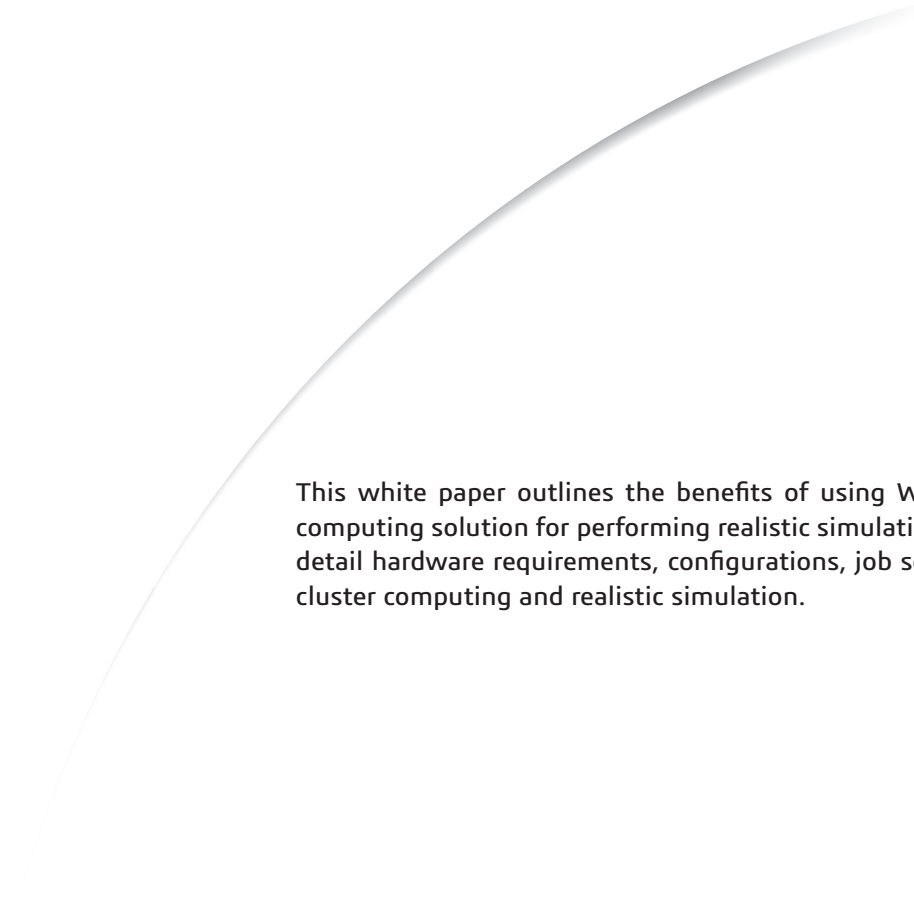


Leveraging Windows HPC Server for Cluster Computing with Abaqus FEA





This white paper outlines the benefits of using Windows HPC Server as part of a cluster computing solution for performing realistic simulation. It is one of an ongoing series that will detail hardware requirements, configurations, job scheduling, and benchmarks pertaining to cluster computing and realistic simulation.

Contents

I. Introduction	3
II. HPC Hardware	3
III. Abaqus Parallel FEA	4
IV. Cost and Benefits of Parallel Execution	5
V. Microsoft Windows HPC Server Configuration	7
VI. Summary	8

I. Introduction

The trends toward higher-fidelity modeling and increasing product complexity are lengthening execution times for many simulation users. At the same time, the advent of multicore chips and inexpensive computing clusters has made parallel computing far more affordable than in the past. Parallel execution is primarily considered in situations where simulation times lead to a serious bottleneck, or where model fidelity is restricted to keep run-times at a reasonable level. The benchmarks provided in this white paper, however, demonstrate that parallel computing can not only save time, but also reduce simulation costs for the vast majority of jobs.

Using Abaqus finite element analysis (FEA) software on a Windows-based multicore workstation or cluster provides an integrated, efficient, and easily scalable high-performance computing (HPC) platform. This hardware/software configuration can be used to solve some of the most complex simulation and multiphysics analysis equations in less time.

Microsoft Windows High Performance Computing Server 2008 provides an integrated platform that makes it much easier than before to implement and administer parallel computing platforms.

This white paper will help FEA users determine whether cluster computing makes sense for them. It also will provide an overview of the most critical factors to consider when implementing an HPC platform/configuration.

II. HPC Hardware

There are three primary types of computers commonly used for running FEA jobs. The first is a low-end workstation, which typically includes two or four cores, two or four gigabytes of RAM, and a single hard disk. The second option is a high-end workstation with eight cores or more, 16 gigabytes or more of RAM, and multiple disks that are striped, or simultaneously read from and written to, for greater performance. The third is a compute cluster, which is a group of individual machines connected by a high-speed network.

The hardware needs of an analyst depend on the type and number of simulations that person typically performs. The analyst may be examining structures under loads that can be simulated at relatively low computational cost, making a low-end workstation sufficient for their needs. However, many analysts may find that accurate and efficient simulation requires compute power beyond what the low-end, or even high-end, workstation can deliver. In addition, greater automation of simulation workflows may require even greater computational capabilities. An analyst building a model with minimal help from automatic tools can often get good results at a lower computational cost just by applying engineering insight to the creation of a model. But as computing costs decrease, the analyst's time becomes more valuable than computer time; so increasingly, greater computational cost (hardware/software) is accepted for increased automation.

Ten years ago, analysts running simulations that demanded more capability than a single workstation provided were required to move to expensive proprietary architectures, such as the Cray vector supercomputers or sophisticated UNIX RISC servers. Beginning in the mid-1990s, the industry shifted towards supercomputers comprised of commodity processors and components. As a result of this development, today's low-end workstations use the same processors as cluster supercomputers. The difference between low-end and high-end machines is that current supercomputers harness multiple processors (from tens to thousands) to work on a single problem. This technique is commonly referred to as parallel computing.

In recent years, the acceptance of parallel computing has been accelerated by the arrival of multi-core computers in which multiple processing units, or cores, are integrated on a single die. Computer processor designers and manufacturers continue to pack more computational cores into each generation of machines, expanding the compute power available to users of parallel codes.

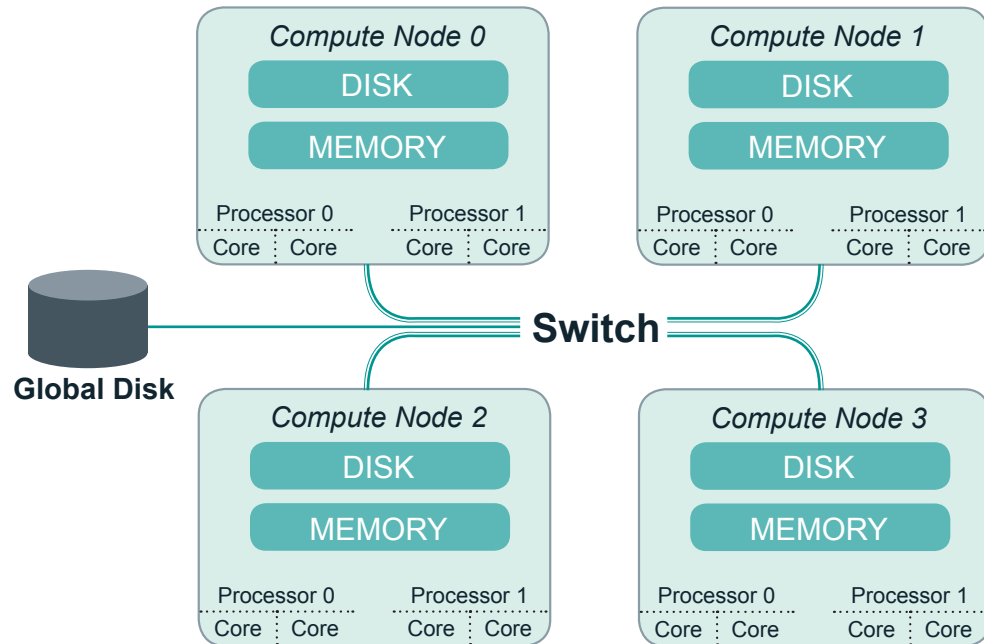


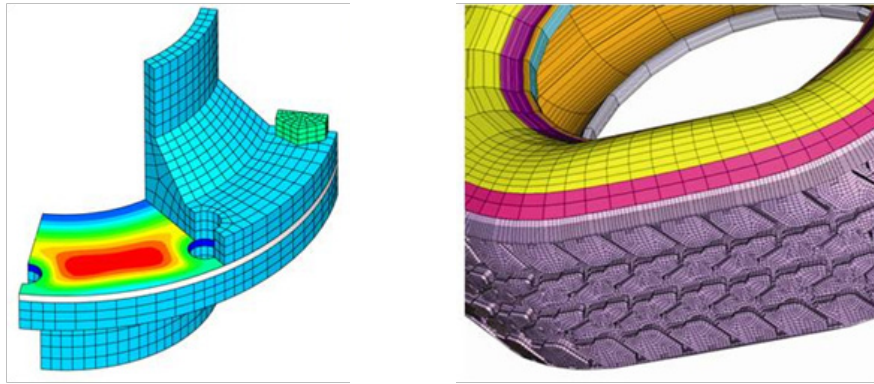
Figure 1. A typical four-node compute cluster.

Figure 1 illustrates a modern “cluster” computer consisting of four individual machines (or nodes). Each node has two processors, or sockets, and each processor has two computational cores. This configuration is called a 2P/dual-core node. Another popular design is a 2P/quadr-core node in which each processor has four cores. As shown in the diagram, each node has its own memory, and usually its own disk. The memory within the node is shared by its cores.

The nodes in Figure 1 are all connected by a high-performance switch that creates a private network, or interconnect, which allows the nodes in the cluster to share data rapidly. A high-speed interconnect is the component that distinguishes a modern compute cluster from a loosely grouped set of machines. The most common high-speed networks used with HPC setups are Infiniband, Gigabit Ethernet, 10 gigabit Ethernet, Myrinet, and Quadrics networks are also regularly employed.

III. Abaqus Parallel FEA

The Abaqus FEA product suite, comprised of Abaqus/Standard and Abaqus/Explicit, provides comprehensive and robust parallel applications that will scale effectively from the workstation to a cluster. Abaqus/Standard employs solution technology ideal for static and low-speed dynamic events in which highly accurate stress solutions are critically important. Examples include sealing pressure in a gasket joint, steady-state rolling of a tire, and crack propagation in an aircraft fuselage. Abaqus/Explicit technology, on the other hand, is well-suited for high-speed dynamic events, such as consumer products drop testing, automotive crashworthiness, and ballistic impact.



Contact pressure on pipe flange gasket (left); Full tire model showing deformations in the tire footprint during steady-state rolling (right)

Both Abaqus analysis applications are coded to effectively scale from low core count workstations to clusters with 64 to 128 cores using a single code installation. Many codes, such as Abaqus, require no special version to switch from a workstation to a cluster.

Abaqus users running smaller models with 500K, or fewer, degrees of freedom (DOF) will likely achieve satisfactory turnaround times for their simulations using two or four cores on a workstation. For smaller models, run-times will not be significantly improved by dedicating more cores to a single job. In many cases, though, users are forced to restrict the fidelity of their models because their hardware is inadequate. In such cases they should consider employing a more powerful workstation or a cluster to obtain better simulation results. This is now feasible since the cost of more powerful compute capability is often low enough to be offset by the improvement in engineering fidelity.

Users running models with 500 thousand to 2 million DOF should consider using four to eight cores on a high-end workstation or a node of a compute cluster to handle their problems. With such complex models, moving from a single core to eight cores on a 2P/quad-core machine will typically increase simulation speed as much as 5X and reduce costs from 50 to 66 percent (see “Cost and Benefits of Parallel Execution”). In such cases, it is important to have sufficient memory and disk space to support the cores being used.

For models larger than 2 million DOF, users will generally benefit from employing a compute cluster. The cluster can effectively handle the simulation requirements of these models, delivering an accurate solution with manageable turnaround times.

IV. Cost and Benefits of Parallel Execution

In the current business climate, decisions to move to parallel processing need to demonstrate return on investment (ROI) either through cost savings, incremental revenues due to higher product performance, reduced time-to-market, or all of the above. The common perception that parallel processing offers faster execution at a higher cost is no longer true. Multicore chips and cluster computing have made parallel execution more affordable than before. Integrated systems such as Microsoft’s out-of-the-box HPC tool have made it easier and less expensive to implement and administer parallel solutions. The old model in which software license costs increased linearly with the number of processors has given way to a new token-based licensing model in which the software cost for parallel execution is often less than for single-core execution.

Parallel processing is making a transition from a highly specialized mode of computing to a common method of solving scientific problems. In recent years, parallel processing has also

become a necessary method used to simply reduce cost through increased throughput and efficiency when solving larger and more complicated problems.

Determining the cost of parallel execution requires a collective analysis of three main contributors:

1. **Software cost per unit hour:** This is based on the number of tokens used.
2. **Hardware cost per unit hour:** This includes the initial cost of the hardware, expected lifespan of the hardware, hardware utilization ratio, maintenance and administration, computer room space, power consumption, cooling costs, etc.
3. **Simulation time:** Runtime normally decreases as the number of cores increases. The actual speed-up, however, varies depending upon characteristics of the models (as will be discussed later). The best approach is to run benchmarks for representative problems to determine the runtime for different numbers of cores.

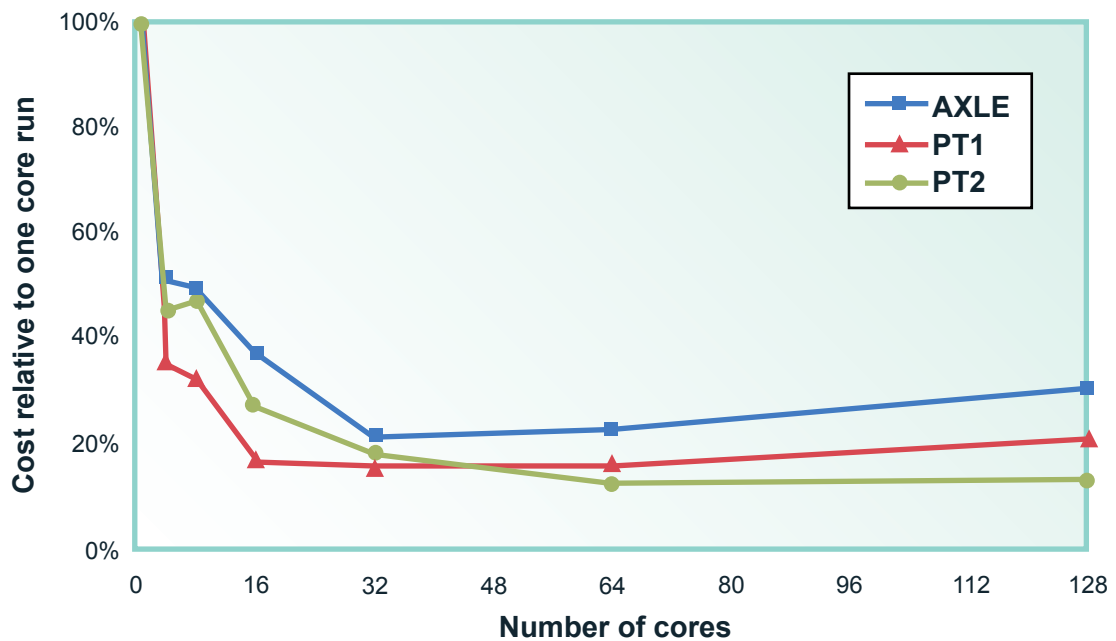


Figure 2: An example of benchmarking the cost of running three models with different numbers of cores

The chart in Figure 2 shows cost curves obtained from three models ranging from 5-9 million DOF. Note the steep drop in cost that occurs when moving from one to multiple cores. While the number of Abaqus tokens required increases with the number of cores used, the increase in cost is typically less than the run-time speed-up.

For example, when moving from one to eight cores executing on a single machine, the computation cost for all three models drops by at least 50 percent. For the PT2 model, running on 128 cores versus a single machine reduces the cost 87 percent and cuts the time 99 percent.

The best balance of cost and speed, or “sweet spot”, for these larger models occurs at 64 cores. The minimum cost for running smaller models generally happens at a lower numbers of cores, although the lowest cost rarely occurs at a single core (even for very small models). As the number of cores is increased past the sweet spot, costs begin to increase again. This happens

primarily because the rate of analysis speed-up decreases as cores are added. On the other hand, cost curves typically remain flat well beyond the “sweet spot”, making additional hardware investment small and the life of the analyst fighting to meet deadlines that much easier. It is also important to note that the costs of hardware and software together are small compared to design and production labor expenditures.

V. Microsoft Windows HPC Server Configuration

When “commodity off-the-shelf” clusters first emerged in the mid 1990s, configuring them was a task for an expert. In recent years, compute clusters have become much easier to manage and some hardware/software vendors have specialized in making clusters accessible to a wider user-base. Until the past two years, the dominant operating system (OS) in cluster computing was the Linux OS. This presented a barrier to many organizations whose IT expertise centered around Windows workstations. Now, as a result of the wide availability of Microsoft IT skills and the investment Microsoft has made in their Windows HPC Server 2008 solution, cluster computing is easy to use and accessible to almost any organization.

Microsoft Windows HPC Server simplifies the deployment, configuration, and management of compute clusters, while also integrating multiple performance improvements. The interface provides a simple and effective way to increase cluster administrator productivity, incorporating wizard-based configuration tools, computer node templates, node monitoring and management methods, integrated diagnostic and reporting utilities, and the Windows PowerShell command line shell.

More specifically, administrators can use wizards to perform many initial configuration tasks. When the application is launched for the first time, for example, it displays a to-do list screen showing the available wizards. After the cluster is configured, additional cluster management tasks (e.g., node management, job management, diagnostics) can be performed using the charts and reports panes of HPC Cluster Manager.

Windows HPC Server 2008 has capabilities that are comparable to Linux and provides a compelling alternative for companies that want to implement high-performance computing with existing Windows IT resources.

		Total Elapsed Time (secs)			
Nodes	Total Cores	Linux RHEL5		Windows HPC Server 2008	
1	8	19970		21230	
2	16	11257	1.8	9214	2.3
4	32	4535	4.4	4923	4.3
8	64	3038	6.6	3392	6.3
16	128	2816	7.1	2759	7.7
32	256	2260	8.8	2658	8.0

Figure 3: Comparison of Linux and Windows HPC operating systems with an FEA problem. The speed-up column (green) represents the increased speed factor as compared to an 8 core scenario.

Figure 3 compares the performance of Windows HPC Server 2008 to Linux RHEL5 for a single step consisting of a nine-iteration simulation using Abaqus 6.8-3. The example illustrates the performance of both Windows and Linux up to 256 cores.

VI. Summary

The information presented in this white paper demonstrates that parallel execution can substantially reduce simulation costs for a high percentage of simulation projects. It also provides a number of guidelines and examples useful in determining the benefits that can be obtained from parallel execution.

As a general rule, relatively small problems can be effectively executed in parallel on a single multi-core machine, while medium- and large-size problems are most efficiently executed on a compute cluster. The benefits are greatest in large, high-iteration models, but overall simulation costs can be reduced for all but the smallest models. Running specific benchmark models on a number of different machines is the best way to determine the optimum parallel execution configuration.

**SIMULIA**

Rising Sun Mills
166 Valley Street
Providence, RI 02909-2499
Tel: (401) 276-4400
Fax: (401) 276-4408
E-mail: Simulia.info@3ds.com

About SIMULIA

SIMULIA is the Dassault Systèmes brand that delivers a scalable portfolio of Realistic Simulation solutions including the Abaqus product suite for Unified Finite Element Analysis, multiphysics solutions for insight into challenging engineering problems, and SIMULIA SLM for managing simulation data, processes, and intellectual property. By building on established technology, respected quality, and superior customer service, SIMULIA makes realistic simulation an integral business practice that improves product performance, reduces physical prototypes, and drives innovation. Headquartered in Providence, RI, USA, SIMULIA provides sales, services, and support through a global network of regional offices and distributors. For more information, visit <http://www.simulia.com>.

Dassault Systèmes Americas

900 Chelmsford Street
Tower 2, 5th Floor
Lowell, MA 01851
United States
Tel: +1 800 382 3342

About Dassault Systèmes

As a world leader in 3D and Product Lifecycle Management (PLM) solutions, Dassault Systèmes brings value to more than 115,000 customers in 80 countries. A pioneer in the 3D software market since 1981, Dassault Systèmes develops and markets PLM application software and services that support industrial processes and provide a 3D vision of the entire lifecycle of products from conception to maintenance to recycling. The Dassault Systèmes portfolio consists of CATIA for designing the virtual product - SolidWorks for 3D mechanical design - DELMIA for virtual production - SIMULIA for virtual testing - ENOVIA for global collaborative lifecycle management, and 3DVIA for online 3D lifelike experiences. Dassault Systèmes' shares are listed on Euronext Paris (#13065, DSY.PA) and Dassault Systèmes' ADRs may be traded on the US Over-The-Counter (OTC) market (DAST4). For more information, visit <http://www.3ds.com>.

©Dassault Systèmes 2011, All Rights Reserved. CATIA, DELMIA, ENOVIA, SIMULIA, SolidWorks and 3D VIA are registered trademarks of Dassault Systèmes or its subsidiaries in the US and/or other countries.

