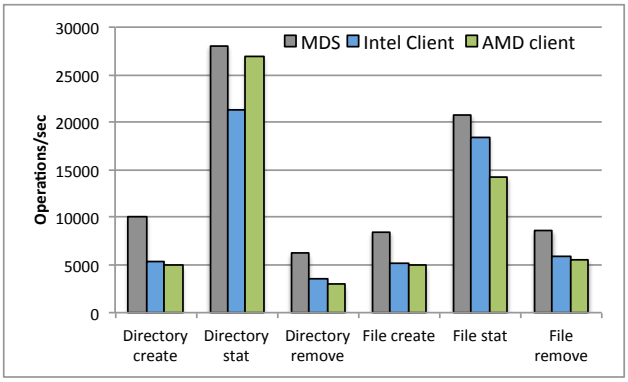

Figure 4: Performance implication of IB network overhead for Metadata operations. Note significantly higher metadata throughput when off-node communication is not involved.

Theoretically, with the given hardware resources we could launch hundreds or thousands of MPI jobs to emulate the behavior of a medium size supercomputing resource but the overloading of local resources could result in much lower than expected performance if the clients are used as non-shared resources as in large-scale supercomputing systems. Figure 5 shows scaling of metadata operations with the number of MPI processes that are launched on available clients (for ~300K file and directory operations). With our existing setup, experiments with 64 MPI tasks yield the optimal client side IOPS rate, therefore in the subsequent sections all results are presented with 64 client tasks. In addition, experiments with 1024 and 2048 files are conducted on the production Cray XT5 system and results are presented in the next section to demonstrate the effect of the metadata wall in our target file systems.

Data in figure 4 and 5 are shown for Lustre setup with Virident cards. With every new hardware setup for both file systems, Lustre and GPFS, we repeat the same sets of experiments to train benchmarks and do not observe significant variations. Network latency impact, in the case of GPFS with distributed metadata servers, is not as significant as in the case of Lustre.

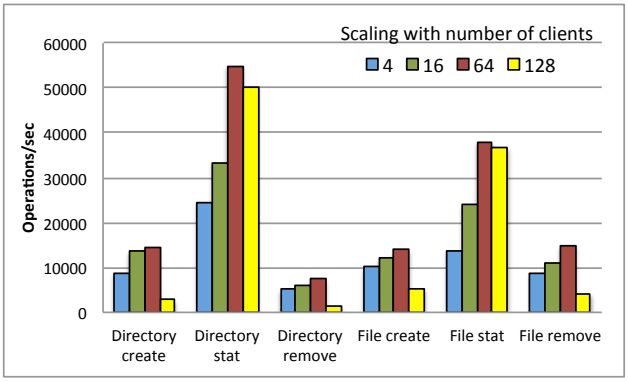

Figure 5: Impact of number of clients on scaling results for the Metadata experiments

The main difference between GPFS and Lustre benchmarking is in tuning benchmark parameters for achieving optimal IOPS. Typically, in scientific applications, when a large number of files are being generated e.g. check-pointing and restart files, there are 100s of files per directory. For Lustre experiments, this configuration yielded a high fraction of the peak while for GPFS, fewer files per directory resulted in a higher fraction of peak. The two benchmarks, mdtest and metarates together, enabled us to identify GPFS tuning potential through a series of experiments. The results presented in the subsequent sections are tuned for better IOPS performance, and thus the input parameters are not consistent between Lustre and GPFS experiments. This is also evidence that with the same hardware, two different parallel file systems with a given file metadata behavior can yield significantly different performance profiles.

3.2 Lustre Metadata Measurements

Figure 6 shows results for experiments using 64 MPI processes and involving a total of 300K files and directories targeting the three different MDTs, PCIe connected Virident, Pikes Peak with FC connection and SATA. Performance for disk based systems, except for the stat operations, are less by a factor of two or more compared to the SSD targets. The two SSD devices have similar performance despite major differences in their technical specification and connection speeds. We do not observe the advantage that the Virident devices offer in terms of IOPS performance from the experimental results. Performance of the stat operation could be attributed to software caching effects.

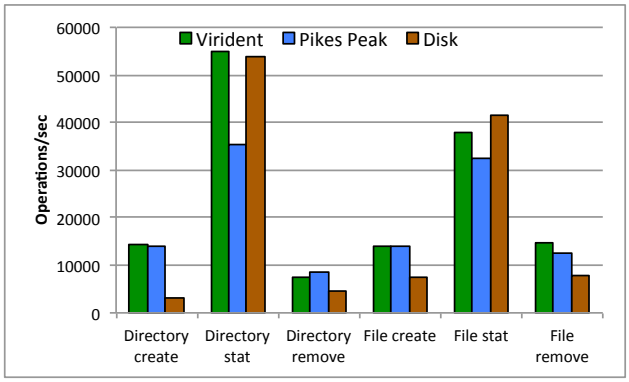

Figure 6: Lustre metadata results on Virident (SSD), Pikes Peak (SSD) and SATA disks as MDT for 64 MPI processors and a total of 300K files and directories.

3.3 GPFS Metadata Measurements

As indicated earlier, GPFS experiments showed significant performance variations for different mdtest input parameters. We measured a very low fraction of the peak performance, significantly lower than Lustre, when using an identical set of parameters as was used in the Lustre experiments (with large number ~100 files per directory). Moreover, our other target benchmark, metarates, consistently showed higher operations/second rates for metadata operations as compared to the mdtest. Therefore, we tuned mdtest to bridge the performance gap. The performance measurements for both un-optimized and tuned versions are shown in figure 7 along with the NetApp Pikes peak (tuned) results. Unlike Lustre, there is noticeable (10-20%) gain for Virident and this could be attributed to the use of multiple metadata servers (two for Virident and one for Pikes peak).

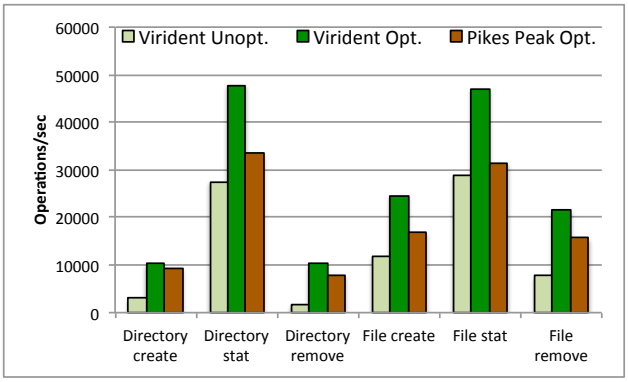

Figure 7: GPFS metadata performance on Viridient and NetApp Pikes Peak. Both tuned and original Lustre parameters results are presented for the Viridient cards (two metadata servers). All experiments are done with 64 MPI tasks and 300K files and directories.

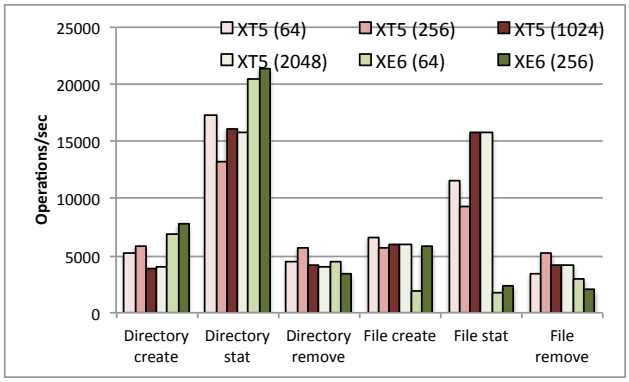

Figure 8: Lustre scratch file system results from a 20-cabinets Cray XT5 and 2-cabinets XE6 platforms. Numbers of MPI processes are shown for each system (MPI tasks). Results are not collected in the dedicated mode.

3.4 Production Platforms Results

The goal of our study is to extrapolate the return of investment in hardware for parallel file systems as the system sizes in terms of the numbers of cores increase and we deploy GPU based clusters. We therefore measured metadata rates on two target systems, a Cray XT5 system and a Cray XE6 system. The Cray XT5 system has an internal Lustre file system while the Cray XE6 has an external Lustre, which is connected through routers (details in figure 1). Both systems use SATA disks as the metadata targets. The storage capacity of the XT5 Lustre file system is ~300 TeraBytes while for the XE6 system, it is ~400 TeraBytes. Figure 8 shows the mdtest scaling measurements while keeping the number of total files and directories constant to 120K. Note that the high-speed network congestion (SeaStarII on the Cray XT5 and Gemini on the Cray XE6) can also contribute to the performance numbers as the experiments are performed when the machines are in production mode with other user jobs running at the same time, and therefore we present the maximum value out of five attempts. Lustre 1.8.4 is installed on the Cray XE6 system and version 1.6.5 is installed on the Cray XT5 system. Except for file stat and create results, we observe a consistent performance behavior across the two systems. One pattern that is rather evident from these experiments is that the metadata performance does not continue scaling with the number of clients. In fact, we observe 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 as well, and in some instances it drops down which could be attributed to the high-speed network congestion as it is shared for communication and IO operations.

4. DISCUSSION

Overall, our observations are:

- 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 have been recorded.
- For a fixed metadata target hardware, the operations/second for directory creation and removal are consistent across both Lustre and GPFS file systems. Tuning resulted in significant performance improvement for GPFS but does not reflect on theoretical specifications.
- Metadata operations are highly dependent on workload characteristics and, in a mixed workload environment, it is rather difficult to argue whether an investment in given hardware is fully justified.
- For the NetApp tests, we used a vendor tool that shows whether or not the metadata tests are saturating the device. In practice however, a larger number of clients can provide us better insights as the metadata performance scales with the number of clients.
- The network setup can also influence the metadata performance as shown for Lustre training experiments. Hence, a balanced storage area network design is also an integral part of a clustered file system solution.

As for the scenarios that we consider for metadata I/O improvements on our target parallel file systems, Lustre and GPFS, we are orders of magnitude away from expected latencies and throughput for file metadata operations. This is an issue since the targeted hardware devices for metadata operations offer a potential for significant speedup but probably the inherent software design limits of the parallel file systems inhibits this performance potential.

Previous work on strategies for improving metadata performance [13][14][15] have demonstrated performance improvements by deploying intelligent software strategies, but these methods are not available in file systems such as Lustre and GPFS that are frequently the only options available on large MPP systems and clusters that are installed at major HPC centers, and therefore when limiting the possibilities to these widely deployed file systems any metadata activity relies on hardware innovations to deliver improved performance.

Hence for Exascale concurrency levels, either the file I/O middleware needs to be developed, for example, SIONLib [16] or applications need to adopt file I/O implementation strategies that do not replicate the scenarios that we considered in this manuscript. Other parallel file systems that we have not considered in this study, for example, Panasas [17] and PVFS [18], may address these issues but we do not find any evidence in the literature survey, where typically the focus is data throughput. Another topic not widely discussed in the literature is strategies and benchmarks for measuring and analyzing metadata performance for networked parallel file systems, for example, benchmarks isolating metadata and data only operations, measuring and reporting impact of both hardware and software middleware, etc. This is particularly important as we consider higher abstraction level IO interfaces such as parallel HDF5 or NetCDF4 that typically rely on MPI-IO implementation and tuning [1][19][20].

5. SUMMARY & FUTURE WORK

We demonstrated the effect of the metadata wall and how technological evolution alone may not be sufficient to address the issue. We also highlighted the challenges in measurement and analysis of parallel file systems performance, as there are several dependencies between the internal high-speed network and the storage area network in addition to parallel file systems middleware. Moreover, the caching that is enabled at the disk controller level could also influence performance measurements, which we are unable to isolate on the client side. Nevertheless, we observe a factor of 2 to 4 improvement targeting the SSD hardware, which is a gain over the currently deployed technology but does not reflect the theoretical capabilities of SSD targets. In the near future, we plan on continuing experiments with a scaled version of our testbed and also using alternate benchmarking schemes with high-level interfaces, for example, MPI-IO and parallel HDF5.

REFERENCES

[1] R. Latham *et. al.* "The Impact of File Systems on MPI-IO Scalability," Lecture Notes in Computer Science, 3241:87-96, September 2004.

[2] Rajeev Thakur *et al.*, "Users Guide for ROMIO: A High-Performance, Portable MPI-IO Implementation", ANL/MCS-TM-234, 2010.

[3] CSCS Cray XT5 and XE6 platforms, http://user.cscs.ch/hardware

[4] Lustre file system, http://wiki.lustre.org

[5] GPFS file system, http://www-03.ibm.com/systems/software/gpfs/

[6] R Freitas, *et. al.* "IBM GPFS Storage Technology Scans 10 Billion Files in 43 Minute," http://www.almaden.ibm.com/storagesystems/resources/GPFS-Violin-white-paper.pdf

[7] N. Master, *et. al.* "Performance Analysis of Commodity and Enterprise Class Flash Devices," 5th Petascale Data Storage Workshop, 2010.

[8] G. M. Shipman, *et. al.* "The Spider Center Wide File System; From Concept to Reality," Cray User Group meeting, 2009.

[9] Milo Polte, *et al.*, "Comparing Performance of Solid State Devices and Mechanical Disks", Petascale Data Storage Workshop, 2008

[10] Metadata benchmark (mdtest), http://sourceforge.net/projects/mdtest/

[11] Metarates benchmark, http://www.cisl.ucar.edu/css/software/metarates/

[12] IOR benchmark, http://sourceforge.net/projects/ior-sio/

[13] Swapnil Patil *et al*, "GIGA+ : Scalable Directories for Shared File Systems", Petascale Data Storage Workshop, 2007

[14] Nawab Ali *et al.*, "Revisiting the Metadata Architecture of Parallel File Systems", Petascale Data Storage Workshop, 2008

[15] Michael P. Kasick *et al.*, "Black-Box Problem Diagnosis in Parallel File Systems", 8th USENIX Conference on File and Storage Technologies (FAST '10), 2010

[16] W. Frings, *et. al.* "Scalable Massively Parallel I/O to Task-Local File," Supercomputing, 2009.

[17] Panasas, http://www.panasas.com/

[18] PVFS file system, http://www.pvfs.org/

[19] HDF5, http://www.hdfgroup.org/HDF5/PHDF5/

[20] NetCDF, http://trac.mcs.anl.gov/projects/parallel-netcdf