Novel HPC Technologies for Scalable CAE: The Case for Parallel I/O and File Systems

Stan Posey
Panasas, Inc., Fremont, CA, USA
510-608-4383, sposey@panasas.com

Abstract

As HPC continues its aggressive platform migration from proprietary supercomputers and Unix servers to HPC clusters, expectations grow for clusters to meet the I/O demands of increasing fidelity in CAE modeling and data management in the CAE workflow. Cluster deployments have increased as organizations seek ways to cost-effectively grow compute resources for CAE applications, and during this migration many also implemented conventional network attached storage (NAS) architectures to simplify IT administration and further reduce costs.

While legacy NAS implementations offer several advantages of shared file systems, most are too limited in scalability for effective management of I/O demands with parallel CAE applications. As such, a new storage migration is underway to replace legacy (serial) NAS with parallel NAS architectures and parallel file systems. This new class of parallel file system and shared storage technology was developed to scale I/O in order to extend the overall scalability of CAE simulations on clusters.

This paper examines CAE motivation for shared parallel file systems and storage, for requirements of multi-physics LS-DYNA® applications on conventional clusters with proper balance for I/O. Model parameters such as size, element types, schemes of implicit and explicit (and coupled), and a variety of simulation conditions can produce a wide range of computational behavior and I/O data management demands. The benefits of a Panasas storage implementation are introduced for such broad requirements, through examples of CAE workflows for a variety of production-level applications in industry.

Introduction

Manufacturing industry and research organizations continue to increase their investments in structural analysis and impact simulations such that the growing number of LS-DYNA® users continues to demand more from HPC resources. These LS-DYNA workload demands typically include rapid single job turnaround and multi-job throughput capability for users with diverse application requirements in a high-performance computing (HPC) hardware and software infrastructure.

Additional HPC complexities arise for many LS-DYNA environments with the growth of multidiscipline CAE coupling of structural and CFD analyses, that all compete for the same HPC resources. Such requirements also drive I/O levels that prevent most system architecture’s ability to scale. Yet for today’s economics of HPC, the requirements of CPU cycles, large memory, system bandwidth and scalability, I/O, and file and data management – must be satisfied with high levels of productivity from conventional systems based on scalable, inexpensive clusters.

In order to manage the extreme I/O demands, entirely new storage system and software architectures have been introduced that combine key advantages of legacy shared storage, yet eliminate the drawbacks that have made them unsuitable for large distributed cluster deployments. Parallel NAS can achieve both the high-performance benefits of direct access to disk, as well as data-sharing benefits of files and metadata, that Linux clusters require for CAE scalability. That is, just as a cluster distributes computational work evenly across compute nodes, parallel NAS storage distributes data evenly across a shared file system for parallel data access directly between distributed cluster nodes and NAS disks.

As the number of compute cores are increased for single CAE simulations, in order to keep pace with fidelity and model growth, I/O operations should be performed in parallel to realize the essential benefits of overall simulation scalability. With a Panasas storage approach, each node on a cluster has direct access to read and write data on the shared storage and parallel file system, in order to maximize I/O performance during the computation phase of a CAE simulation. Once the simulation is complete, the same shared storage provides an end-user with direct access to the CAE results files for subsequent post-processing and visualization of the CAE simulation.

This paper examines HPC workload efficiencies for sample multidiscipline LS-DYNA applications on a conventional HPC Linux platform with proper balance for I/O treatment. Model parameters such as size, element types, schemes of implicit and explicit (and coupled), and a variety of simulation conditions can produce a wide range of computational behavior and I/O management requirements. Consideration must be given to how HPC resources are configured and deployed, in order to satisfy growing LS-DYNA user requirements for increased fidelity from multidiscipline CAE.

HPC Characteristics of LS-DYNA®

Finite element analysis software LS-DYNA™ from Livermore Software Technology Corporation (www.lstc.com) is a multi-purpose structural and fluid analysis software for high-transient, short duration structural dynamics, and other multi-physics applications. Considered one the most advanced nonlinear finite element programs available today, LS-DYNA has proved an invaluable simulation tool for industry and research organizations who develop products for automotive, aerospace, power-generation, consumer products, and defense applications, among others.

Sample LS-DYNA simulations in the automotive industry include vehicle crash and rollover, airbag deployment and occupant response. For the aerospace industry, LS-DYNA provides simulations of bird impact on airframes and engines and turbine rotor burst containment, among others. Additional complexities arise from simulations of these classes since they often require predictions of surface contact and penetration, models of loading and material behavior, and accurate failure assessment.

From a hardware and software algorithm perspective, there are roughly three types of LS-DYNA simulation characteristics to consider: implicit and explicit FEA for structural mechanics, and computational fluid dynamics (CFD) for fluid mechanics. Each discipline and associated algorithms have their inherent complexities with regards to efficiency and parallel performance, and also regarding modeling parameters.

The range of behaviors for the three disciplines that are addressed with LS-DYNA simulations, highlights the importance of a balanced HPC system architecture. For example, implicit FEA using direct solvers for static load conditions, requires a fast processor and a high-bandwidth I/O subsystem for effective simulation turnaround times, and is in contrast to dynamic response, which requires very high rates of memory and I/O bandwidth with processor speed as a secondary concern. In addition, FEA modeling parameters such as the size, the type of elements, and the load condition of interest all affect the execution behavior of implicit and explicit FEA applications.

Explicit FEA benefits from a combination of fast processors for the required element force calculations, and memory bandwidth for efficient contact resolution that is required for nearly every structural impact simulation. CFD also requires a balance of memory bandwidth and fast processors, but benefits most from parallel scalability. Each discipline has inherent complexities with regard to efficient parallel scaling, depending upon the particular parallel scheme of choice. In addition, the I/O associated with result-file checkpoint writes for both disciplines, and increasing data-save-frequency by users, must also scale for overall simulation scalability.

Implementations of both shared memory parallel (SMP) and distributed memory parallel (DMP) have been developed for LS-DYNA. The SMP version exhibits moderate parallel efficiency and can be used with SMP computer systems only while the DMP version, exhibits very good parallel efficiency. This DMP approach is based on domain decomposition with a message passing interface (MPI) for communication between domain partitions, and is available for homogenous compute environments such as SMP systems or clusters.

Most parallel CAE software employ a similar DMP implementation based on domain decomposition with MPI. This method divides the solution domain into multiple partitions of roughly equal size in terms of required computational work. Each partition is solved on an independent processor core, with information transferred between partitions through explicit message passing in order to maintain the coherency of the global solution. LS-DYNA is carefully designed to avoid major sources of parallel inefficiencies, whereby communication overhead is minimized and proper load balance is achieved. In all cases the ability to scale I/O during the computation is critical to overall scalability in a simulation.

A New Generation of Parallel Storage

Currently, there are two types of network storage systems, each distinguished by its command sets. First is the SCSI block I/O command set, used by storage area networks (SAN), which provides high random I/O and data throughput performance via direct access to the data at the level of the disk drive or fibre channel. NAS systems use protocols such as NFS or CIFS command sets for accessing data with the benefit that multiple nodes can access the data as the metadata (describes where the data exists) on the media is shared. To achieve the high-performance and data-sharing benefits that Linux clusters can provide requires a fundamentally new storage design, one that can offer both the performance benefits of direct access to disk and the easy administration provided by shared files and metadata. That new storage design is an object-based storage architecture.

Object storage offers virtually unlimited growth in capacity and bandwidth, making it well-suited for handling large results-data generated by LS-DYNA simulations on Linux clusters. Unlike conventional storage systems, data is managed as large virtual objects. An object is a combination of application (file) data and storage attributes (metadata) that define the data. Managing data as objects, as opposed to traditional storage blocks, means that files can be divided into separate pieces. Such object storage blocks are then distributed across storage media known as object-based storage devices (OSDs). So just as the Linux clusters spread the work evenly across compute nodes for parallel processing, the object-based storage architecture allows data to be spread across OSDs for parallel access. It is massively parallel processing on the front end cluster, matched by massively parallel storage on the back end.

Such an architecture delivers substantial benefits to a distributed LS-DYNA application. By separating the control path from the data path, file system and metadata management capabilities are moved away from the path to the nodes in the Linux cluster, and provide nodes with direct access to storage devices. By doing so, OSDs autonomously serve data to end-users and radically improve data throughput by creating parallel data paths. Instead of pushing all information (data and metadata) through one path, which creates major bottlenecks as data size and number of nodes increase, Linux cluster nodes can securely read and write data objects in parallel to all OSDs in the storage cluster system.

With object-based storage, the Linux compute cluster has parallel and direct access to all of the data spread across the OSDs within the shared storage. The large volume of data is therefore accessed in one simple step by the Linux cluster for computation. While the simulation and visualization data may still need processing for weeks at a time, the object model of storage drastically improves the amount, speed, and movement of data between storage and compute clusters.

Panasas Parallel Storage Technology

The Panasas parallel storage system has been designed to provide the benefits of a NAS parallel file system. The Panasas parallel file system (PanFS) allows for a single namespace that can be shared across the entire pool of storage and load-balanced dynamically. This system provides the foundation for a single, shared storage infrastructure that can be fully leveraged and utilized for CAE applications.

The key differentiation for Panasas from other RAID storage solutions lies in the software architecture. Panasas is leading the development of object-based storage. The core principle of this approach is that data is managed in large virtual objects and not as small blocks or files. This allows for parallel communications and I/O between cluster compute nodes and storage, eliminating the bottlenecks that invariably come with traditional storage architectures.

The Panasas architecture divides files into objects, which are logical units of storage and can be accessed with file-like methods such as open, close, read, and write. This approach is especially effective for large files. Objects have associated application data, attributes, and metadata. Each object is designed to be managed, grown, shrunk, deleted, and dynamically distributed across physical media. As these objects are distributed across storage devices, they can be accessed in parallel by the cluster compute nodes. Meanwhile, the management of the objects is done by a metadata manager. The object attributes allow the system to offload work from the traditional filer head or file server to the storage device.

Object storage enables two primary technological breakthroughs. First, since the system is able to offload work directly to the storage device instead of going through a central filer head or file server, the system is able to deliver parallel performance directly from disk. Secondly, since each object is injected with attributes as well as application data, it can be managed intelligently.

This architecture allows Panasas to scale with performance that grows with capacity. The scaling of performance with capacity is almost linear; Panasas does this with an architecture that uses finely tuned hardware components to optimize the software architecture's capabilities. Panasas DirectorBlades serve as a 'virtual filer' to scale and manage metadata growth and Panasas StorageBlades act as smart disk drives to scale and manage capacity growth.

Computational Performance of LS-DYNA

Performance and parallel efficiency of any CAE software has certain algorithm considerations that must be addressed. The fundamental issues behind parallel algorithm design are well understood and described in various research publications. For grid-based problems such as the numerical solution of partial differential equations, there are five main sources of overhead that can degrade ideal parallel performance: 1) non-optimal algorithm overhead, 2) system software overhead, 3) computational load imbalance, 4) communication overhead, and 5) I/O operations.

Parallel efficiency for LS-DYNA is dependent upon among others, MPI latency, which is determined by both the specifics of a system architecture and the implementation of MPI for that system. Since system architecture latency is determined by design of a particular interconnect, overall latency improvements can only be made to the MPI implementation. Modifications to the MPI software to ensure "awareness" of a specific architecture are a way to reduce the total latency and subsequently the communication overhead. For certain applications parallel efficiency is also greatly affected by the ability to scale I/O operations. Parallel computations require parallel I/O in some cases, in order to scale the overall simulation.

Specifically for structural FEA simulations in LS-DYNA, they often contain a mix of materials and finite elements that can exhibit substantial variations in computational expense, which may create load-balance complexities. The ability to efficiently scale to a large number of processors is highly sensitive to load balance quality of computations and I/O. For example, the crash worthiness of automotive vehicles exhibit these characteristics, and especially as models begin to approach 10 MM elements. Similarly, an aerospace application for design of gas turbine engines for aircraft, has utilized the parallel scalability of LS-DYNA to reduce the time it requires to complete a 5M-element model for blade-out simulation. There is a growing desire to couple both implicit and explicit schemes in such blade-out simulations which requires intermediate I/O to scale in order to scale the overall simulation.

Additional developments between LSTC and Panasas applications engineering include an enhanced I/O scheme that significantly improves overall model turnaround in a mix of LS-DYNA jobs in an HPC production environment. This development is particularly important in production environments that include other applications and disciplines such as NVH and CFD that might request similar HPC resources as LS-DYNA during a multi-job throughput workload.

Summary and Conclusions

A review was provided on the HPC resource requirements of various LS-DYNA applications, including characterizations of the performance behavior typical of LS-DYNA simulations on distributed memory clusters. Effective implementation of highly parallel LS-DYNA simulations must consider a number of features such as parallel algorithm design, system software performance issues, hardware communication architectures, and I/O design in the application software and file system.

Development of increased parallel capability will continue on both application software and hardware fronts to enable FEA modeling at increasingly higher resolutions. Examples of LS-DYNA simulations demonstrate the possibilities for highly efficient parallel scaling on HPC clusters in combination with the Panasas parallel file system and storage.

LSTC and Panasas continue to develop software and hardware performance improvements, enhanced features and capabilities, and greater parallel scalability to accelerate the overall solution process and workflow of LS-DYNA simulations. This alliance will continue to improve FEA modeling practices in research and industry and provide advancements for a complete range of engineering applications.

References

1. Gibson, G.A., R. Van Meter, "Network Attached Storage Architecture," Comm. of the ACM, Vol. 43, No 11, November, 2000.

2. LS-DYNA User's Manual Version 971, Livermore Software Technology Corporation, Livermore, 2005.

3. IOzone Filesystem Benchmark

4. A. M. David Nagle, Denis Serenvi. The Panasas ActiveScale Storage Cluster – Delivering Scalable High Bandwidth Storage. In *Proceedings of Supercomputing 04*, November 2004.

5. M. DeBergalis, P. Corbett, S. Kleiman, A. Lent, D. Noveck, T. Talpey, and M. Wittle. The Direct Access File System. In *Proceedings of Second USENIX Conference.*