2011 SIMULIA Customer Conference, May 17-19, 2011, Barcelona, Spain

High Performance Computing and CAE within a modern Engineering Environment

Frank Popielas, Rohit Ramkumar, Blake Garretson, Brad Jones

Dana Holding Corporation, USA

Abstract: The state of the current automotive or heavy duty industries is changing rapidly, especially with the recent recession still in mind or to some extend still looming. In such an environment the competitiveness of a company is tested on every step. Engineering-based companies have to re-define their strategies, what means engineering to them.

With the emerging nations catching up in basic technologies very fast and very competitive and even pushing in certain areas the technology further forward it becomes for the traditional countries for the automotive and heavy duty industries essential to make a change in their engineering approach. This first of all means to change from a manufacturing-driven to an engineering-driven business.

Further more, it means that engineering needs to be driven by simulation – CAE (computer aided engineering). This encompasses all engineering disciplines and not just the traditional CAE analyst's environment. The complex environment of such dimensions requires the right resources and capacities in order to provide this service in a highly competitive manner and quality.

Dana recognized early in the game the need for HPC (High performance computing) to support this effort. This paper will discuss the advantages of HPC in an engineering environment focus around CAE, provide examples and outlook into the future HPC / CAE driven engineering environment.

Keywords: HPC, CAE, engineering environment, automotive industry, heavy duty industry, competitive environment, System Engineering

1. Introduction

The state of the current market is defined by recovering from a deep recession. Companies which have survived this recession are still not guaranteed to survive in the long run. We have to understand the dynamics of the market and parameters involved that drive it. Even though there are certain common dominators across all industries, this paper focuses on engineering companies with a particular interest in the automotive and heavy duty industries.

Both industries have specific drivers. While the automotive industry is more consumer-driven, the heavy duty industry is driven by the demand in construction. Traditionally, this means the automotive industry is primarily price driven, while the heavy duty industry is mainly focused on the performance of their vehicles during operating conditions that can be challenging. But, does this really still hold true in our current environment?

Only providing the lowest cost product is not the final solution anymore. If this truly is the case products would be simply outsourced to the so-called "low cost countries" and manufactured there. In a global economy in a highly competitive market the following areas need to be balanced:

- Low cost
- Top level manufacturing
- High quality
- Low warranty returns
- Global availability

Low cost is easily doable, but by focusing only on cost, the product would become a commodity in the industry. Even the companies currently leading in their particular areas will eventually be exposed to this danger. Major new product innovations are becoming rare. Smaller incremental innovations have become more common. While incremental innovations help support the company, it may not help the company's long term viability.

From a long-term perspective, those companies need to change from a manufacturing driven company to an engineering driven company. The engineering driven companies are where the major innovations are happening now and in the future. These companies are where IP (intellectual property) is being generated on a large scale. Terminology linked to this topic includes:

- 24x7 engineering
- Outsourcing
- Simulation driven design
- Virtual testing
- Virtual manufacturing
- System engineering.

From our experience and from others in the industry, we know physical testing is very expensive due to the time and equipment costs. However the investment costs into the "virtual" environment can also be very large. Only the companies that are able to keep these costs under control will survive. The companies that had the basics of these engineering-driven processes in place before the recession started are now leading the innovation and the market. Dana Holding Corporation is a very good example of this (4.1).

Dana Holding Corporation participates in several markets:

- Light Vehicle
- Commercial Vehicle
- Off-Highway
- Service Parts

These market segments are serviced by these business units (Figure 1):

- Drivetrain products

- Sealing products
- Thermal management products
- Tire management products
- Transmission and controls

Figure 1: Dana Holding Corporations - Products and Market Segments

All those products are represented and supported by strong brand names (Figure 2):

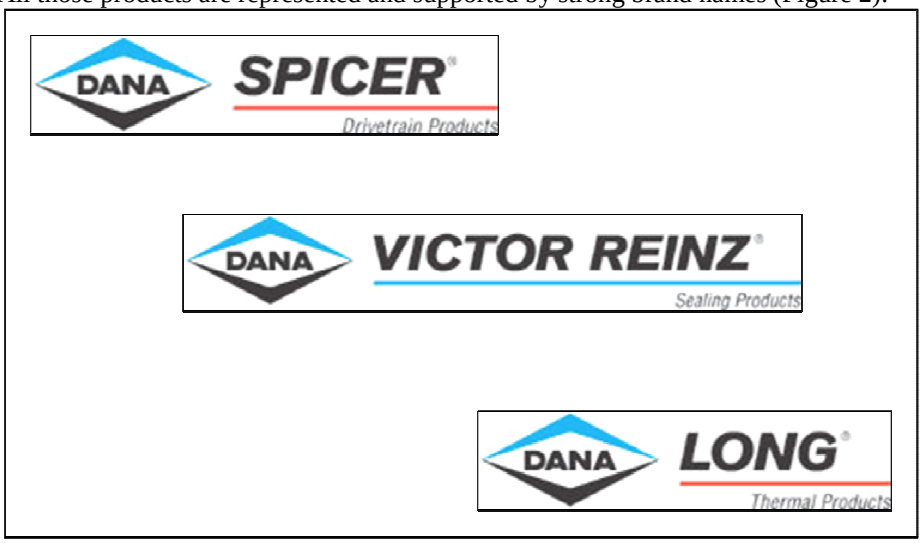

Figure 2: Dana Holding Corporation - Brand names

The products shown above demonstrate the product complexity to which Dana Holding Corporation is exposed. During the design process multiple engineering departments rely and depend on our CAE engineering department for ways to improve these products.

Historically, in the industry and at Dana, CAE was done in a simplified fashion. We were forced to solve analyses with simplified models and linear equations based on the limits of the software and computer hardware. The problem, of course, is that real world engineering problems do not follow simple assumptions and linear equations. They require more complex modeling techniques and non-linear equations. Without advancements in software and hardware technology we would be missing some of the failure modes needed to accurately design and develop our products. We must capture all of the possible failure modes that occur in our lab and in the field. In previous papers and publications we have discussed in order to accomplish this, a simple component simulation is not sufficient anymore. Most of the time, we evaluate sub-systems or complete systems in order to accurately capture failures as they occur in lab environment and in the field.

The simple use of CAD data, even if we consider the entire system, is not good enough anymore. Basic manufacturing steps are already integrated into the simulation process before the functionality simulation takes place. In the future the whole manufacturing process needs to be captured up-front and, thus, its influence on the product and its functionality will be captured accurately.

Our goal is to create a virtual manufacturing, testing, and optimization environment while improving the timing and quality of our product. Our intent is to not necessarily replace the lab testing, but to help make design decisions that will reduce the number of design iterations. Final validation of the product will still always happen in a perfect engineering environment.

Areas that we focus on to try to achieve system engineering are (4.2):

- Supporting Activities, like SLM
- Software, like Abaqus, iSight, Hypermesh,
- HPC (High Performance Computing)

Let us break these down further to understand the background of each one. One of the biggest advances in predicting the failure modes at Dana was the use of non-linear FEA software such as Abaqus. Nature seldom acts simply linearly. Most of our products are interacting through contact with other components which immediately makes the problem non-linear. Sub-system and system simulation further magnify the need for the right CAE software to support simulation activities. Abaqus was eventually chosen as the main FEA code for Dana (4.1).

The right choice of the CAE software is only one site of the equation. The hardware system has to be optimized for our application and the software used as well. Not only is this a requirement in order to be able to run our models, it is necessary also to ensure a competitive engineering environment from timing and cost perspectives.

In the past, FEA software packages were written mainly for SMP (Shared Memory Processing) architectures, while CFD codes were targeting DMP (Distributed Memory Processing) architecture in a very early stage due to the much larger model sizes. FEA codes are usually more memory-intensive compared to CFD codes and more of a challenge to break up into portions that can be solved in a DMP environment effectively. On the other hand, in the past the hardware technology was not sufficiently advanced to provide enough CPU power and sufficient memory for that environment at the same time. Finally, just several years ago, solutions emerged on the market where efficient DMP environments became available from both a hardware and software point of view.

2. High Performance Computing and CAE within a modern Engineering Environment

High Performance Computing (HPC) – this is the new buzz-word in computing, in particular for engineering driven companies. Simply put, HPC is a cluster of computers that work together. But just the hardware itself does not make it a high performing system. The system needs to be optimized for the specific needs of the end user and for the specific application. Therefore, "just a bunch of computers" put into a cluster is not sufficient anymore. The system needs to be balanced and the nodes used across any analysis should be uniform in order to ensure maximum efficiency and utilization of resources.

The primary measure for the performance of a cluster system is the so-called sweet spot. The sweet spot is defined by the following (4.3):

- Model (application)
- Simulation type (FEA, CFD, …)
- CAE software scalability
- Software and hardware architecture
- Cost structure of software and hardware.

Basically, the sweet spot for a specific application is the lowest possible cost/performance ratio.

Since Dana Holding Corporation has a wide variety of products, we identified benchmark models covering the major simulation types and applications (Figure 3). Using those models we evaluate new hardware systems and, of-course, also CAE software. They are the basis to establish the sweet spots for our products.

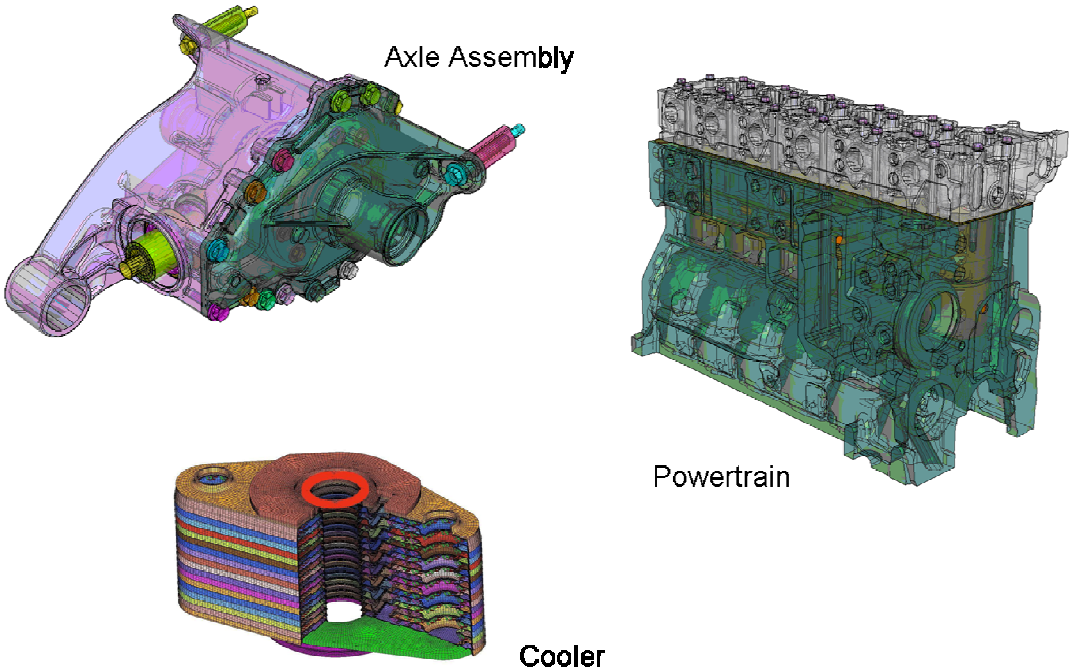

Figure 3: Dana CAE Benchmark Models

A typical basic HPC structure is shown in Figure 4. The different components of such a system are:

- Analyst's workstation
- Cluster:
    - Head node
    - Compute nodes

    - Interconnect
- San storage device
- Queuing system.

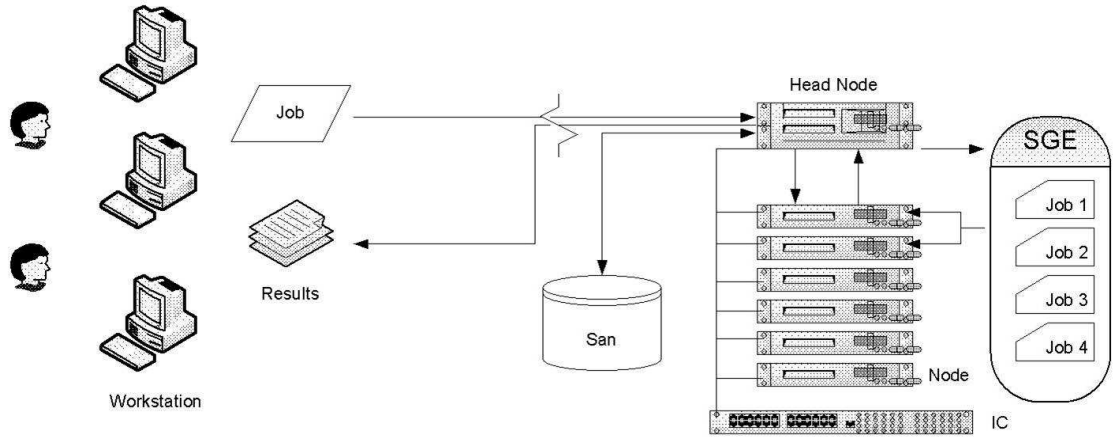

Figure 4: HPC Architecture

The analyst prepares the simulation job on a workstation. For solving the simulation, he submits the job to the cluster's head node. There the system is managed by a queuing system. In our case we are using SGE (Sun Grid Engine). SGE holds the job until the CPU cores requested by the analyst become available. The analyst specifies the number of cores for the specific simulation based on the sweet spot for the particular application.

The compute nodes used for a job have all the same specifications:

- CPU type
- Memory
- Disc space

The nodes' local disks are used only for temporary working space. No data is permanently stored on them. After the job is completed, the files to be saved are copied to the SAN (Storage Area Network) for storage and post-processing. All temporary files on the node are then deleted.

The memory on each node is optimized for the jobs to be run on those nodes. We have different partitions on our systems. The more challenging and detailed jobs, such as jobs with a large number of degrees of freedom (DOF) and system simulations require that we specify the nodes with more memory. For lighter weight jobs, the clients with less memory are sufficient. The interconnect is the channel connecting the different nodes. It has to have enough bandwidth to allow the huge amount of data to be transferred back and forth during a simulation.

As mentioned before, SGE is the "manager" of the cluster and sits on the head node. In order to make life for an analyst easier, there are usually different queues defined in the system. Those are defined based on the different types of jobs to be run on the cluster and the sweet spot analysis for the applications. We usually have 4 different queues defined:

- Thick client
- Thin client
- Benchmark runs
- Development runs.

Before we implement a new compute system throughout the company we go through a benchmark study with the major HPC vendors using our benchmark models as shown before (Figure 3). Those are evaluated

in cooperation with our major CAE software providers. In our case, we use Abaqus for structural simulation, thus, we work very closely with SIMULIA.. The axle and powertrain benchmark models are detailed below in Table 1. The powertrain is represented by two models with the major difference being the number of DOFs. The PT2 model represents the trend in the near future for such models. This is already the standard today for system simulation.

	Axle	Powertrain 1 (PT1)	Powertrain 2 (PT2)
Num. Elements	1.20E+06	8.75E+05	1.94E+06
Num. Nodes	1.92E+06	1.50E+06	3.12E+06
Num. Equations (DoFs)	9.43E+06	5.28E+06	9.30E+06
FLOPS/iteration	3.32E+13	2.57E+13	1.60E+14
Memory to minimize I/O on one node (GB)	71.8	44.3	128.3
Steps	10	3	8
Increments	10	14	34
Iterations	65	47	154

Table 1: Benchmark models details

During the initial HPC evaluation we used the following system configuration (Table 2).

Processor:	Intel® Xeon® Quad-core CPU Model E6462 @ 2.80GHz
Node Configuration:	2 socket compute nodes with 8 cores/node
Memory:	32GB/node
Number of Nodes:	32
Interconnect:	InfiniBand (Quad Data Rate)
Disks:	One 250GB SATA drive on each node
Operating System:	Redhat Linux Enterprise Server 5.1

Table 2: Benchmark system details

The results of the evaluation are shown in Table 3. The values in the grayed out cells are estimated based on the time increments since it would have not been practical to wait for job to actually complete. Figure 5 shows the scalability of the models in reference to an eight-core run.

		Number of cores						
		1	4	8	16	32	64	128
AXLE	Total Wall (hrs)	107.9	50.3	42.7	20.7	7.5	4.5	3.4
	Avg. Iter. (min)	98.8	45.6	38.6	18.0	6.1	3.6	2.6
	Iter. Contrib.	–	98%	98%	94%	89%	88%	84%
PT1	Total Wall (hrs)	49.6	16.2	12.8	4.3	2.4	1.4	1.1
	Avg. Iter. (min)	63.1	20.4	16.1	5.2	2.9	1.6	1.1
	Iter. Contrib.	–	99%	98%	95%	94%	90%	84%
PT2	Total Wall (hrs)	967.3	394.6	360.1	128.4	51.3	19.6	11.3
	Avg. Iter. (min)	376.6	153.4	140.0	49.7	19.8	7.5	4.3
	Iter. Contrib.	–	–	–	99%	99%	98%	97%

Table 3: Benchmark performance data

Earlier in this paper it was mentioned that based on the model there are different memory requirements. During this evaluation, we compared the PT1 and PT2 models directly with each other. The information we have learned holds true with all the different applications. More DOFs requires more memory. The main outcome of this study shows that we were able to accurately specify how much memory is needed for our applications. Figure 6 and Figure 7 show the different requirements for the PT1 and PT2 benchmark models respectively.

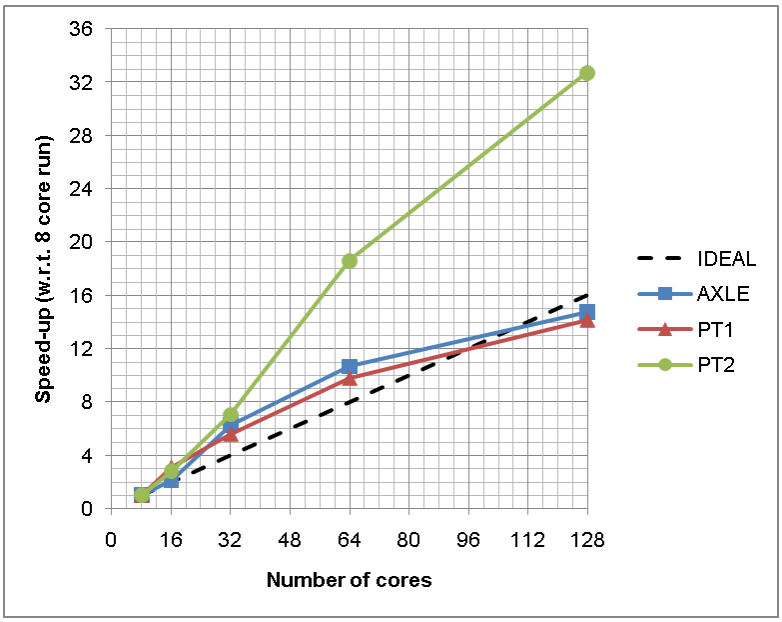

Figure 5: Scalability data based on iteration time

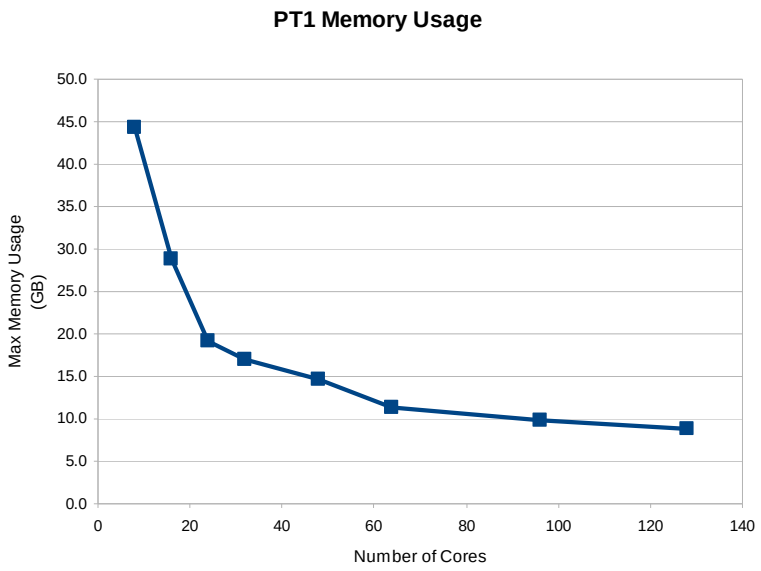

Figure 6: memory requirement for PT1 benchmark model

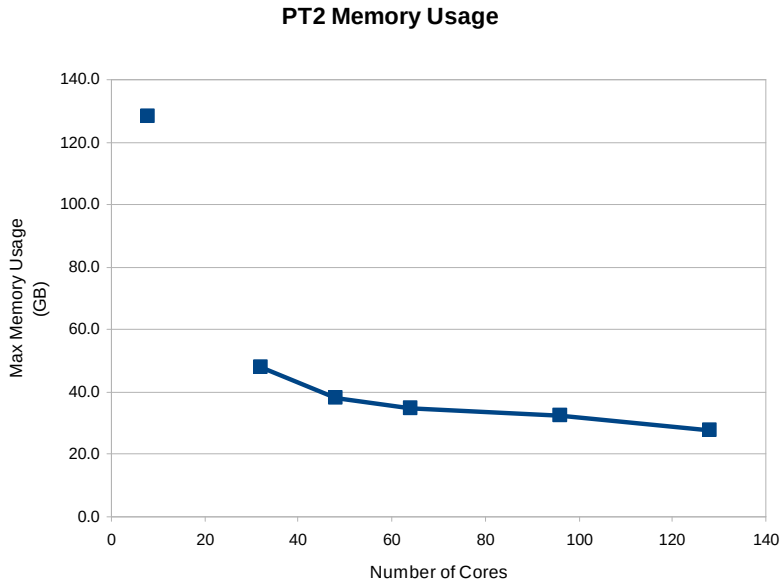

Figure 7: Memory requirement for PT2 benchmark model

The results from those studies defined our thin and thick clients for our clusters. As implemented our thin clients have a minimum of 48GB per node.

In addition to technical goals, we are also able to improve hardware utilization and efficiency by transitioning to clusters. The major factor in defining how we load the new clusters effectively is, as mentioned before, the sweet spot. Figure 8 shows the results of that study. One interesting point from this study is that the curve towards more cores is relatively flat. This indicates that if we are under enormous time pressure to complete a job we can add more cores to the simulation with no and little cost penalty. The results show that our sweet spots lie around 32 to 64 cores. Our queues were defined based on this outcome.

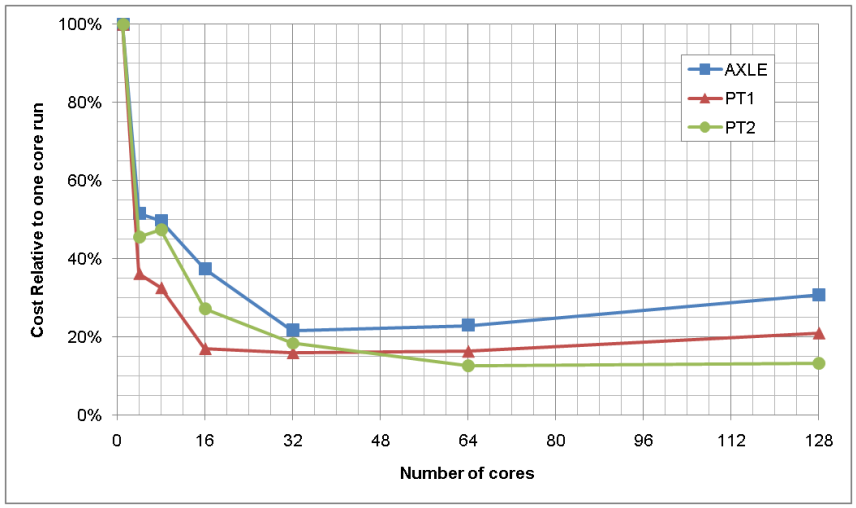

Figure 8: Sweet spot analysis for benchmark models

Based on the sweet spot analysis, a typical cluster load using 160 cores with a partition of thin and thick client focusing on runs for PT1 and PT2 type models can look like shown in Table 4. While we can use 32 cores for the PT1 type of models, we submit the PT2 type models to the thick client utilizing 48 or more cores.

Jobs Scenario (# of cores used)	Abaqus Token Usage	
2×48 + 64	78	Ideal situation
32 + 2×64	77	
3×48 (16 unused)	75	
2×32 + 2×48	92	
3×32 + 64	91	Extreme situation
5×32	105	

Table 4: Example Cluster Load

In addition to the above mentioned sweet spot study we believe that a good measure of server utilization is queue wait-time. CPU/load, I/O, and memory utilization by themselves do not capture the complete situation since the bottleneck in high-performance computing is often any one of these at various times in the life of an analysis. (CPUs can be idle while waiting for I/O, and then vice-versa.) Wait-time directly adds to the overall lead-time of a project, so it is important to minimize it as much as possible. Moving to clusters reduced the average wait-time from nearly 3.5 hours to well under 1 hour at one of our facilities.

Over our previous systems we achieved impressive results from both perspectives:

- Time savings
- Cost savings

We were able to reduce our time spent by about half at least. At the same time our costs per simulation were reduced by a minimum of 50%. But as indicated above, this is only half of the story. We need also to keep in mind how we load the cluster in a way so that we significantly can increase our number of design iterations during the same amount of time. Through an optimal loading of the cluster we have the option of increasing the number of simultaneous runs. The scalability information for our benchmark models helps to specify the number of cores needed for a job (Table 4). It doesn't benefit us to increase the simulation speed by (hypothetically) 40% if I still can only complete one design iteration per day. If I can add more

cores to the job and increase my simulation speed by 50% and I then can go through two design iterations a day and realize a tangible benefit. When the first job completes during working hours, I can adjust my design before the end of the day and submit another run. We have really gained time in the development process. For our powertrain simulation this allows us to complete on average two additional iterations per week compared to our previous system. In extreme cases for a PT2 model, increasing the number of cores from 64 to 128 can reduce the simulation time by half (Figure 9).

This is how we used the information collected during the evaluation. Of course, we continue performance studies all the time even with our new systems and evaluate the progress on the hardware market. In a large global company like Dana Holding Corporation, the deployment of such a compute system is not completed over night. We can then incrementally take advantage of new developments in hardware technology. For example, we made use of new hex-core CPUs by replacing the quad-core CPUs for clusters put in service later. Even though the hex-core system is only running four cores hot like a quad-core system, we already gained additional performance. Figure 9 shows the improvements achieved just by switching the CPU to the latest available for the HPC market.

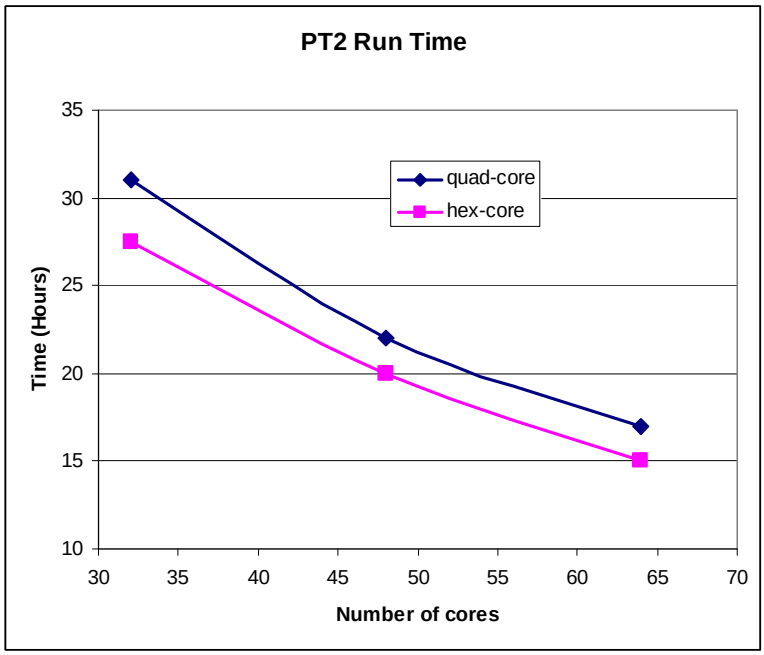

Figure 9: Comparison of quad vs hex-core system for PT2 benchmark model

But where do we go from here? One factor that we have not mentioned plays a major role in decision making for global corporations – cloud computing. For completely centralized engineering companies, meaning those which have only one major technical center, this does not play a critical role since they would have just one compute cluster. Dana Holding Corporation on the other hand has several technical centers around the world. This ensures close proximity to the customer base and allows for true 24x7 engineering, especially when looking at the interactive products within CAE.

In environments like Dana's we see the compute environment as a private cloud. The globally available hardware capacity will be managed by a central queuing system, not just a local one. This way it does not really matter any more where the clusters are located. The management system determines where the job will be run based on the compute requirements specified by the analysts. Software systems to enable this are pretty much already available today. The hurdle currently is that the network bandwidth to realize this is either not available or still too expensive. This is where the software side remote visualization needs to make a major step forward and network switches need to be developed to support data streams like we have in CAE to keep the costs reasonable.

Enabling efficient remote visualization means that post-processing is done at the cluster site. Currently it is done on the cluster site only if the cluster is local. Otherwise the data is first transferred to the local workstation. This takes though too much time. Sending input decks over the network is not an issue at all today.

The future network will be characterized by a central server managing the metadata while the large data itself is always stored locally. The rest will be accessed through remote visualization. This shows that the major break-through has to come from software technology and network technology. The compute hardware of-course will also develop further, but from a cost perspective it already plays a minor role on a per job basis.

This infrastructure is the most flexible. It also enables true system engineering where simulation technology has to be available not just to the “classical” CAE analyst but to all levels of engineering.

3. Conclusions

High performance computing is finding its way into engineering in a rapid way. HPC is a requirement for engineering companies to stay competitive. In the case of our situation the savings and gains where impressive. While reducing the relative costs in engineering, the time to market was significantly improved and more detailed simulation models were introduced. The more detailed models allowed us to improve quality up-front in the design process.

Since further improvements in HPC are very closely linked with software development, especially for CAE codes like Abaqus, it is very important to continue joint development efforts between end user, compute hardware provider and CAE software developer.

HPC architectures are a pre-requisite for a large scale system engineering implementation. This will require software for process and asset management in the CAE area, like SLM. It will enable the effective use of optimization codes, like iSight and it will incorporate simulation techniques for virtual testing and manufacturing in a way that has not been done before.

HPC is a largely deployed technology; however it is still only in its infancy today. The future of HPC is only just starting.

4. References

1. Abaqus as an integral part of Dana’s engineering strategy; Frank Popielas; ABAQUS World User Conference 2006, Boston
2. Simulation Lifecycle Management (SLM) as Central Part of Future Engineering; Frank Popielas; SCC 2010 – General Lecture, Providence
3. Accelerated Simulation Performance through High Performance Computing for Advanced Sealing Applications; Frank Popielas, et.al.; SCC 2009; London