ANSYS-Colfax HPC Study

Contents

Purpose

Extensive use of ANSYS Release 14.5.7, with two high-performance computing (HPC) licenses, over the last eight months, has required an excessive amount of waiting time for computer processing. Reducing the elapsed time spent waiting for the processing can be accomplished by purchasing HPC hardware and software. The business case for spending money on hardware and software to reduce elapsed time depends on the size of the reduction in time relative to the cost of the hardware and software. One purpose of this study is to measure the size of the reduction in elapsed time, which can be achieved by many combinations of hardware and software choices. Another purpose of this study is to evaluate which combination provides the most benefit for the least cost.

Executive Summary

Nine models solved at Carestream in 2013 were exercised on HPC hardware provided by Colfax International. **Speedup = (Elapsed Time on 4 cores)/(Elapsed Time on HPC configuration)** was the metric used. Speedup for the nine models using up to 8 cores, or accelerators, is shown in the table below.

Model	Nodes	Elements	GPU Acceleration	Best 1 Pack Speedup	Best Speedup Configuration	Best Solver Configuration	Best Memory Configuration	Speedup vs. 2 cores
m5-BC	9,953	1,742	No	1.14	8 cores	Sparse^	Shared	1.43
m1-VH	155,730	106,640	Yes	1.47	8 cores	PCG	Distributed	2.35
m4-SH	168,494	104,550	Yes	1.76	8 cores	PCG	Distributed	3.42
m6-H3	199,363	128,843	No	1.19	8 cores	Sparse^	Distributed	1.61
m3-AA	209,458	95,680	Yes*	1.17	8 cores	Sparse	Shared	1.41
m7-DT	454,082	243,037	Yes	1.30	8 cores	Sparse	Distributed	2.06
m8-WH	702,183	159,470	Yes*	1.62**	5 core, 3 GPU	PCG	Distributed	5.40
m2-TS	851,865	559,661	No	1.74	8 cores	Sparse^	Distributed	2.56
m9-RC	2,332,547	1,411,030	Yes*	1.59	8 cores	PCG	Distributed	2.41
		* model change required		**1.48 for 8 cores, 0 GPUs		^ only choice		

The average speedup was 1.43 when using 8 cores. Little improvement was seen with more cores. GPU acceleration does not provide any benefit on eight of these models. When two models can be solved at the same time, the speedup for two models is doubled by running them in parallel on a 16-core machine.

ANSYS Background

ANSYS solves a large-matrix equation by using either the Direct Sparse or the Iterative PCG solver. Not all models can use the Iterative PCG solver if an element used in the model is not supported. A few models only need to solve the matrix equation once, because they are linear. Most models have nonlinearity due to contact, large displacements, or nonlinear materials, and they need to converge by taking small increments toward the final load. The matrix equation is being solved either tens or hundreds of times.

ANSYS developed parallel technology using Shared Memory then Distributed Memory methods. Distributed Memory generally results in the minimum elapsed time in Release 14.5.7. Release 15.0, which was available on December 3, 2013, introduced support for another type of hardware accelerator, the Xeon Phi. Release 15.0 claims that sparse-solver performance has been significantly improved when running on the latest nVIDIA GPUs such as the Tesla K40 [1].

ANSYS HPC License Choices

One ANSYS solver license will run on 2 cores. To reduce elapsed time, the following can be purchased:

- One ANSYS HPC license will add one core or accelerator, multiple HPC licenses can be used together.
- One ANSYS HPC Pack can support a total combination of 8 cores or accelerators on one job.

  Such as: 8 cores, 0 accelerators,
  7 cores, 1 accelerator,
  6 cores, 2 accelerators,
  5 cores, 3 accelerators, and
  4 cores, 4 accelerators.

Two ANSYS HPC Packs can support a total combination of 32 cores or accelerators on one job. Capabilities of HPC licenses and Packs changed in Release 15.0.

One HPC Pack adds 6 cores to the 2 that NLS provide, but the entire Pack is checked out when a job runs, so if a user with a quad-core workstation needs only two HPC licenses and checks out a Pack to get them; the user is potentially “wasting” 4 cores in the Pack if someone else needs a full Pack. Single HPC licenses are more flexible. License costs are higher than hardware costs, so this study examines how to minimize elapsed time within the constraints imposed by an HPC license choice.

HPC Computer Hardware Choices

Most desktop computers have at least a 4-core processor, but they may not support RAM beyond 64 GB and may not have a spare PCIe x16 slot for an accelerator. These computers are usually less than $2K. An HPC desktop will have two CPU sockets, and each CPU, can have 8 cores, for a total of 16 cores. An HPC desktop will have several PCIe x 16 slots, support over 256 GB of RAM, and cost about $20K. There is a tradeoff between higher clock speeds with fewer cores, or lower clock speeds with more cores. The ANSYS solver needs sufficient RAM installed so it will not need to use disk storage while solving, as that greatly increases elapsed time. RAM is the least-expensive way to reduce elapsed time if the solver has run out of it. All jobs in this study were solved in less than 64 GB of RAM, and the hardware used in the study had 128 GB of RAM.

Accelerator Hardware Choices

The accelerators studied were the nVIDIA Tesla K40 GPU and the Intel Xeon Phi 7120P co-processor, which are each the top-of-the-line accelerators available at this time and supported by Release 15.0 [1]. Each job can only run with either the nVIDIA or the Intel accelerator. The cost of each accelerator is about $5.5K.

Comparison between ANSYS Release 15.0 support for Tesla GPU and Xeon Phi co-processor [2]

	Tesla GPU	**Xeon Phi co-processor**
OS	Windows or Linux	Linux
Solutions	Static linear or nonlinear analyses using the sparse, PCG, or JCG solver. Modal analyses using the Block Lanczos, subspace, PCG Lanczos, QR damped, unsymmetric, or damped eigensolver.	Static linear or nonlinear analyses using the sparse solver (symmetric matrices only). Iterative PCG solver not supported. Modal analyses using the Block Lanczos, subspace, or QR damped eigensolver.
Shared-Memory Parallel Behavior	For the sparse solver (and eigensolvers based on the sparse solver), if one or more GPUs are requested, only a single GPU is used no matter how many are requested. For the PCG and JCG solvers (and eigensolvers based on the PCG solver), all requested GPUs are used.	Only the sparse solver (and eigensolvers based on the sparse solver) can utilize Xeon Phi coprocessors. If one or more coprocessors are requested, all requested coprocessors are used.
Distributed Memory Parallel Behavior	For the sparse solver (and eigensolvers based on the sparse solver), if the number of GPUs exceeds the number of processes (the-na value is greater than the -np value on the command line), the number of GPUs used equals the -np value. If the number of GPUs is less than the number of processes, all requested GPUs are used. For the PCG and JCG solvers (and eigensolvers based on the PCG solver), if the number of GPUs is less than the number of processes, all requested GPUs are used.	Distributed-memory parallel processing is not supported.
No Acceleration if…	Partial pivoting is activated when using the sparse solver. This mostly occurs when using current technology elements with mixed u-P formulation or Lagrange multiplier based contact elements (TARGE169 through CONTA178).	Same as Tesla.

ANSYS 15.0 supports using multiple cores to mesh parts in parallel, but only on 64-bit Windows OS, not Linux. Without running any benchmarks, the table above suggests that the nVIDIA GPU has fewer limitations.

Colfax Hardware Test Configuration

Dr. Andrey Vladimirov, Head of HPC Research at Colfax-International (an HPC hardware vendor), generously agreed to support this study by making available a CXT/P9000 server configured with:

- Dual-socket Intel Xeon E5-2640 V2 8 core, 2.0 GHz
- 128 GB of 1600 MHz RAM
- Four Intel Xeon Phi 7120P, driver MPSS 3.1
- Four Nvidia Tesla K40m, driver CUDA 5.5
- Two spinning hard drives (one 6.0 GB/sec interface and one 3.0 GB/sec interface)
- CentOS 6.3 with kernel 2.6.32-279.22.1.el6.x86_64 (The public version of Red Hat Enterprise Linux)

Secure remote access to the HPC computer was through SSH and VNC viewer software.

ANSYS HPC Test Automation

Jason Zbick, Analyst at SimuTechGroup (an ANSYS distributor), generously agreed to support this study by making available temporary licenses for the ANSYS software, installing the software and writing the batch script template that was used to set up and run all of the jobs in this study.

The batch script consists of a series of lines like the ones shown below:

- ansys150 -dis -b nolist -j file -i m3_Dist_Sprs.dat -o m3_Dist_Sparse_08cpu_0gpu.out -np 8
- ansys150 -dis -b nolist -j file -i m3_Dist_Sprs.dat -o m3_Dist_Sparse_07cpu_1tesla.out -np 7 -acc nvidia -na 1
- ansys150 -dis -b nolist -j file -i m3_Dist_Sprs.dat -o m3_Dist_Sparse_06cpu_2tesla.out -np 6 -acc nvidia -na 2

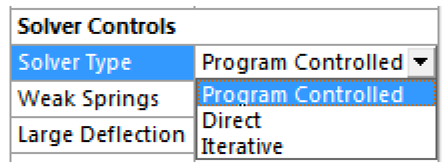
Each line will begin a job and the next line will not start until the current line completes.
The file name of the output file encodes the HPC configuration in terms of number of cores and number and type of accelerators used. The file name also encodes the type of memory ANSYS used (Shared or Distributed) and the type of solver ANSYS used (Direct Sparse or Iterative PCG). The last line of an output file contains the elapsed time for the job. In the above example, those three jobs represent HPC combinations possible with one HPC Pack and two nVIDIA GPUs.

Using the maximum number of cores and accelerators does not guarantee a minimum elapsed time. Decomposing the job into parallel streams and recombining the parallel results, at the end, adds overhead to the job. It’s possible to increase the overhead beyond the advantage of solving in parallel.

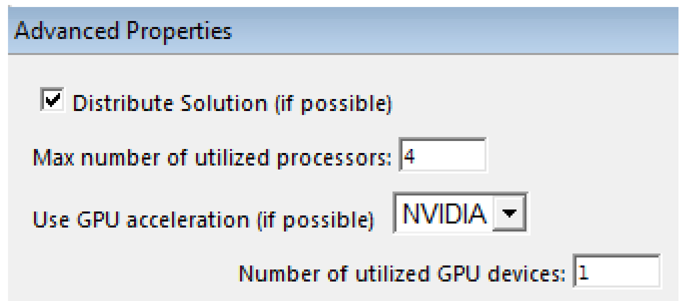
ANSYS Input Files

Peter Newman, Systems Engineer at Carestream Health, Inc. (a medical device manufacturer), provided nine ANSYS input files from jobs run in 2013 that represent typical Nonlinear Structural models. Details are in the sections below. The jobs were initially run on an HP Z420, E5-1620, 4-core, 3.6 GHz CPU with 64 GB RAM using 2 HPC licenses. The elapsed time for the complete solution for those nine models ranged from 7 minutes to 10 hours.

Models in Workbench default to a Program Controlled selection of solver. Often, the program chooses the solver that minimizes elapsed time. For this study, a solver (Direct Sparse or Iterative PCG) was chosen for each run. This choice writes different lines into the input file.



Solvers use a default setting for memory usage (Shared or Distributed), defined in Workbench by checking the first line in the Advanced Properties of the Solve Process Settings menu. The number of processors (cores) must be specified on line 2. If an HPC license is installed, a choice for the type and number of GPU accelerators is made on lines 3 and 4. These parameters are applied at the command line to run the solver as seen in the batch file above.



HPC Test Plan

One HPC Pack enables any combination of 8 cores or accelerators. Therefore, interesting combinations to study are those that add up to 8, such as:

- 8 cores, 0 accelerators,
- 7 cores, 1 accelerator,
- 6 cores, 2 accelerators,
- 5 cores, 3 accelerators, and
- 4 cores, 4 accelerators.

Cost increases with each accelerator. The purpose of the study is to determine the incremental value over the least-cost option, which is 8 cores, 0 accelerators. A shared memory sparse solver can only use one nVIDIA GPU.

A test plan with 30 runs exercises these combinations for each model and compares the performance to two baselines: a 2-core run that uses no HPC licenses, and a 4-core run that uses two HPC licenses.

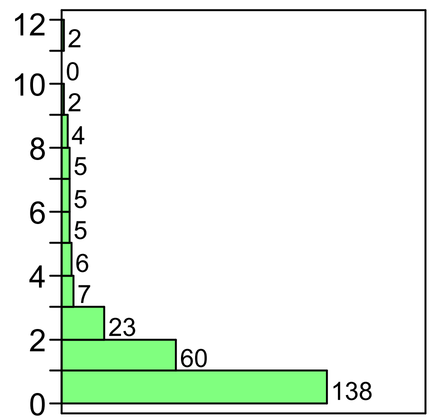
The best performance from the one HPC Pack test is chosen to run with more cores to see the possible benefit of using two HPC Packs (or additional HPC licenses).

Four Iterations Used on Long-Running Models

A model that completes in 7 minutes can be run through 30 different HPC combinations without taking too much time. A model that completes in 10 hours cannot be run through 30 different HPC combinations in the time available for this study. ANSYS provides a command (NCNV,1,,n,) to let the solver run through n iterations and then stop. A value of n between 3 and 5 is suggested to get an accurate estimate of the time per iteration. M2-TS took 10.5 hrs and 70 iterations to complete. When the job was run for 4 iterations, the elapsed time was extrapolated to 70 iterations and the estimated full-solution elapsed time was within 1% of the actual time.

Use of HPC License Checkout Log for Justification

Purchasing (or evaluating) an HPC license is a good way to gather data on the total time a user spends waiting for jobs to solve. The license server will automatically log the elapsed time of every job. Harvesting the log files after a few months of HPC use and summarizing the usage can provide the data needed to justify additional HPC licenses and the hardware to run them on.

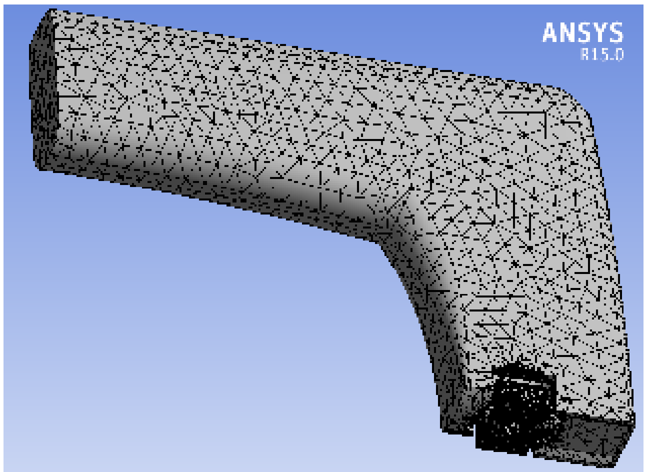


Count of jobs with elapsed times > 10 minutes grouped by Wait Time (hours) logged between 8/1 and 11/20.

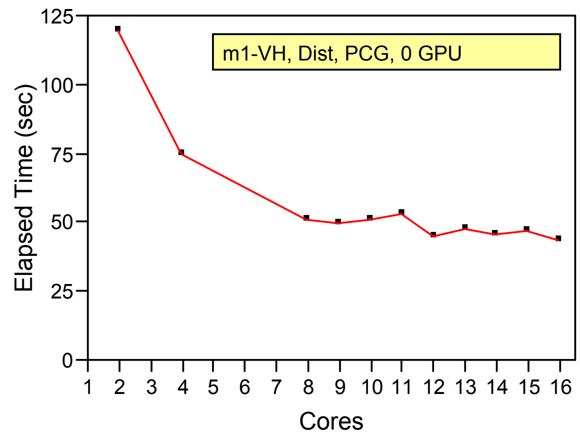
There were 257 jobs where the elapsed time was over 10 minutes. A histogram of those jobs shows the job count by elapsed time in hours. There were 138 jobs where I waited between 10 minutes and 1 hour, 60 jobs where I waited between 1 and 2 hours, 23 jobs where I waited between 2 and 3 hours, and 36 jobs where I had to wait more than 3 hours to get a result. The total time spent waiting for computer processing between August 1 and November 20, excluding the jobs that ran in less than 10 minutes, was **432 hours or 18 days**. On an annual basis, I could be waiting a total of **56 days** for FEA results from the desktop. Using the data in this study, the financial benefit of reducing wait time can be estimated to justify the cost of purchasing HPC licenses and hardware.

Performance Results for m1-VH

Model Size: 156,000 nodes
Nonlinearities: Material and Large Deformation
Solution time on a 4-core HP Z420: 87 minutes



Elapsed Time (sec) for 4 out of the 150 iterations



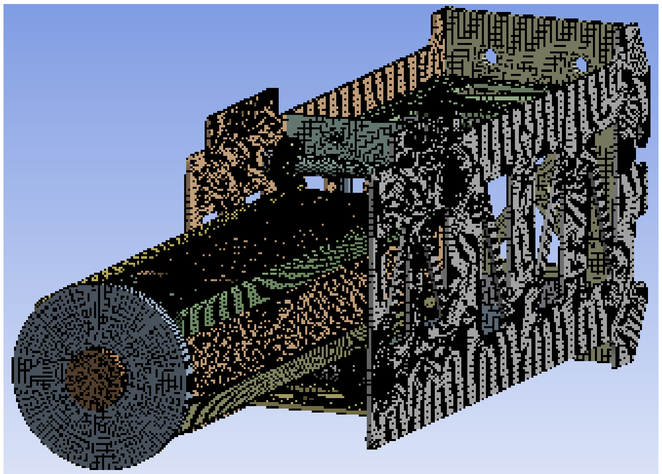
Configuration	Elapsed Time (min)	Speedup vs 2 HPC	Speedup vs No HPC	HPC	Comments
m1_VH_Dist_PCG_02cpu_0gpu.out	75	N/A	1.00	None	
m1_VH_Dist_PCG_03cpu_1tesla.out	63	0.75	1.20	2 HPC	No benefit using GPU vs 4 core below
m1_VH_Dist_PCG_04cpu_0gpu.out	47	1.00	1.60	2 HPC	Dist-PCG is best for this model
m1_VH_Dist_PCG_04cpu_4tesla.out	54	0.86	1.38	1 Pack	
m1_VH_Dist_PCG_05cpu_3tesla.out	51	0.93	1.48	1 Pack	
m1_VH_Dist_PCG_06cpu_2tesla.out	53	0.89	1.43	1 Pack	
m1_VH_Dist_PCG_07cpu_1tesla.out	51	0.93	1.48	1 Pack	No benefit using GPUs vs 8 core below
m1_VH_Dist_PCG_08cpu_0gpu.out	32	1.47	2.35	1 Pack	Best 1 Pack

Note: All models were run through the two types of memory usage (Distributed and Shared) and the two types of solvers (Sparse and PCG) when supported. All data is included in a separate spreadsheet embedded below, but only the best configuration has been copied to this report.

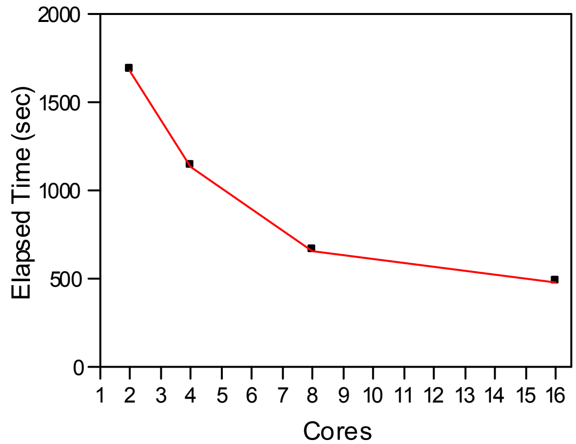
HPC results for
ANSYS150 Dec 2013

Performance Results for m2-TS

Model Size: 852,000 nodes
Nonlinearities: Contact and Material
Solution time on a 4-core HP Z420: 10.5 hours



Elapsed Time (sec) for 4 out of the 70 iterations

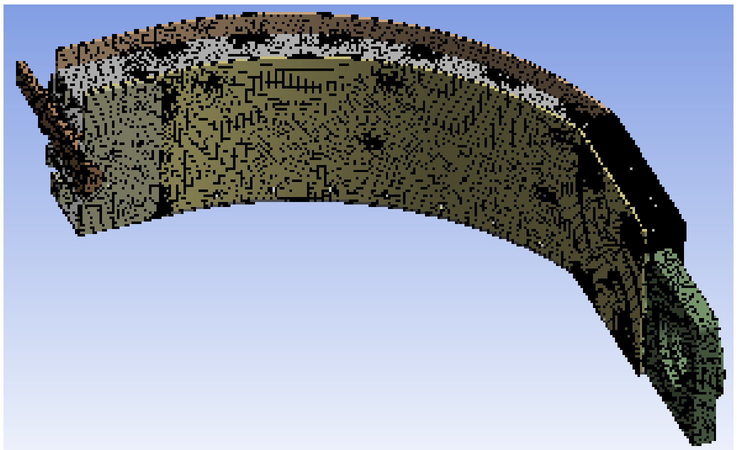


HPC Configuration	Elapsed Time (min)	Speedup vs 2 HPC	Speedup vs No HPC	HPC	Comments
m2_TS_Dist_Sparse_02cpu_0gpu.out	491	N/A	1.00	None	
m2_TS_Dist_Sparse_03cpu_0gpu.out	361	0.92	1.36	2 HPC	No GPU possible due to Joint element
m2_TS_Dist_Sparse_04cpu_0gpu.out	333	1.00	1.47	2 HPC	Dist-Sparse is best for this model
m2_TS_Dist_Sparse_04cpu_0gpu.out	342	0.97	1.43	1 Pack	No GPU possible due to Joint element
m2_TS_Dist_Sparse_05cpu_0gpu.out	290	1.15	1.69	1 Pack	No GPU possible due to Joint element
m2_TS_Dist_Sparse_06cpu_0gpu.out	266	1.25	1.85	1 Pack	No GPU possible due to Joint element
m2_TS_Dist_Sparse_07cpu_0gpu.out	244	1.36	2.01	1 Pack	No GPU possible due to Joint element
m2_TS_Dist_Sparse_08cpu_0gpu.out	192	1.74	2.56	1 Pack	Best 1 Pack
m2_TS_Dist_Sparse_16cpu_0gpu.out	141	2.36	3.48	2 Pack	Best 2 Pack
m2_TS_Dist_Sparse_16cpu_0gpu.out	153	2.18	3.21	2 Pack	No GPU possible due to Joint element
m2_TS_Dist_Sparse_16cpu_0gpu.out	153	2.19	3.22	2 Pack	No GPU possible due to Joint element
m2_TS_Dist_Sparse_16cpu_0gpu.out	152	2.20	3.24	2 Pack	No GPU possible due to Joint element
m2_TS_Dist_Sparse_16cpu_0gpu.out	150	2.22	3.27	2 Pack	No GPU possible due to Joint element

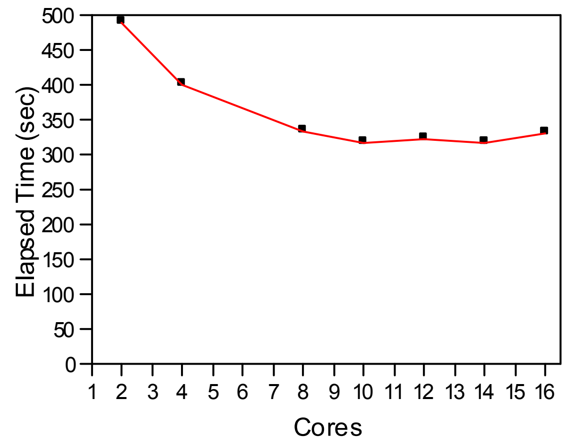
This model used three joints (two revolutes, one cylindrical) that could not be replaced by beams so it could not use any GPUs or the PCG solver.

Performance Results for m3-AA

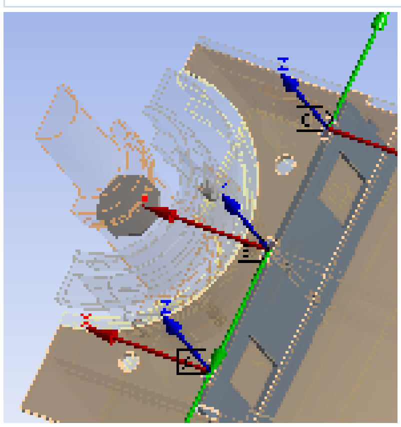
Model Size: 209,000 nodes
Nonlinearities: Contact and Material
Solution time on a 4-core HP Z420: 114 minutes

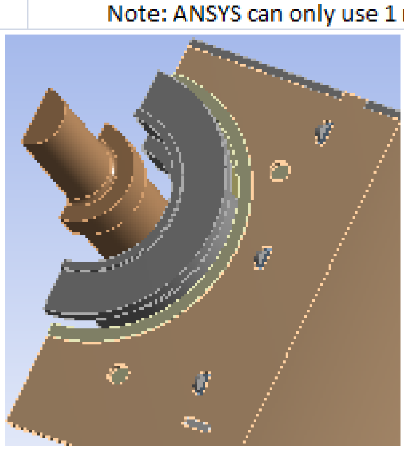


Elapsed Time (sec) for 4 out of the 137 iterations



HPC Configuration	Elapsed Time (min)	Speedup vs 2 HPC	Speedup vs No HPC	HPC	Comments
m3_AB_Shared_Sparse_02cpu_0gpu.out	279	1.91	1.00	None	
m3_AB_Shared_Sparse_03cpu_1tesla.out	260	0.88	1.07	2 HPC	No benefit using GPU vs 4 core below
m3_AB_Shared_Sparse_04cpu_0gpu.out	229	1.00	1.22	2 HPC	Shared Sparse is best for this model
m3_AB_Shared_Sparse_04cpu_1tesla.out	227	1.03	1.23	1 Pack	See note.
m3_AB_Shared_Sparse_05cpu_1tesla.out	232	1.00	1.20	1 Pack	See note.
m3_AB_Shared_Sparse_06cpu_2phi.out	218	1.07	1.28	1 Pack	
m3_AB_Shared_Sparse_06cpu_1tesla.out	231	1.00	1.21	1 Pack	See note.
m3_AB_Shared_Sparse_07cpu_1phi.out	211	1.10	1.33	1 Pack	
m3_AB_Shared_Sparse_07cpu_1tesla.out	210	1.11	1.33	1 Pack	No benefit from using GPUs
m3_AB_Shared_Sparse_08cpu_0gpu.out	199	1.17	1.41	1 Pack	Best 1 Pack
	Note: ANSYS can only use 1 nVIDIA GPU when solving with Shared Memory				

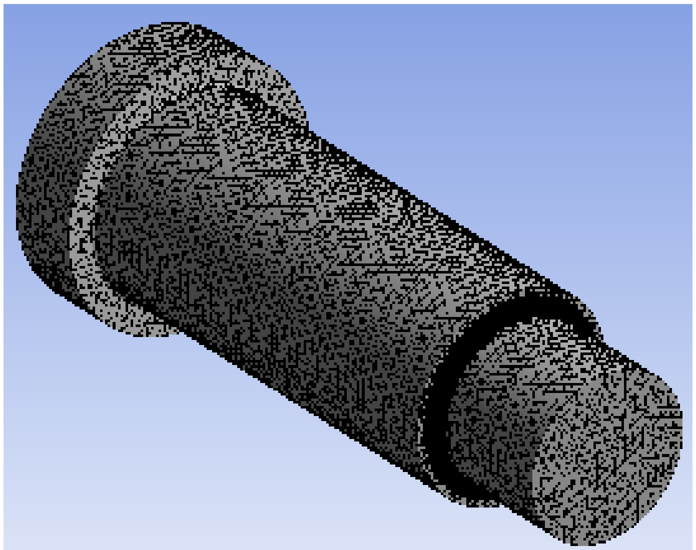


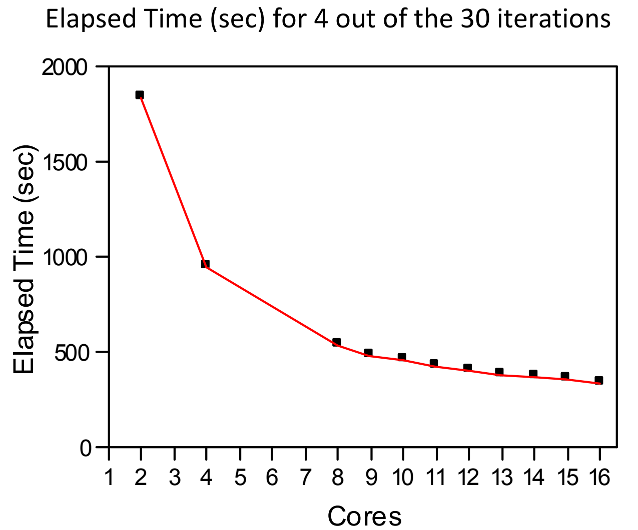


This model used three fixed joints that were replaced by beams to enable GPU acceleration and the PCG solver.

Performance Results for m4-SH

Model Size: 343,000 nodes
Nonlinearities: Contact (compression only support) and Material
Solution time on a 4-core HP Z420: 10 minutes



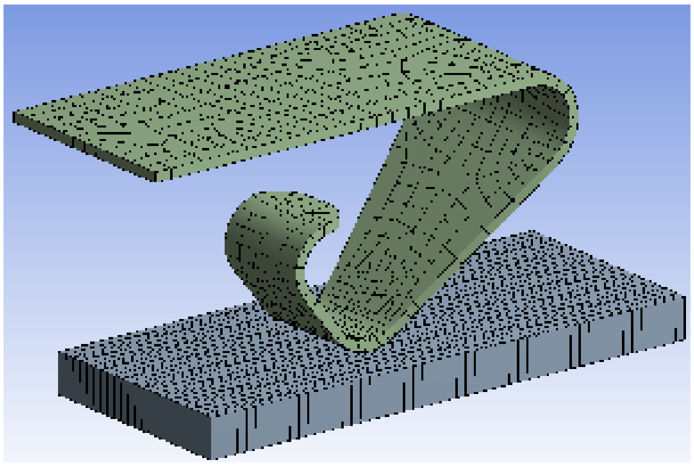


HPC Configuration	Elapsed Time (min)	Speedup vs 2 HPC	Speedup vs No HPC	HPC	Comments
m4_SH_Dist_PCG_02cpu_0gpu.out	31	N/A	1.00	None	
m4_SH_Dist_PCG_03cpu_1tesla.out	21	0.75	1.46	2 HPC	No benefit using GPU vs 4 core below
m4_SH_Dist_PCG_04cpu_0gpu.out	16	1.00	1.95	2 HPC	Dist-PCG is best for this model
m4_SH_Dist_PCG_04cpu_4tesla.out	17	0.93	1.81	1 Pack	
m4_SH_Dist_PCG_05cpu_3tesla.out	14	1.11	2.16	1 Pack	
m4_SH_Dist_PCG_06cpu_2tesla.out	13	1.19	2.32	1 Pack	
m4_SH_Dist_PCG_07cpu_1tesla.out	12	1.31	2.54	1 Pack	No benefit using GPUs vs 8 core below
m4_SH_Dist_PCG_08cpu_0gpu.out	9	1.76	3.42	1 Pack	Best 1 Pack

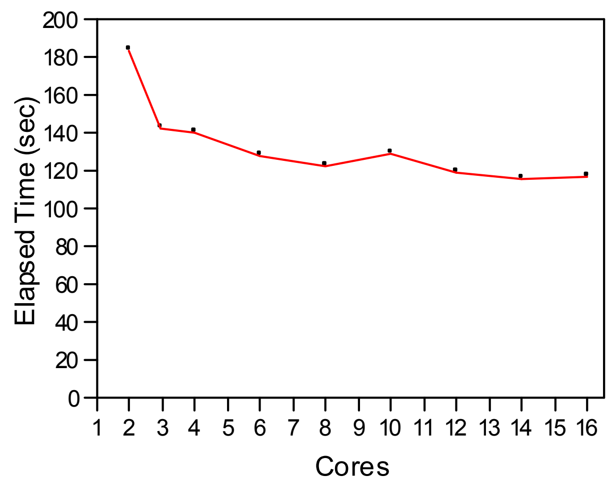
The output file for this model reports that the solution exceeded a threshold, and there may be rigid body motion. This is because the contact algorithm failed. Failure frequently happens on the first run of a nonlinear model with contact. Sometimes that is due to a mistake in the model, but often it is just because the initial step was too large and the correction is to reduce the initial step size. That was the case in this model, where 30 initial steps results in contact failure, while 50 initial steps succeeded. Unfortunately, the solver will often complete all of the iterations before stopping to show the contact failure. This is one of many reasons why tens of runs are required to complete a study. Other reasons are mesh density studies to determine the appropriate level of mesh refinement. Initial step studies and mesh refinement studies are good examples of jobs that can be run simultaneously to decide which initial step size or mesh density to use for the rest of the study.

Performance Results for m5- BC

Model Size: 10,000 nodes
Nonlinearities: Contact and Large Deformation
Solution time on a 4-core HP Z420: 18 minutes



Elapsed Time (sec) for 4 out of the 30 iterations

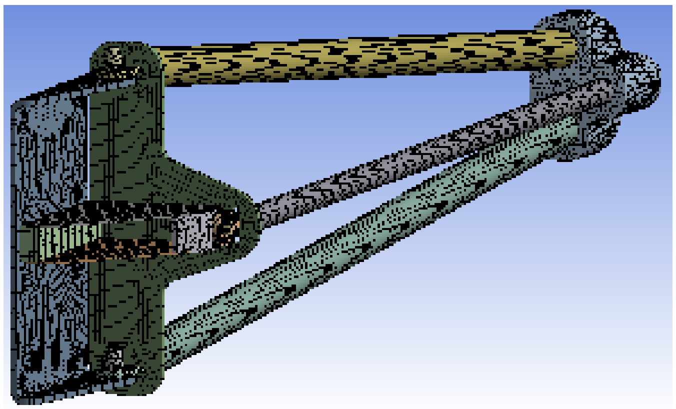


HPC Configuration	Elapsed Time (min)	Speedup vs 2 HPC	Speedup vs No HPC	HPC	Comments
m5_BC_Shared_Sparse_02cpu_0gpu.out	23	0.95	1.00	None	
m5_BC_Shared_Sparse_03cpu_0gpu.out	18	0.99	1.29	1 HPC	No GPUs used due to Contact element
m5_BC_Shared_Sparse_04cpu_0gpu.out	18	1.00	1.30	2 HPC	Shared-Sparse is best for this model
m5_BC_Shared_Sparse_06cpu_0gpu.out	16	1.10	1.44	None	
m5_BC_Shared_Sparse_07cpu_0gpui.out	18	0.99	1.30	1 Pack	No GPUs used due to Contact element
m5_BC_Shared_Sparse_07cpu_0gpu.out	17	1.05	1.37	1 Pack	No GPUs used due to Contact element
m5_BC_Shared_Sparse_08cpu_0gpu.out	15	1.15	1.50	1 Pack	Best 1 Pack
m5_BC_Shared_Sparse_10cpu_0gpu.out	16	1.09	1.43	1 Pack	
m5_BC_Shared_Sparse_12cpu_0gpu.out	15	1.18	1.55	1 Pack	
m5_BC_Shared_Sparse_14cpu_0gpu.out	15	1.22	1.59	1 Pack	Best 2 Pack
m5_BC_Shared_Sparse_16cpu_0gpu.out	15	1.21	1.57	1 Pack	

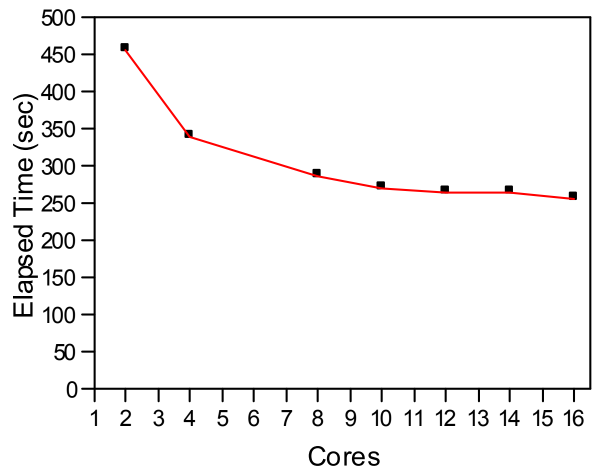
This model used contact elements that did not support GPU acceleration or the PCG solver.

Performance Results for m6-H3

Model Size: 200,000 nodes
Nonlinearities: Contact and Material
Solution time on a 4-core HP Z420: 96 minutes



Elapsed Time (sec) for 4 out of the 104 iterations

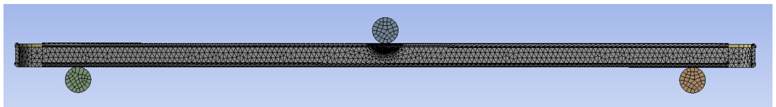


HPC Configuration	Elapsed Time (min)	Speedup vs 2 HPC	Speedup vs No HPC	HPC	Comments
m6_H3_Dist_Sparse_02cpu_0gpu.out	198	N/A	1.00	None	
m6_H3_Dist_Sparse_03cpu_0gpu.out	165	0.89	1.20	1 HPC	No GPUs used due to Contact element
m6_H3_Dist_Sparse_04cpu_0gpu.out	147	1.00	1.35	2 HPC	Dist Sparse is best for this model
m6_H3_Dist_Sparse_04cpu_0gpu.out	154	0.95	1.29	1 Pack	No GPUs used due to Contact element
m6_H3_Dist_Sparse_05cpu_0gpu.out	158	0.93	1.26	1 Pack	No GPUs used due to Contact element
m6_H3_Dist_Sparse_06cpu_0gpu.out	141	1.04	1.40	1 Pack	No GPUs used due to Contact element
m6_H3_Dist_Sparse_07cpu_0gpu.out	139	1.06	1.43	1 Pack	No GPUs used due to Contact element
m6_H3_Dist_Sparse_08cpu_0gpu.out	123	1.19	1.61	1 Pack	Best 1 Pack

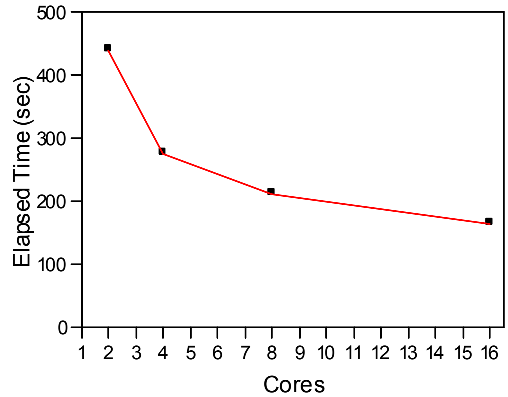
This model used contact elements that did not support GPU acceleration or the PCG solver.

Performance Results for m7-DT

Model Size: 454,000 nodes
Nonlinearities: Contact
Solution time on a 4-core HP Z420: 30 minutes



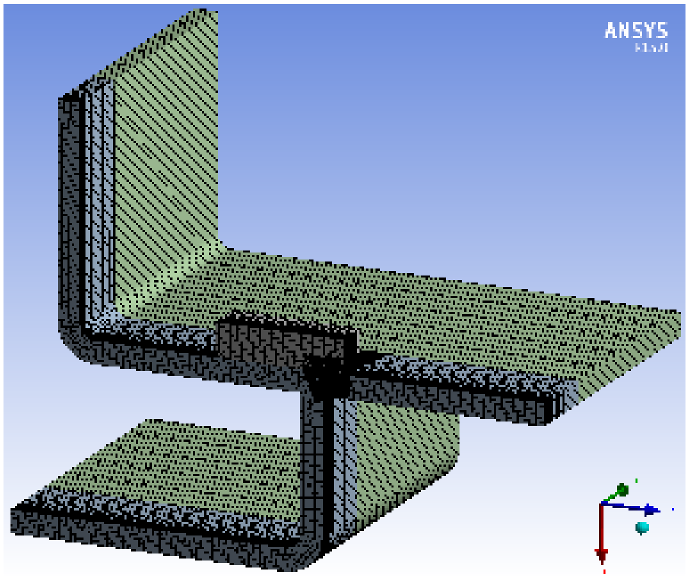
Elapsed Time (sec) for 4 out of the 47 iterations

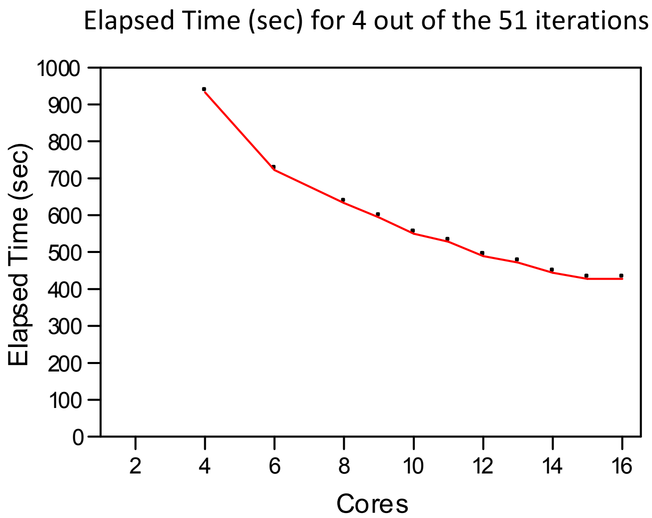


HPC Configuration	Elapsed Time (min)	Speedup vs 2 HPC	Speedup vs No HPC	HPC	Comments
m7_DT_Dist_Sparse_02cpu_0gpu.out	86	N/A	1.00	None	
m7_DT_Dist_Sparse_03cpu_1tesla.out	62	0.87	1.38	2 HPC	No benefit from using GPU
m7_DT_Dist_Sparse_04cpu_0gpu.out	54	1.00	1.58	2 HPC	Dist-Sparse is best for this model
m7_DT_Dist_Sparse_04cpu_4tesla.out	51	1.07	1.70	1 Pack	
m7_DT_Dist_Sparse_05cpu_3tesla.out	47	1.14	1.81	1 Pack	
m7_DT_Dist_Sparse_06cpu_2tesla.out	44	1.22	1.93	1 Pack	
m7_DT_Dist_Sparse_07cpu_1tesla.out	44	1.23	1.95	1 Pack	No benefit from using GPUs
m7_DT_Dist_Sparse_08cpu_0gpu.out	42	1.30	2.06	1 Pack	Best 1 Pack
m7_DT_Dist_Sparse_16cpu_0gpu.out	33	1.67	2.64	2 Pack	Best 2 Pack
m7_DT_Dist_Sparse_16cpu_1tesla.out	40	1.36	2.15	2 Pack	No benefit from using GPU
m7_DT_Dist_Sparse_16cpu_2tesla.out	41	1.31	2.08	2 Pack	

Performance Results for m8-WH

Model Size: 700,000 nodes
Nonlinearities: Contact
Solution time on a 4-core HP Z420: 10.8 hours





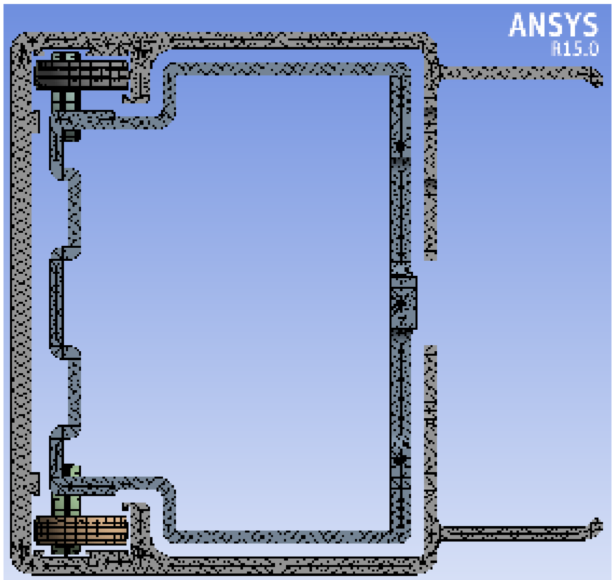
Cores plot is without GPU accelerator.

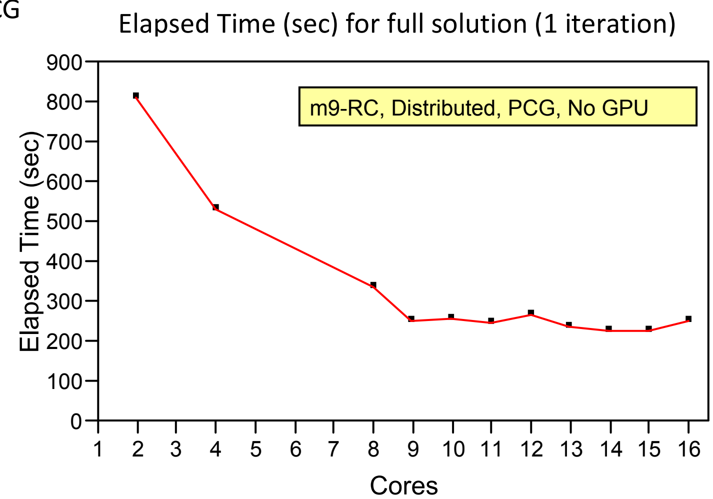
HPC Configuration	Elapsed Time (min)	Speedup vs 2 HPC	Speedup vs No HPC	HPC	Comments
m8_WH_Dist_PCG_02cpu_0gpu.out	663	N/A	1.00	None	
m8_WH_Dist_PCG_03cpu_1tesla.out	392	0.51	1.69	2 HPC	No benefit using GPU vs 4 core below
m8_WH_Dist_PCG_04cpu_0gpu.out	199	1.00	3.33	2 HPC	Distributed PCG is best for this model
m8_WH_Dist_PCG_04cpu_4tesla.out	131	1.53	5.07	1 Pack	
m8_WH_Dist_PCG_05cpu_3tesla.out	123	1.62	5.40	1 Pack	Best 1 Pack
m8_WH_Dist_PCG_06cpu_2tesla.out	126	1.59	5.28	1 Pack	
m8_WH_Dist_PCG_07cpu_1tesla.out	126	1.58	5.27	1 Pack	One tesla not much different than 4 tesla
m8_WH_Dist_PCG_08cpu_0gpu.out	135	1.48	4.91	1 Pack	Performance with no GPU

This was the only model where a GPU was useful.

Performance Results for m9-RC

Model Size: 2,333,000 nodes
Nonlinearities: None
Solution time on a 4-core HP Z420: 7 minutes with PCG

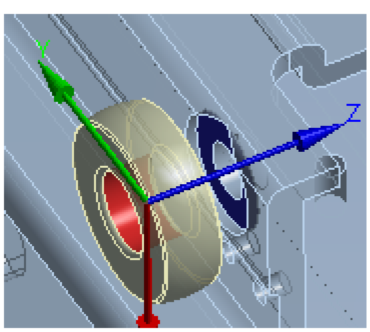


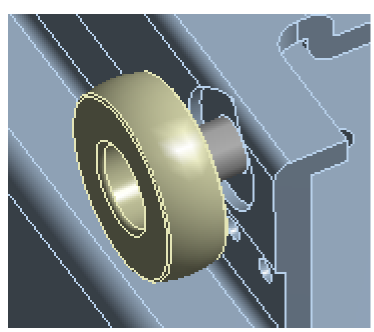


The only linear model of the nine. Linear models can solve in one iteration. Runs for cores 9-16 were run on disk1 while runs for cores 2-8 were run on disk2 (see below).

HPC Configuration	Elapsed Time (sec)	Speedup vs 2 HPC	Speedup vs No HPC	HPC	Comments
m9_RC_Dist_PCG_02cpu_0gpu.out	810	N/A	1.00	None	
m9_RC_Dist_PCG_03cpu_1tesla.out	648	0.82	1.25	2 HPC	No benefit from using GPU vs 4 core below
m9_RC_Dist_PCG_04cpu_0gpu.out	534	1.00	1.52	2 HPC	Distributed-PCG is best for this model
m9_RC_Dist_PCG_04cpu_4tesla.out	540	0.99	1.50	1 Pack	
m9_RC_Dist_PCG_05cpu_3tesla.out	413	1.29	1.96	1 Pack	
m9_RC_Dist_PCG_06cpu_2tesla.out	498	1.07	1.63	1 Pack	
m9_RC_Dist_PCG_07cpu_1tesla.out	355	1.50	2.28	1 Pack	No benefit from using GPUs
m9_RC_Dist_PCG_08cpu_0gpu.out	336	1.59	2.41	1 Pack	Best 1 Pack

This model used fixed joints that were replaced by beams to enable GPU acceleration. The surprising result was that the change to beams also enabled the PCG solver, which cut the elapsed time in **half** compared to the Sparse solver.





Processor Choice

The 4-core HP Z420 has a 3.6 GHz clock on an older chip design. The 16-core server has a 2.0 GHz clock on a newer chip design. The clocks have a 1.8 ratio. Clock speed affects the elapsed time. For example, on the m4-SH model, the solution time for the Shared-Sparse solution with 4 cores on the 2.0 GHz server was 27 minutes, while that same solution on the 3.6 GHz workstation was 18 minutes, which is a 1.5 ratio. The clock alone would predict a time of 32 minutes on the server, but the newer chip design is more efficient, so it makes up for some of the reduced clock speed. A second example is the m9-RC model, where the solution time for the Dist-PCG solution with 4 cores on the 2.0 GHz server was 534 seconds, while that same solution on the 3.6 GHz workstation was 436 seconds, which is a 1.2 ratio. The clock alone would predict a time of 784 seconds on the server. Intel offers a range of clock speeds. I will choose the fastest clock available to reduce the elapsed time.

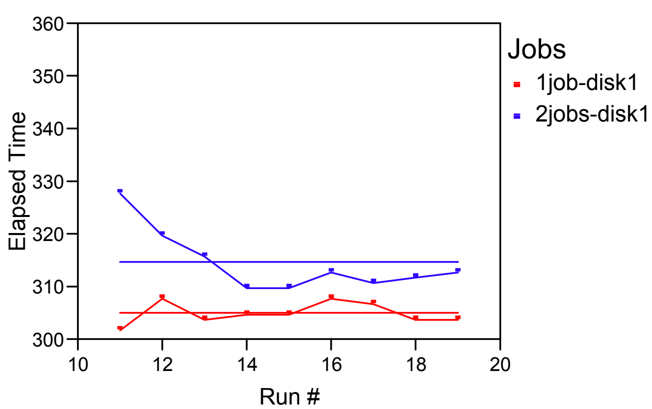
Solving Two Models Simultaneously

Most FEA studies will solve several variations of the initial model before the study is complete. Ten to 40 jobs are often run to complete a study. When these model variations are known in advance, solving two models simultaneously guarantees cutting the solving time in half compared with running them sequentially. Simultaneous jobs could be run by having a second computer, but it is much more convenient to keep all of the files in one place.

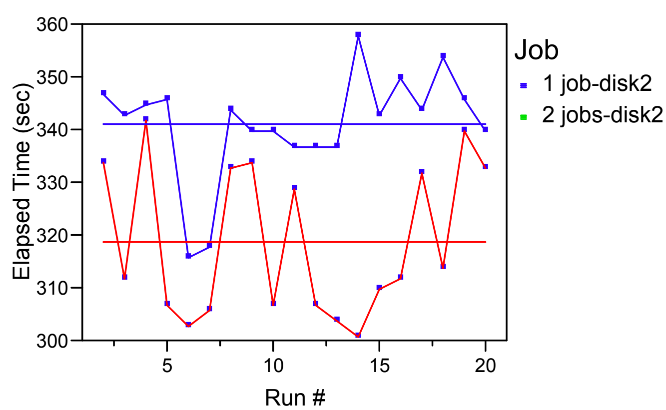
A model was configured to run using 8 cores, and a script with 10 repeats of that model was started on disk2. Running simultaneously, a similar copy of that script was started, but using disk1. Once those two scripts finished running together, the script on disk1 was started to run alone. Finally, the script on disk 2 was started to run alone. The result was a 3% increase in average elapsed time when running two jobs simultaneously compared to running a single job. Therefore, running two jobs simultaneously will deliver a 2x increase in speed.

The server was configured with a high-performance HDD as disk1 and a low-performance HDD as disk2. The plots below show the difference in performance of disk 1 with 6.0 GB/sec transfer performance vs. disk 2 with 3.0 GB/sec performance. Some of the ripple in the plots may be due to being run on disk2 (m9) vs. disk1 (m8).

High Performance HDD, 6.0 GB/sec



Low Performance HDD, 3.0 GB/sec



This data shows the importance of high-performance storage, even if the solution is done in-core. Solid state drives are recommended by ANSYS for the solve disk.

Conclusions

1. Eight models saw no benefit from a GPU, so I would not purchase one to reduce the hardware cost.
2. When two models can be solved at the same time, the speedup for two models is doubled by running them in parallel on a 16-core machine, which requires two HPC Packs or 12 HPC licenses.
3. The same argument can be made when three models can be solved at the same time, in order to triple the speed for all three to solve, by running them in parallel. Each job gets 5 cores and three HPC licenses using a total of nine HPC licenses. This requires the purchase of a third NLS license. Three models that require 64 GB of RAM will consume 192 GB RAM, the limit that 64-bit Windows 7 can support.
4. The average speedup of a single job was 1.43 when using 8 cores. Little improvement was seen with more cores; therefore, there is little value in purchasing two HPC Packs to run single jobs. Using 7 cores is just as good as 8 cores for small models and almost as good for large models.
5. A third NLS license and many HPC licenses are more flexible for multiple ANSYS users than are HPC Packs.
6. Fast drives are required for minimizing solve times, even if the solution is done in RAM.
7. Use beams instead of fixed joints to enable the PCG solver.
8. Continue to default to Distributed ANSYS. It was faster on seven of the nine models. The two models that were faster on Shared memory would have been 13% and 24% slower on Distributed; a significant benefit, therefore it is worth testing which one is faster.
9. Initially, use the NCNV,1,,3, command on nonlinear models to get the elapsed time for three iterations to determine if the Sparse or PCG solver will be best, then do one additional run on the Shared memory model to determine if that, or Distributed memory, is best before running the 10-40 jobs in the study.

Recommendations

1. Purchase a workstation with Dual CPUs: Intel Xeon E5-2687Wv2, 3.4 GHz, 8 core, with 192 GB of RAM
2. Add a RAID0 solve disk configured with 4 x 256 GB Solid State Drives.
3. Run 64-bit Windows 7 for CAD compatibility.
4. Purchase a third ANSYS NLS license.
5. Purchase eight HPC licenses to add to the two that Carestream owns.
6. Set up this workstation for Remote Solve Manager to allow all ANSYS users to submit remote jobs to it. This combination will allow the FEA Workstation to run:
    - One job on up to 12 cores,
    - Two jobs on 7 cores simultaneously,
    - Three jobs on 5 cores simultaneously.

If another user with a quad core workstation is using one NLS license and two HPC licenses, the FEA workstation can run one job on 10 cores, or two jobs on 6 cores. This investment will maximize the productivity of FEA work.

References

[1] ANSYS, Inc. Release Notes, Release 15.0, November 2013, 000504

[2] ANSYS Mechanical APDL Parallel Processing Guide, Release 15.0, November 2013