Scalability of Implicit LS-DYNA® Simulations Using the Panasas® PanFS® Parallel File System

Bill Loewe
Panasas, Inc.
Sunnyvale, CA

Abstract

This work examines the parallel scalability characteristics of LSTC's Finite Element Analysis software LS-DYNA for up to 288 processing cores for implicit mechanics simulations. This study was conducted on a Linux Intel Xeon cluster with a Panasas PanFS parallel file system and with engineering contributions from LSTC. The motivation for these studies was to quantify the performance and scalability benefits of parallel I/O in FEA software on a parallel file system, comparing with both local storage and conventional NFS for implicit mechanics cases.

Introduction

The parallel efficiency and turnaround time for Computer-Aided Engineering (CAE) simulation continue to be an important factor behind engineering and scientific decisions in developing models at higher fidelity. For their demanding High Performance Computing (HPC) requirements, many parallel CAE simulations use scalable Linux clusters with great success. For certain classes of Finite Element Analysis models (FEA), however, the ever-growing issue of data I/O performance can severely degrade overall FEA job scalability and impact wallclock time.

In looking at this class of models from the perspective of the I/O, a primary high-level difference between explicit and implicit mechanics is the volume of scratch data that is generated and used in the simulation. In this process, the significant data generated in the implicit case can be to the extent that it impacts the overall wallclock time for the simulation, turning the compute-bound problem into an I/O-bound problem. With these models that are heavy in I/O relative to numerical operations, directing all of the results and scratch files to a centralized target has not been a preferred method generally because the overall reduction of I/O performance would impact the work efficiency, effectively leaving idle CPU cycles during the I/O.

As the architecture of HPC systems has evolved over time, the focus has been on the compute moving from monolithic supercomputers with more cost-effective, scalable, and parallel clusters using commodity hardware. HPC clusters enable solving complex engineering problems very quickly by employing multiple concurrent jobs across a cluster of nodes in parallel. While this is particularly suitable for compute-intensive engineering applications, attention must be paid to the I/O as well. To maintain balance with the parallel compute cluster, a parallel storage cluster may be used to maintain throughput and avoid bottlenecks. The massively parallel HPC architecture, then, requires that system components of the compute environment (compute nodes, network, and storage) be balanced. An imbalance in any of these elements can quickly become a bottleneck, impeding the overall performance of the system.

As well, as FEA model sizes grow and the number of processing cores and jobs are increased for the simulation, the I/O operations are more commonly performed in parallel, rather than relying on master thread to collect and operate on each I/O process serially. Just as the parallel solvers require scalable clusters, parallel I/O generated from multiple parallel jobs operates more efficiently with a parallel file system. The parallel file system design and capabilities enable the application to overcome I/O bottlenecks, enabling I/O-bound FEA simulations to scale to their full potential.

In order to increase productivity and maximize Return-On-Investment (ROI), these massively parallel HPC clusters can be designed with a distributed, parallel file system, such as the Panasas PanFS parallel file system, to minimize the I/O bottleneck and maintain a balanced system. The objective of this whitepaper is to demonstrate the performance gains when using a parallel file system such as PanFS when running the LS-DYNA application in an HPC cluster over Panasas ActiveStor® appliances.

System Configuration

The test configuration used for this study was a cluster of 288 cores, provided by 24 SuperMicro servers each with dual 6-core Westmere (2.67GHz) Xeon CPUs and 24GB of RAM per node. Each node ran CentOS 6.2 (2.6.32-220) and was connected via 10GbE NICs to a Force10 S4810 interconnect. The networking for the cluster and storage were configured with jumbo frames set to MTU=9000. On the clients, networking was configured with the tcp_min (Retransmission Timeout) setting reduced from the default of 200ms to 15ms.

For local storage, each of the 24 Linux nodes had a 2TB SATA (7200RPM) local disk with EXT3 configured. Though a compute node may have more disks in a local hardware configuration, for this testing, however, such resources were not available.

For parallel storage, 2 Panasas ActiveStor 14T shelves were used running PanFS 5.0.1. For the PanFS parallel file system, a cluster of object-based storage devices provide true parallelism as data access is balanced across the storage cluster from the compute node cluster. The compute nodes had the Panasas DirectFlow® 5.0.1 client software installed with a single global mount point for the storage system. We expect to see the maximum performance and scalability with this solution.

For NFS access, two NFS servers running NFSv3 were provided. The server blades were part of a 2U quad chassis, with each server running a Sandy-Bridge-based Xeon at 3.20GHz (E5-1650) with 64GB of RAM and using a dual-port 10GbE PCI Express NIC for connectivity. Compute nodes were mounted to either of the two NFS servers for a balanced load from the cluster to this storage. The backend for the NFS servers was a reexport of the Panasas file system to provide a sufficient number of disks (40 in total) for comparison.

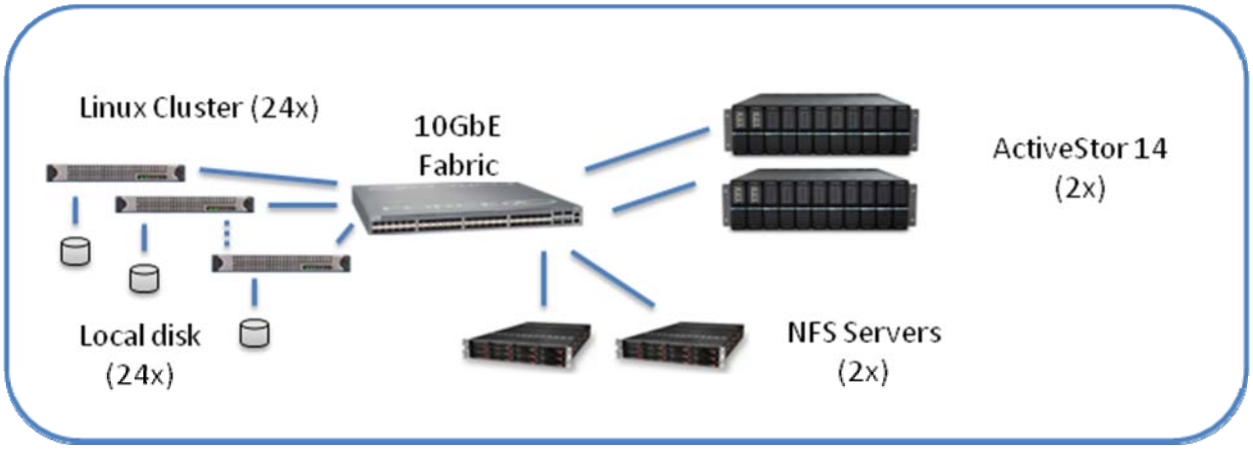

Figure 1: Diagram of System Configuration

Benchmark Performance

For the initial benchmark testing, the cyl0p5e6 benchmark available from LSTC benchmarking website was used to demonstrate the high I/O demands of the implicit mechanics simulations. This benchmark model is originally from a collection of test problems created by AWE in the UK consisting of a series of co-axial cylinders. These nested cylinders are fixed at their bases with pressure and motion applied during the simulation. For this size of model, there are 500K solid elements.

The LS-DYNA 6.1.1 implicit benchmark was run with a single job per 12-core node using OpenMPI. Each of the jobs was run concurrently on the nodes, scaling from 1 to 24 jobs corresponding to 1 to 24 nodes. In order to determine the relative performance of the three types of storage, the overall wallclock time was used as the metric as the non-I/O runtime would be otherwise consistent between the tests with the scratch and d3plot files accessing PanFS, NFS, or local disk.

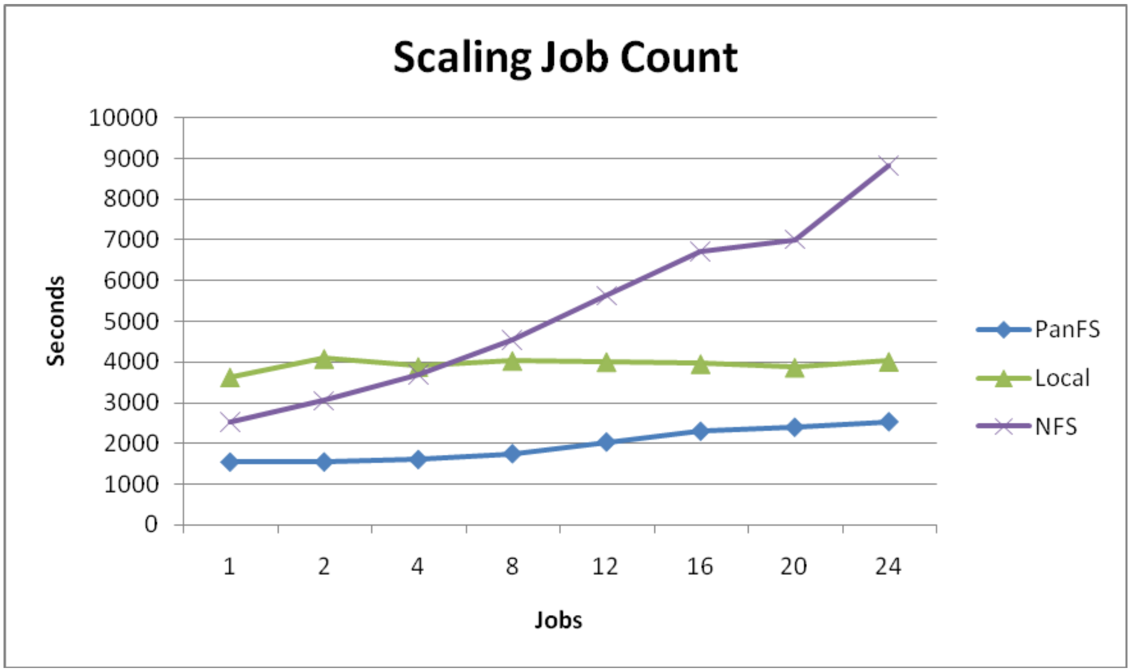

Figure 2: Scaling Job Count, cyl0p5e6

In Figure 2, the results from increasing job count show a consistent flat line for the local disk performance. This is to be expected as these are concurrent jobs that are not competing for shared network or remote storage resources. This line is the bar against which the other storage should be measured to see if there is a performance advantage.

With the NFS jobs, this is actually faster than the local disk initially with only a few nodes, but the slope shows with increasing jobs that the NFS servers become the bottleneck. By 24 jobs, this is sufficiently slow to have more than doubled the wallclock time of the corresponding 24-job test on local disk. While having a centralized pool of storage via NFS offers certain usability conveniences in terms of locality and availability, the performance impact makes it clear why this approach to storage is not commonly used.

For the case with the PanFS jobs, however, the wallclock time is fairly flat, but does increase from 1537 seconds with a single job to 2530 seconds with 24 jobs. This is due to the use of the fixed-source of the parallel file system being increasingly taxed with additional jobs. But, this demonstrates the viability and effectiveness of using the parallel file system for the storage during the run, particularly as it can outperform the local disk in this experiment, offer the advantage of globally-available storage, and still maintain only a modest performance impact when heavily pushed by an increasing job count. As well, it outlines the potential for maintaining consistent strong performance by using a balanced system with proportional compute nodes and storage.

From these scaling tests, it becomes clear that with the PanFS parallel file system even for large numbers of jobs the wallclock time is 63% of the time as that of local disk. This, in turn, yields a commensurate amount of additional work capacity for the LS-DYNA jobs.

Additional benchmarks were then conducted on a larger calculation using the cyl1e6 benchmark as seen in Figure 3 to determine if the advantage of the parallel file system held. This benchmark is twice the size of the cyl0p5e6 at 920K solid elements. For this test, each job was restricted to 6 cores per node to provide sufficient memory per core, and again scaling 1-24 nodes with 1-24 jobs using LS-DYNA 6.1.1. Implicit.

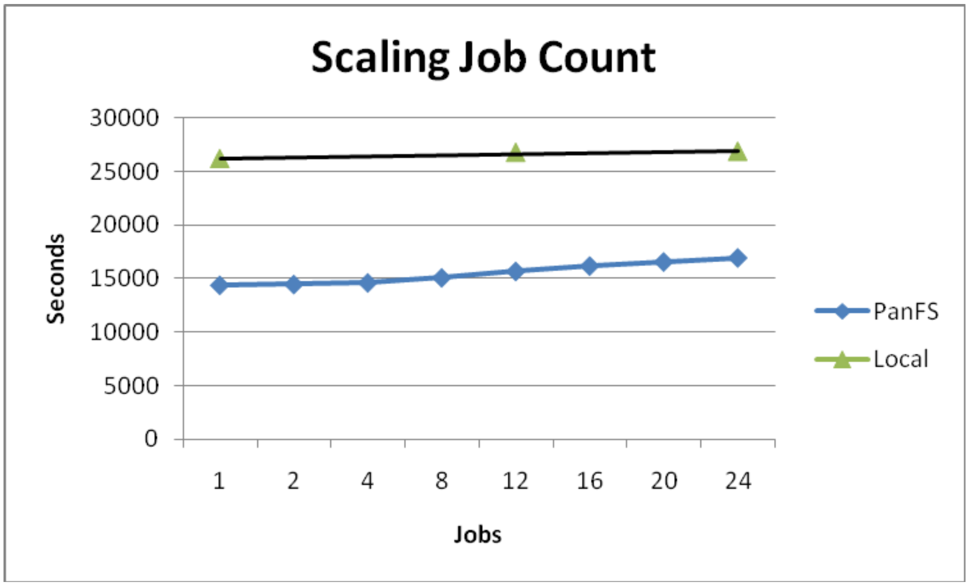

Figure 3: Scaling Job Count, cyl1e6

The cyl1e6 tests were conducted with the local disk and PanFS, and these show the same characteristics as the cyl0p5e6 dataset. The local disk again remains constant as expected, and PanFS shows a gradually-increasing slope with greater job count. Essentially this test was conducted to confirm that the larger dataset still exhibits the expected behavior in comparing the local disk and PanFS. In this case, the relative proportion of wallclock time between the different storage is maintained, with the larger dataset taking a longer time to complete as expected. For the cyl1e6 case, this was not run on NFS as this had already been demonstrated to not show scalable performance using the smaller cyl0p5e6 dataset.

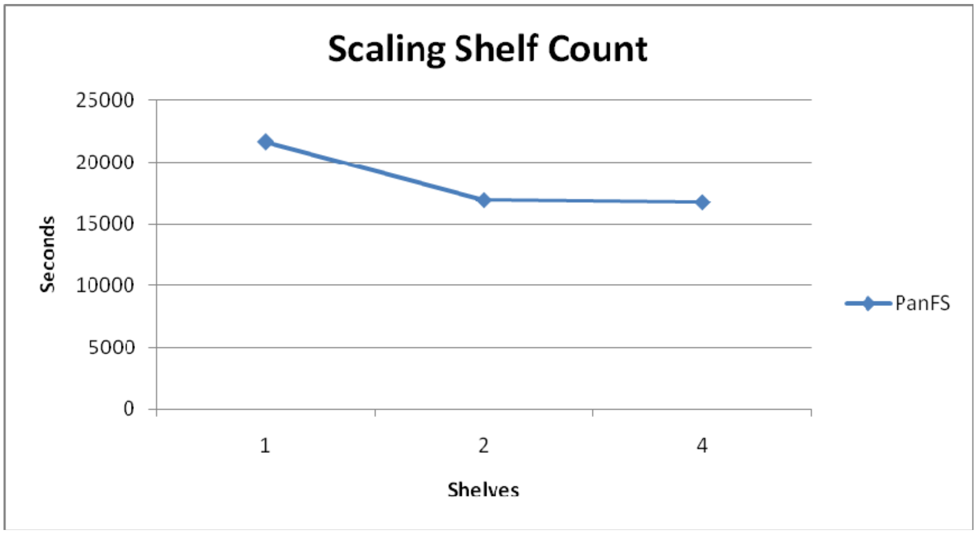

Figure 4: Scaling Shelf Count, cyl1e6

Further, to determine the saturation point for the Panasas AS14T ActiveStor shelves, a single set of 24 jobs (on 24 nodes) was run against 1, 2, and 4 shelves as seen in Figure 4. For the single shelf case, the wallclock time for the 24 jobs was 21644 seconds. Increasing the number of storage shelves to 2 reduced this bottleneck to wallclock 16909 seconds for the 24 jobs. But in adding 2 additional shelves for a total of 4 shelves, the wallclock time was 16737 seconds. This shows the balance of nodes and shelves for this test case, but also implies that 48 nodes would be balanced by 4 shelves to avoid a storage bottleneck. With increasing the nodes and shelves proportionally, the scalable nature of the system would be maintained.

As to why this implicit I/O workload shows such advantage on the PanFS parallel file system, an analysis of the data access pattern yields some insight. For these models, the traffic to the scratch files is performed in large, contiguous 256KB I/O calls to large files on the order of 2-10 GB in size. This is primarily sequential I/O which allows for the gains from client side caching and generally avoids read-modify-write patterns throughput the I/O path. With each process accessing its own, unique file there is neither coherency nor locking overhead to contend with. And while the write and read access to each of these scratch files is by its nature interspersed at times, this is relatively infrequent and instead the pattern shows distinct sections of contiguous and similar operations. Finally, for what random offsets are necessary, these are less than 10% of access pattern and aligned on the large, 256KB boundaries to further minimize unaligned I/O and corresponding read-modify-write overhead. In short, the I/O design for LS-DYNA is well-considered and is essentially optimal for use in a large, high-performing parallel file system.

Advantages of a parallel file system

This attention to providing efficient I/O interaction with the underlying file system allows for greater utilization of the LS-DYNA license, moving from a higher percentage of I/O operations to more numerical operations. The high efficiency of a parallel I/O solution equates to as much as two times more LS-DYNA utilization that can be achieved with the same license as on local disk or NFS.

With the globally-accessible file system, the post-processing advantage as well becomes clear. Moving away from a model in which users are forced to leave data in-place or copied-off reduces the overall work time as the move phase is not necessary. Rather than use a large compute node to run a simulation and then effectively idle it while moving the generated data to a centralized storage location, the large node may be further used in running additional simulations. With the globally-visible data, the post-processing can be performed in-place, optionally using alternate nodes than the compute nodes.

Further, the ability to review in-flight workloads is made easier by opening a consistent file share and visualizing actively running results. This in turn reduces the time to validate and correct the model, being able to end an invalid run earlier rather than waiting for completion before resubmitting.

For these reasons of high performance and improved data accessibility, the combination of scalable CAE application software, Linux HPC clusters, high speed network interconnects and a parallel file system for storage, can provide significant performance advantages for implicit FEA simulations.

Panasas PanFS Parallel File System

In recent years, a new approach to parallel file systems and shared storage technologies has come forth with a scalable I/O design aiming at extending the overall scalability of CAE simulations on compute clusters. The objective of these innovative storage architectures has been to combine key advantages of existing legacy shared storage systems, but without the bottlenecks and design limitations that have made them unsuccessful for large distributed HPC cluster systems.

Parallel Network-Attached-Storage can achieve both the high-performance benefits of direct access to disk, as well as the data-sharing benefits of files and metadata that HPC clusters require for CAE performance scalability. The Panasas implementation of Scale-Out NAS provides this technology with an object-based storage architecture to eliminate serial I/O bottlenecks. Inherent in object-based storage are two primary technological breakthroughs that extend storage capabilities beyond that of conventional block-based storage.

For the first advance, a file is comprised of virtual objects distributed across storage physical storage components. As each object is a self-contained set of information maintaining a combination of user data and metadata attributes, the object architecture is able to offload I/O directly to the storage device instead of going through a central file server or provide an interface to a block-based backend. To deliver parallel I/O capability, the client I/O work is spread evenly across the virtual objects in the storage architecture allowing for true parallelism from the clients to the storage devices.

With the second, since each object has metadata attributes in addition to user-data, the object can be managed intelligently within large shared volumes under a single namespace. These volumes may be managed by multiple metadata manager devices for scaling of the global namespace.

The inherent scalability of object-based storage architectures provides virtually unlimited growth in bandwidth and capacity, making them ideal for addressing CAE's high I/O demands for simulation, post-processing, and data management. With this design, the cluster has parallel and direct access to all data spread across the object-based storage system. The implication of this use of a scalable storage cluster is that a large volume of data can be accessed for either computation and visualization to improve the simulation turnaround time.

The Panasas PanFS parallel file system has been developed as the premier storage system for scalable Linux clusters. Built on an object-based storage clustering architecture and an innovative parallel file system, the Panasas storage system delivers the performance scalability, high storage capacity, and appliance-like ease of management necessary for CAE manufacturing and for workloads with high-I/O requirements in particular.

Conclusion

This study demonstrates that the LSTC LS-DYNA software using the Panasas PanFS parallel file system can greatly improve job completion times for implicit simulations that are heavy in I/O operations relative to numerical operations.

For this implicit workload, the Panasas PanFS parallel file system increases productivity and efficient use of cluster resources, offering up to 2 times more aggregate I/O throughput to the LSTC LS-DYNA user as compared to a local disk solution. This enables engineers to run more jobs in the same time frame, speeding time to results and providing increased ROI.

References

[1] LS-DYNA User's Manual Version 971, Livermore Software Technology Corporation, Livermore, CA, 2010.
[2] Gibson, G.A., R. Van Meter, "Network Attached Storage Architecture," Comm. of the ACM, Vol. 43, No 11, November, 2000.