ACCELERATING MECHANICAL SOLUTIONS WITH GPUs

Graphical processing units can be used with ANSYS structural mechanics software to solve large, complex models faster.

By Sheldon Imaoka, Principal Engineer, ANSYS, Inc.

ANSYS structural mechanics products have supported parallel processing for over two decades, and the distributed solution capabilities provide extremely fast solution turnaround times. ANSYS users, however, always push the boundaries related to size and complexity of models that can be solved with current hardware. The use of high-end graphical processing units (GPUs) can provide analysts with a way to decrease overall solution times, since they can solve larger models in the same amount of time to provide better numerical accuracy or solve models more quickly for shorter turnaround times. Although the actual speedup is dependent on hardware constraints and model characteristics, the methods used in this article can help you to obtain optimal performance for a given scenario using the GPU acceleration capability in ANSYS structural mechanics software.

ACCELERATING BOTH OLD AND NEW HARDWARE

New and older hardware both can benefit from the use of GPU acceleration. You can use this tool on a single machine as well as on a cluster configuration.

GPUs provide analysts with a way to decrease overall solution times.

For example, a block Lanczos cyclic symmetry modal analysis of a radial impeller was solved on a two quad-core Intel® Xeon® E5530 (Nehalem architecture) workstation. (This chip was made available in early 2009.) By today's standards, this CPU is a few generations behind. However, using an NVIDIA® Quadro® 6000 card with GPU acceleration, you can obtain significant speedup without having to replace the entire workstation.

▲ Radial impeller cyclic symmetry model (visually expanded)

GEOMETRY COURTESY PADT, INC.

Solve larger models in the same amount of time to provide better numerical accuracy, or solve models more quickly for shorter turnaround times.

Number of cores	GPU (Quadro 6000)	Speedup
2	no	1.00
8	no	2.67
2	yes	3.60
8	yes	4.30

DSLP model on Intel E5-2690 and NVIDIA Tesla C2075

Compared with the default case of using two cores, adding the GPU provided a 3.6 times speedup. Because of the older architecture, the speedup from two to eight cores resulted in only a 2.7 times speedup, but adding a GPU increased it further by 1.6 times.

On the other hand, when the newer Intel Xeon E5-2690 (Sandy Bridge architecture) was used to solve a PCG Lanczos modal analysis of a dual-segmented langmuir probe (DSLP) on two compute nodes, each with a GPU and employing GPU acceleration, the overall solution decreased by a factor of 1.8 times.

Even on new hardware, GPU acceleration can provide faster solutions, and the ability to use multiple GPUs in a single simulation is very attractive for large models.

HOW GPU ACCELERATION WORKS

While a CPU has eight or fewer cores, a GPU may have hundreds or thousands of cores. CPU and GPU cores are not directly comparable — a CPU core is designed to handle general, complex instructions, while a GPU core is meant for specific, simpler tasks — but the idea behind GPU acceleration is to leverage hundreds or thousands of GPU cores for reducing overall solution time.

Some overhead is associated with packing the data on the CPU, sending it to the GPU, then retrieving the information. GPU cores are designed for specific tasks, so only the most computationally intensive portion of a solution is sent to the GPU when GPU acceleration is used.

Analyses that spend a significant portion in matrix factorization in which the computation is offloaded to the GPU will benefit the most from GPU acceleration. If the factorization is a small percentage of the overall elapsed time, then GPU acceleration will not have as great an impact in reducing the solution time.

Number of cores	GPU (Tesla C2075)	Speedup
16 (2 x 8)	no	1.00
16 (2 x 8)	yes (2)	1.83

2 million DOF impeller model on Intel E5530 and NVIDIA Quadro 6000

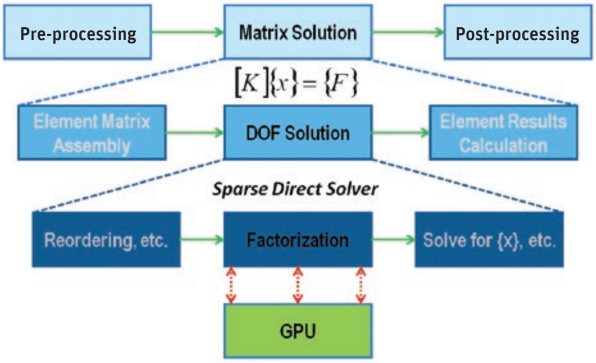
When solving in batch mode, the Mechanical APDL 14.5 solver output file contains the following line after the solution is complete:

```
Total CPU time for main thread:    3868.9 seconds
```

The last line of the solver output is:

```
Elapsed Time (sec) = 3877.000    Date = 03/18/2013
```

If the total CPU time for the main thread is a high percentage of the total elapsed time (in this example, 99.8 percent), the solution is compute-bound, and adding more cores, GPUs or both could decrease the solution time even further. If the total CPU time for the main thread is not a high percentage of the total elapsed time, then the solution is I/O bound, meaning that disk speed, network speed, memory bandwidth and/or other factors are a bottleneck.



Flowchart for sparse direct solver and GPU acceleration. The computation is offloaded to the GPU during matrix factorization for the sparse direct solver.

In the sparse solver statistics output file (“file.BCS”), you can find the following line:

```
time (cpu & wall) for numeric factor =
6763.03  3450.27
```

The first value can be ignored, but the second value gives the total time spent on factorization. If this value is large compared with the overall elapsed time (in this case, 89 percent), this indicates that the solution should benefit with GPU acceleration.

For the PCG iterative solver, the same idea applies, but, in this case, use the following line in the PCG solver statistics file (“file.PCS”):

```
Multiply With A22     1309.82   1307.28
```

The second number shown under `Multiply With A22` provides information on the portion of the solution that can be accelerated with the GPU accelerator. If this value is a large percentage of the total elapsed time, GPU acceleration should have a substantial influence in reducing the turnaround time.

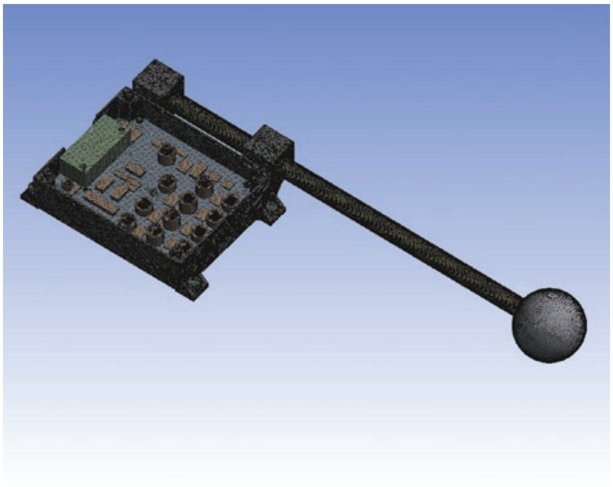
If the factorization or `Multiply With A22` time is a smaller fraction of the overall elapsed time, you can still use GPU acceleration, but the benefit will not be as great as some of the numbers published within this article. Similarly, if the CPU time for the main thread is a smaller fraction of the overall solution time, it is likely that general parallel processing (that is, using more CPU cores) may not have an appreciable impact because the solution is not compute-bound.

TIPS ON USING GPU ACCELERATION

You can use GPU acceleration to solve many problems, either with the SMP or distributed version of Mechanical APDL. Here are a few tips to get the most out of GPU acceleration:

- *If using the sparse direct solver, ensure that the solution is always solving in-core.* The sparse direct solver can utilize a lot of memory because the factored stiffness matrix is significantly more dense than the original matrix. If the solution runs in out-of-core mode, there will be a lot of I/O operations performed, causing a bottleneck so that the data will not be sent to the CPU and GPU fast enough and, therefore, compute power will be underutilized. A minimum of 48 GB of physical RAM is suggested, although this depends on the size of the models you typically solve.

- *If using the PCG iterative solver, consider using multiple GPUs.* While the sparse direct solver requires a lot of physical RAM compared to the amount of graphics memory, the PCG iterative solver is the opposite: Less physical RAM is needed, but more graphics memory is desirable. The supported graphics cards typically have 5 GB to 6 GB of graphics memory, so the only option for increasing graphics memory is to use additional cards. In ANSYS 14.5, multiple GPUs can be used for supported iterative solvers, allowing for larger problems to be solved on GPUs.



▲ **DSLP model, 7.6 million DOF** GEOMETRY COURTESY SVS FEM S.R.O.

- *For nonlinear analyses, use the NCNV command to obtain initial statistics of the solution.* Instead of waiting for a long nonlinear analysis to complete to determine if the solution is compute-bound or I/O-bound, use the `NCNV,,,3` command to stop the analysis after three iterations. (In ANSYS Mechanical with ANSYS Workbench, insert a “Commands (APDL)” object under the analysis branch and add this command.) The .PCS and .BCS files will reflect the solver time for a single iteration, so multiply this value by three to get a rough estimate of the time spent by the solver.

- *Avoid use of Lagrange multipliers.* ANSYS GPU acceleration does not support Lagrange multipliers, which typically arise in three areas: “normal Lagrange” contact, “mixed u-P” formulation, and joints/MPC184 elements. For contact elements, use augmented Lagrange or other contact algorithms. For hyperelastic materials, add some small compressibility (real-life materials are not fully incompressible anyway) to avoid using the mixed u-P formulation. For joints, consider if the joint can be replaced with remote points and constraints or constraint equations. For example, a body-to-ground revolute joint could be replaced with a remote displacement support.

HARDWARE AND LICENSING REQUIREMENTS

Although this article makes reference to GPUs, not all graphics cards are supported for GPU acceleration. There are a few important requirements for using GPUs in any FEA solution:

- *Double-precision performance:* The Mechanical APDL solver uses double-precision for its calculations for accuracy, but not all GPUs can handle double-precision computations efficiently.

State-of-the-art hardware can be utilized to provide the fastest solution times.

- *Adequate graphics memory:* The benefit of GPU acceleration is most evident on large models, for which computational costs are high. Because of this, the graphics card should have at least 5 GB to 6 GB of memory.

- *High memory bandwidth:* Iterative solvers, such as the PCG and JCG solvers, require high GPU memory bandwidth for optimal performance.

For the above reasons, only the NVIDIA Quadro 6000 and Tesla series cards listed in the ANSYS Mechanical APDL 14.5 help system are supported with GPU acceleration. Although the Tesla K10 and Quadro K5000 are also supported, they are recommended only for the PCG and JCG iterative solvers, since the peak double-precision performance is low but the memory bandwidth is fast.

To use GPU acceleration, you also need an ANSYS HPC Pack license — at release 15.0, GPU acceleration will be enabled through all ANSYS HPC product licenses (ANSYS HPC, ANSYS HPC Pack, ANSYS HPC Workgroup and ANSYS HPC Enterprise).

SUMMARY

GPU acceleration provides immediate benefits for single-workstation or cluster configurations. For older machines, use of GPU acceleration can effectively decrease solution time prior to upgrading hardware and, once a new workstation is obtained, the GPU can be reused in the new system. For newer hardware, GPU acceleration provides immediate benefits, including the ability to use more than one GPU for the iterative solver in a single machine.

ANSYS was one of the first commercial FEA vendors to introduce support of GPU computing. Continual investments and enhancements are made to the solver to ensure that state-of-the-art hardware can be utilized to provide the fastest solution times.