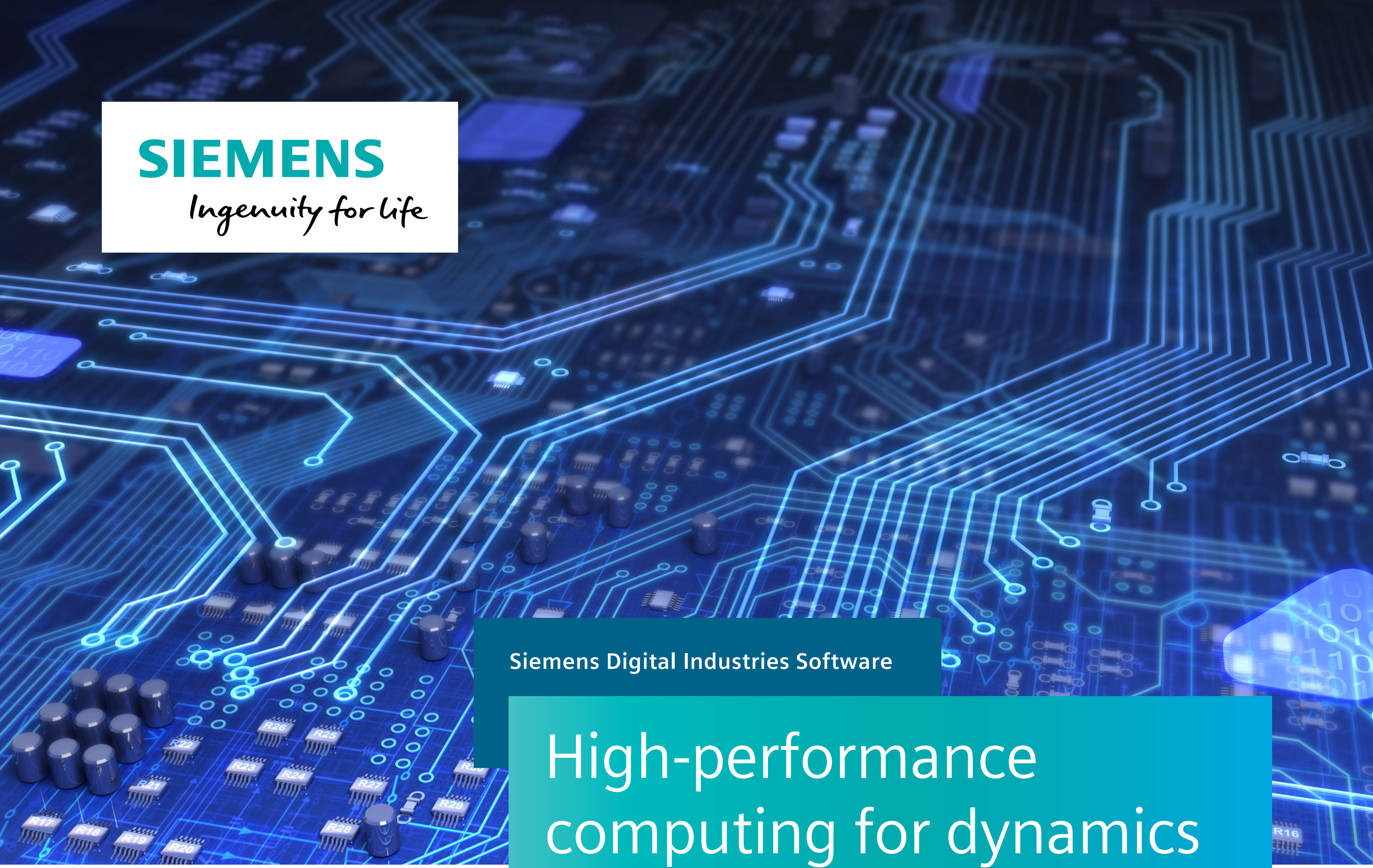

Siemens Digital Industries Software

# High-performance computing for dynamics and NVH analysis

## Simcenter Nastran improves performance for modal and dynamic response analysis


### Executive summary

This white paper describes recent enhancements to Simcenter™ Nastran software for modal and dynamic solution sequences and provides insights on the software and hardware factors that influence the performance of dynamic response and noise, vibration and harshness (NVH) analyses. The goal is to enable users to obtain good performance with existing hardware and to select the appropriate hardware for the problem at hand.



Easwaran Viswanathan, Senior Product Manager
Ben-Shan Liao, Senior Software Engineer
Wei Zhang, Senior Software Engineer
Siemens Digital Industries Software

# Contents

**1 Introduction ... 3**

**2 Parallel processing ... 3**
2.1 Shared memory parallelism ... 3
2.2 Distributed memory parallelism ... 4

**3 Direct frequency response analysis ... 5**

**4 Normal modes analysis ... 7**
4.1 Lanczos method ... 7
4.2 RDMODES ... 9

**5 Modal frequency response ... 11**

**6 Recommendations for hardware configuration ... 12**
6.1 Understanding memory usage ... 12
6.2 RAM recommendations ... 15
6.3 I/O and (scratch) disk recommendations ... 15
6.4 CPU recommendation ... 15
6.5 Recommended configurations ... 15

## 1 Introduction

Simcenter Nastran is a well-known finite element program for analyzing a wide variety of linear, nonlinear, and multidisciplinary problems. Siemens Digital Industries Software continuously focuses on enhancing Simcenter Nastran by providing discipline-specific extensions, ease-of-use, and process improvements.

As model complexity and model size continue to grow and time-to-market becomes increasingly important, solver performance is a key element that makes Simcenter Nastran the solution of choice for users who need to solve today's increasingly large dynamics and noise, vibration and harshness (NVH) problems.

Recent versions of Simcenter Nastran have therefore strongly focused on performance enhancements for the various modal and dynamic solution sequences. The main highlights are:

- The performance of direct frequency response (SOL 108) - enhanced distributed memory parallel (DMP) support and multifrontal massively parallel sparse solver (MUMPS) support.
- New recursive domain normal modes analysis (RDMODES) approach based on shared memory parallel (SMP) architecture and with improved stability, performance and scalability - over 20 to 50 percent reduction in elapsed time with a good scaling to 32 processors on a single node.
- Improved modal frequency response performance - automatic fast frequency algorithms for low-rank and high-rank damping problems.

## 2 Parallel processing

Parallel processing is commonly used to perform complex tasks and computations. A complex task is broken down into smaller, independent parts that can be executed simultaneously by multiple processors. The resources for parallel processing can be either a single node or a set of nodes on a compute cluster.

Simcenter Nastran supports two types of parallelism: shared memory parallelism (SMP), and distributed memory parallelism (DMP). The following sections introduce the two types of parallelism. Subsequent sections describe the effect of parallelism on Simcenter Nastran performance for dynamics and NVH.

### 2.1 Shared memory parallelism

With shared memory processing, parts of Simcenter Nastran that are performance-critical, for example, matrix factorization, are spawned off as threads or tasks. These threads are then run concurrently on the available processors. Threads communicate with each other through global memory and shared hardware resources. By design, shared memory parallelism works only on a single node.  Simcenter Nastran does not require any additional licenses to take advantage of shared memory parallelism as it is part of the Simcenter Nastran Basic package.

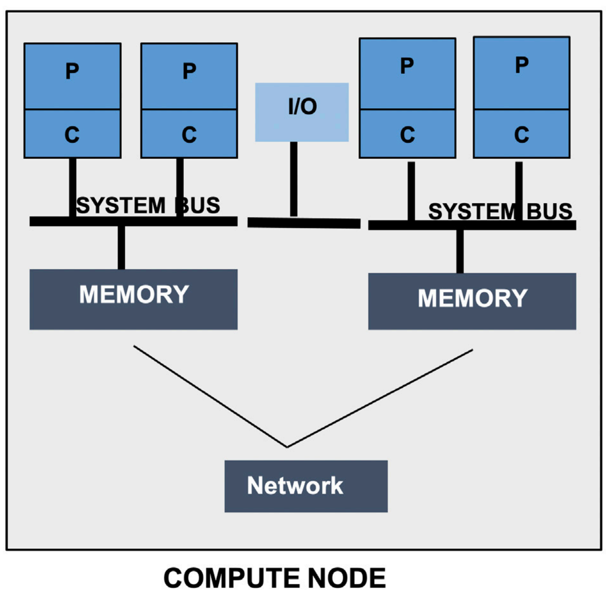


Figure 1: Shared memory architecture.

## 2.2 Distributed memory parallelism

With distributed memory processing, multiple instances of Simcenter Nastran are launched (higher-level parallelism) as opposed to a single instance in the SMP mode. These instances perform a portion of the task and exchange data by sending and receiving messages using the message passing interface (MPI).

On a compute cluster, multiple instances of Simcenter Nastran are launched either on a single node or a set of nodes on the cluster. Since distributed memory parallelism instances can be run on multiple hosts, one can take advantage of computing resources available across these hosts. Distributed memory parallelism can also be used in a hybrid mode with MPI communication across compute nodes and shared memory parallelism for intra-node communication.

Support of DMP for dynamic analysis in Simcenter Nastran requires the Simcenter Nastran Advanced Dynamics package or the Simcenter Nastran DMP license.

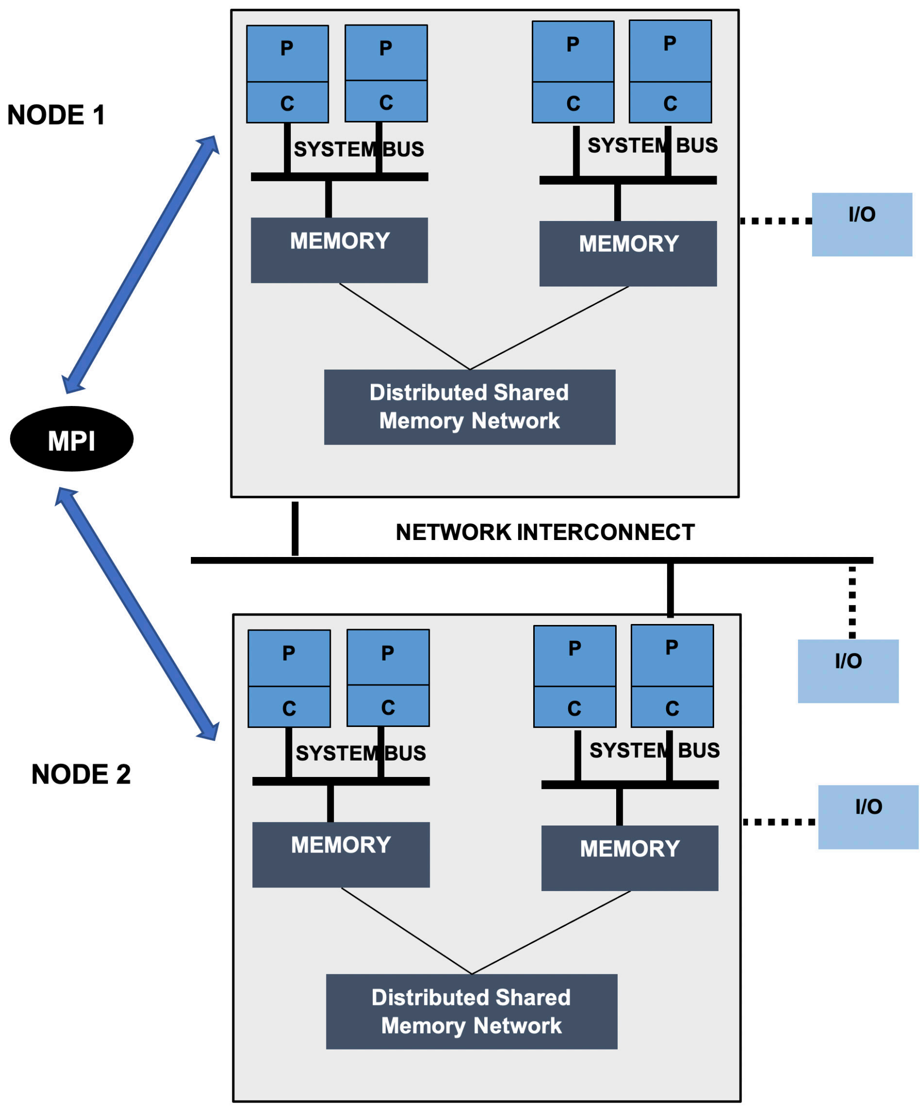


Figure 2: Distributed memory architecture.

## 3 Direct frequency response analysis

Frequency response analysis in Nastran can be performed using either the direct or modal methods. While modal methods are preferred for dynamic analysis of structural problems, direct frequency response is often needed to check the validity or accuracy of modal solutions. Also, direct frequency response is used for root cause analysis, especially at frequencies resulting in response peaks.

With direct frequency response, equations are assembled and solved at every excitation frequency. The more the number of frequencies or the larger the number of degrees of freedom, the greater the elapsed time, thus the need to consider various options available to speed up the solution time. Simcenter Nastran supports both SMP and DMP parallelization for direct frequency response.

When using DMP in direct frequency response analysis, excitation frequencies are distributed across Nastran instances. This works well when the DMP instances can divide the number of frequencies evenly. This means that for ideal scaling and best balancing, the number of frequencies is an even number of DMP instances.

On a high-performance computing (HPC) cluster, each node of the cluster can then process a set of frequencies using the compute resources (memory, disk, etc.) on that node. However, on a single node, the

- memory available for each DMP instance = maximum memory/N, where N is the value in DMP keyword or number of instances. If the model is large, the memory available per instance might not be enough to do a matrix factorization.
- matrix factorization disk requirements = N * I/O requirements per instance.

The above two resource requirements limit the DMP scalability on a single node. The optimal number of DMP ranks that can be supported will then depend on the disk configuration and I/O throughput. The larger the I/O throughput, the greater the number of DMP ranks that can be supported with good scaling. As a rule of thumb, up to 2 instances (ranks) can be supported by faster small computer system interface (SCSI) or serial-attached SCSI (SAS) disk drives (mechanical drives), and up to 4 instances can be supported by a single solid-state drive (SSD) using non-volatile memory (NVMe) or SAS controllers.

Figure 3 shows the scaling of direct frequency in DMP mode for a full vehicle NVH model with 7.47 million degrees of freedom. Details of the hardware used are as follows:

- Processor: Intel Xeon E5-2667
- Number of sockets: 2
- Number of cores per socket: 8
- Memory: 16 GB/core (total memory = 256 GB)
- Number of disk drives for Nastran scratch: 5 SSD disk drives in RAID 0 configuration

This figure shows good scaling up to DMP=4 and levels off beyond DMP=8 as the I/O and memory resources are shared between the various Nastran instances as the job is run on the single node.

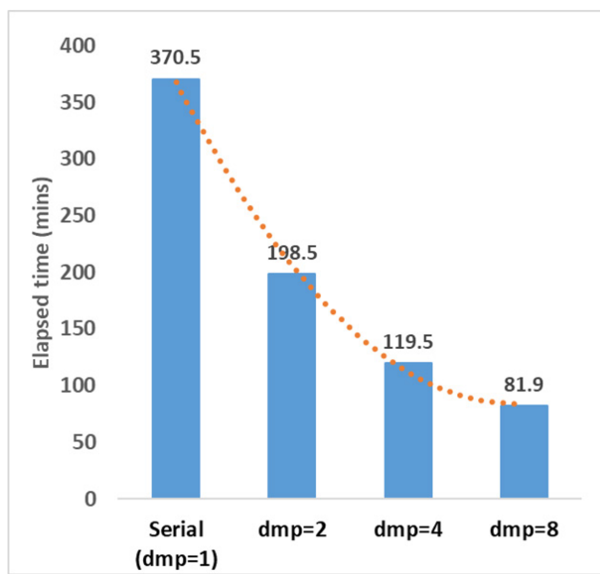


Figure 3: Effect of DMP ranks on elapsed time on a single node for a full-vehicle NVH model with 7.47 million DOF.

Figure 4 shows the influence of solver selection on elapsed time. The selection of MUMPS as the solver reduces the elapsed time significantly. MUMPS is the default solver in the Simcenter Nastran 2020.2 release for frequency response when the problem is dominated by solid elements. With the Simcenter Nastran 2021.2 version, the MUMPS solver is the default for direct frequency response problems regardless of the type of elements present in the model.

Figure 5 shows Nastran performance data for the same model using shared memory parallelism. At each frequency, the linear equations are solved with shared memory processing. The linear equation solver then makes use of multiple processors. This directly contrasts with DMP where excitation frequencies are distributed to various processors. The scaling in SMP mode can be seen to taper off as more processors are used. The scaling often depends on the size of the problem, that is, the number of degrees of freedom in the model. The larger the number of degrees of freedom, the better the utilization of multiple processors.

Direct frequency response can also benefit from SMP scaling when run in DMP mode, that is, in hybrid mode. When DMP and SMP are combined, frequencies are divided between the different Nastran DMP instances.
At each frequency, SMP processors are then used for solving the linear system of equations.

Figure 6 illustrates the performance of the previous model when using the combination of SMP and DMP. As seen from the figure, the hybrid mode performs better than using either DMP or SMP. As mentioned previously, the number of DMP ranks is limited by the hardware resources.

The rule of thumb is to use as many processors for DMP as permitted by the I/O resource and memory. The remaining processors can then be used for SMP.

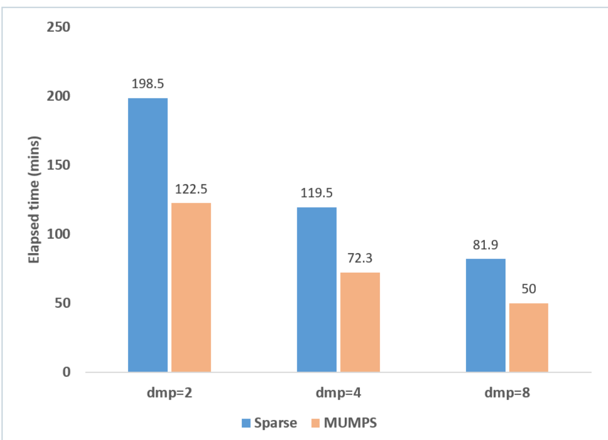


Figure 4: Effect of solver choice on DMP (single node performance data).

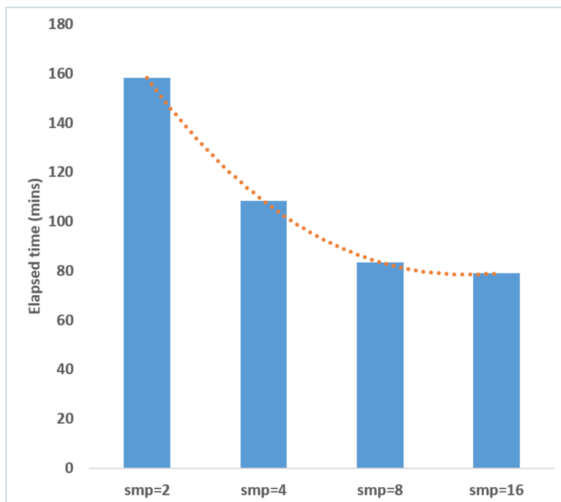


Figure 5: Effect of SMP on elapsed time.

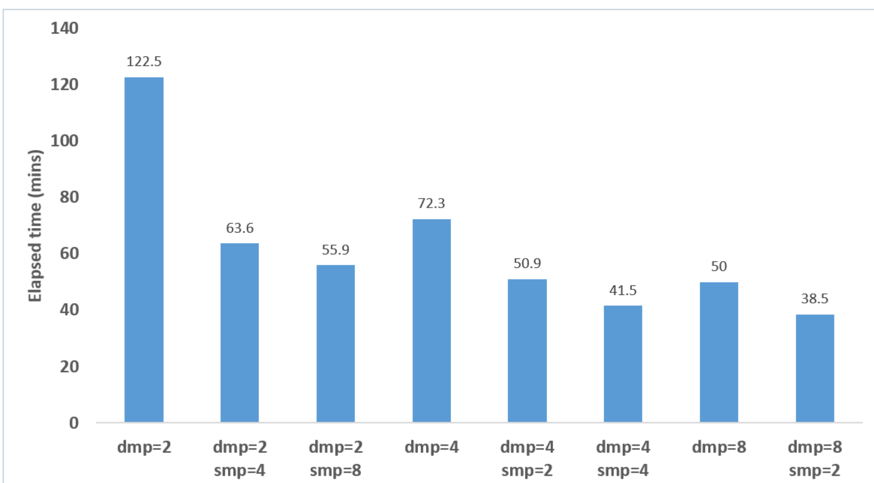


Figure 6: Elapsed time using a hybrid SMP and DMP on a single compute node.

## 4 Normal modes analysis

Structural analysis or vibro-acoustic analysis problems typically tend to be large, that is, have large numbers of degrees of freedom. These are often analyzed by converting the problem from physical to modal space – modal analysis is often used instead of direct frequency response. A prerequisite for a modal response analysis or modal analysis is the computation of normal modes or eigenvalues.

Two main methods are available in Simcenter Nastran to compute the normal modes:

- Lanczos block solution
- Recursive domain normal modes (RDMODES)

### 4.1 Lanczos method

The default method to compute normal modes in Simcenter Nastran is the Lanczos block solution. For very small problems, Simcenter Nastran offers other methods, such as Householder, modified Householder and Givens method. The Lanczos normal mode solution supports both the DMP and SMP forms of parallel processing. Figure 7 shows how elapsed time changes as the number of processors is increased. The data was generated for a powertrain model with 14.8 million degrees of freedom.

When used with DMP, Simcenter Nastran will, by default, partition the system matrices and distribute the partitioned matrices to the different DMP ranks or instances. As the number of processors for DMP is increased, the elapsed time can be seen to decrease as expected. However, beyond DMP=2 the scaling on a single node is rather limited. As the number of processors for DMP is increased, the DMP ranks compete for the same resources, for instance the I/O and memory. The matrix partitioning does not scale very well when the number of processors is increased. Also, we can see from Figure 7 that use of SMP parallelism has comparatively less influence on elapsed time (compared to DMP).

Simcenter Nastran allows the user to split the frequency range of interest between processors in a Lanczos normal mode solution. Users can specify how the DMP ranks need to be split via the Simcenter Nastran keywords "gdoms" and "fsegs." Here, the keyword "gdoms" is used to specify the number of matrix partitions and the keyword "fsegs" for the number of frequency partitions. Figure 8 shows the schematic of such a partitioning. The illustration shows "j" frequency partitions and four geometry partitions.

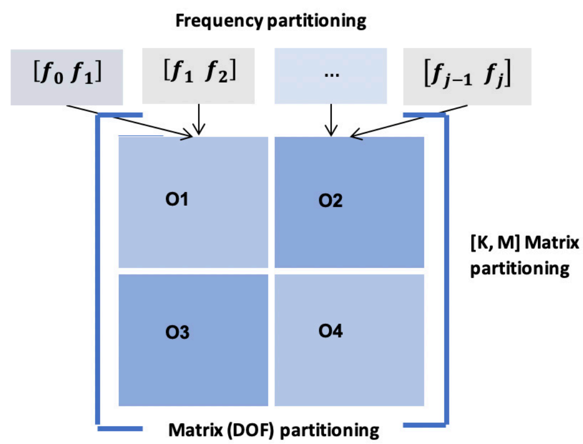


Figure 8: Partitioning of matrices and frequency. In this example fsegs=j and gdoms=4.

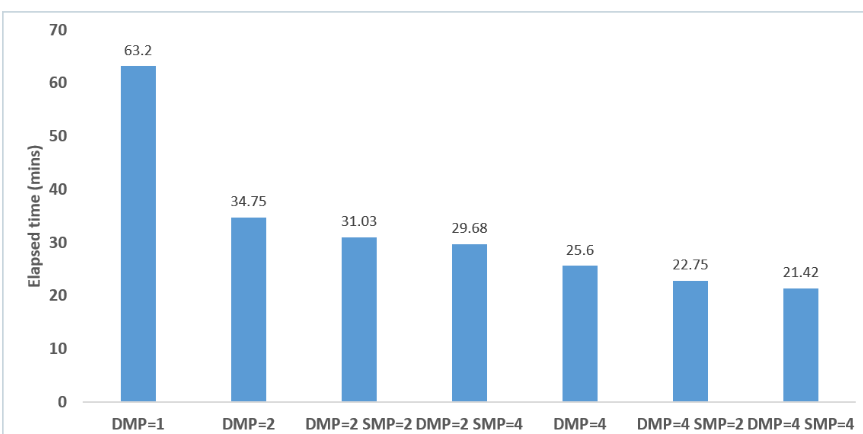


Figure 7: Effect of DMP ranks on elapsed time for normal mode solution for a powertrain model on a single node.

For example,
**nastran filename.dat mem=64gb dmp=4 gdoms=2 fsegs=2**

The product of “gdoms” and “fsegs” keywords must be equal to the number of processors specified by the dmp keyword. If “gdoms” and “fsegs” are both specified in the command line, the “dmp” keyword becomes optional. The default in Simcenter Nastran is to do matrix partitioning.

Figure 9 shows the effect of frequency partitioning as the frequency range of interest of normal modes increases. As seen from the figure, the benefits of frequency partition become more pronounced as the frequency range increases.

In summary, DMP ranks become more effective if the matrices and frequencies are both partitioned between the DMP ranks. Again, a good rule of thumb is to use as many processors for DMP as permitted by the I/O resource and memory. The remainder of the processors can then be utilized for SMP processing. If there are more modes in the frequency range, increasing the frequency partitions (as opposed to geometry partitions) will help with performance. Similarly, if the number of degrees of freedom is large and only a few modes exist in the given frequency range, increasing the matrix partition instead of frequency partitions will help reduce the elapsed time.

Figure 10 shows how the elapsed time changes when the upper bound frequency limit for normal modes increases. The cost of computation is no longer linear; that is, the trendline shows a polynomial behavior (order 4). Also, since Lanczos works with global matrices, disk I/O increases rapidly as the frequency range of interest increases. Therefore, good disk I/O performance is of paramount importance for modal computation using the Lanczos method.

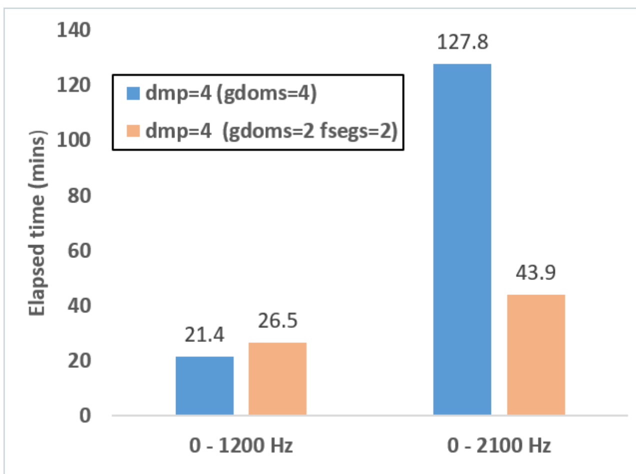


Figure 9: Effect of frequency partitioning on elapsed time for two different frequency ranges on a single compute node.

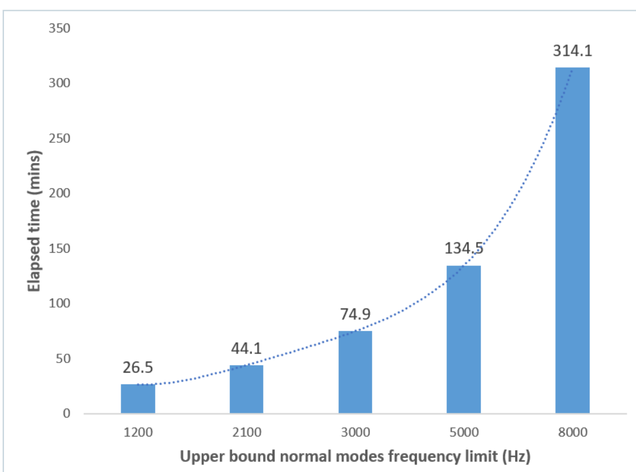


Figure 10: Elapsed time as a function of upper bound frequency limit for normal modes (gdoms=2 fsegs=2) on a single compute node.

### 4.2 RDMODES

As previously described, when the number of degrees of freedom becomes greater than a few million and the frequency range of interest has many modes, the Lanczos block eigen solution methodology becomes computationally very expensive, and in some instances such as full vehicle NVH, not practically viable. The preferred method is to use RDMODES for large models. RDMODES is part of the Simcenter Nastran Advanced Dynamic package.

As noted earlier, Lanczos block solution can be used when only a few modes need to be computed, that is, less than 100 modes (rule of thumb). If the model is smaller (just a few million degrees of freedom), this limit can be increased.  Also, Lanczos does not introduce any approximations and is still a preferred solution for smaller problems.

RDMODES is a smart form of component mode synthesis. The global eigenvalue problem is partitioned into smaller components and the corresponding component modes are computed. Component modes are then assembled, and the system modal solution is computed.

RDMODES supports both DMP and SMP, but the best performance for RDMODES is SMP if using Simcenter Nastran 2020.1 or higher. Performance of RDMODES has been and vastly improved in the last several releases.

Figure 11 shows the comparison of RDMODES and Lanczos when the upper bound frequency range of interest is varied. As can be seen, RDMODES significantly outperforms Lanczos as the frequency range of interest increases.

Figure 12 shows the same plot as in figure 11 but with just RDMODES data. The goal is to look at the cost of computation when requesting more modes. The cost of requesting more modes with RDMODES can be seen to be just a linear function.

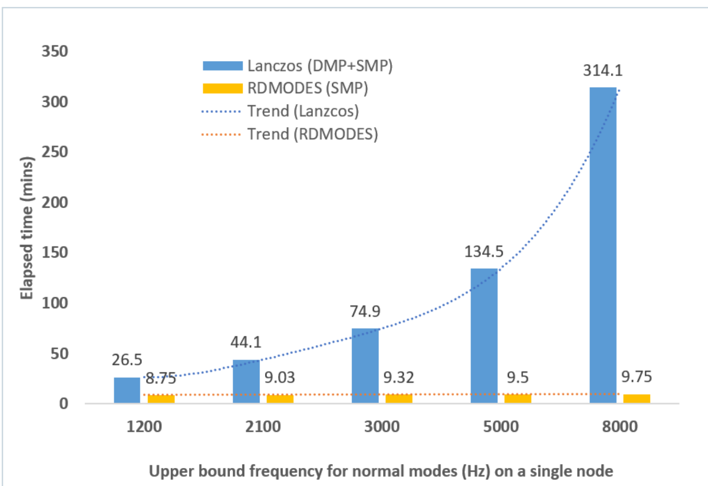


Figure 11: Comparison of elapsed time for Lanczos and RDMODES using 16 processors on a single node.

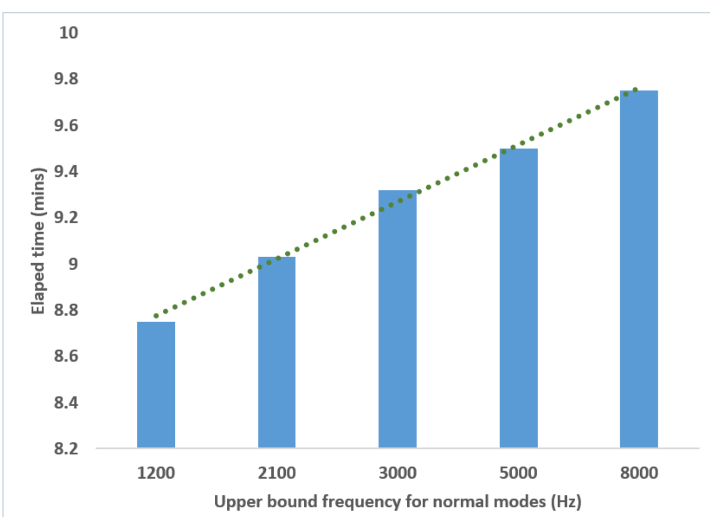


Figure 12: Elapsed time for RDMODES with varying upper bound frequency limit.

RDMODES can be triggered by the RDMODES bulk data entry, which is a companion entry for EIGRL bulk data. The RDMODES entry allows the user to specify the method for partitioning the model into components. It also allows the user to specify the ratio of component modes to global modes (RDSCALE).

Figure 13 shows the performance of RDMODES with SMP parallelism for a trim body model with 18 million degrees of freedom resulting in about 3,600 modes. The figure also shows the performance of RDMODES as a function of release.

Figure 14 shows improvement in performance with RDMODES for different NVH models over different Simcenter Nastran releases.

| | Number of degrees of freedom (in millions) | Number of modes in the frequency range |
|---|---|---|
| **Model 1** | 13.4 | 4200 |
| **Model 2** | 11.5 | 5500 |
| **Model 3** | 14.9 | 5200 |
| **Model 4** | 30.3 | 5300 |
| **Model 5** | 22.2 | 5400 |

This figure also shows improvements being made to RDMODES functionality in each release.

In summary, for large problems with many modes, it is more efficient to use RDMODES for modal computation. RDMODES also scales well with SMP and is hence preferred over DMP.

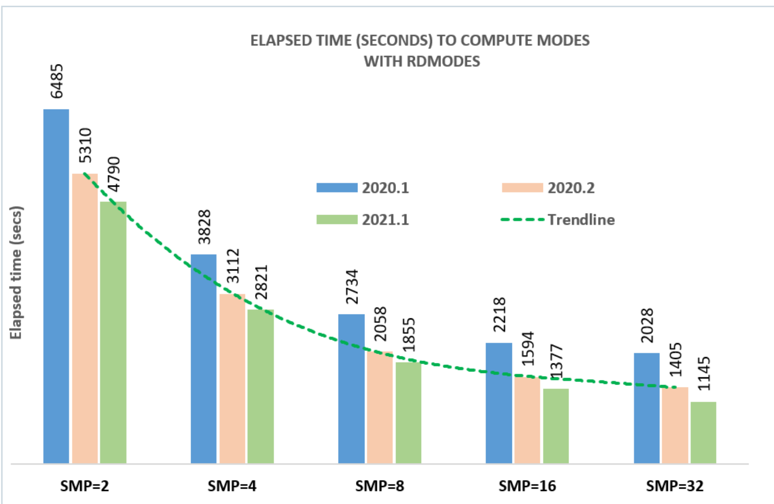


Figure 13: Performance of RDMODES with SMP.

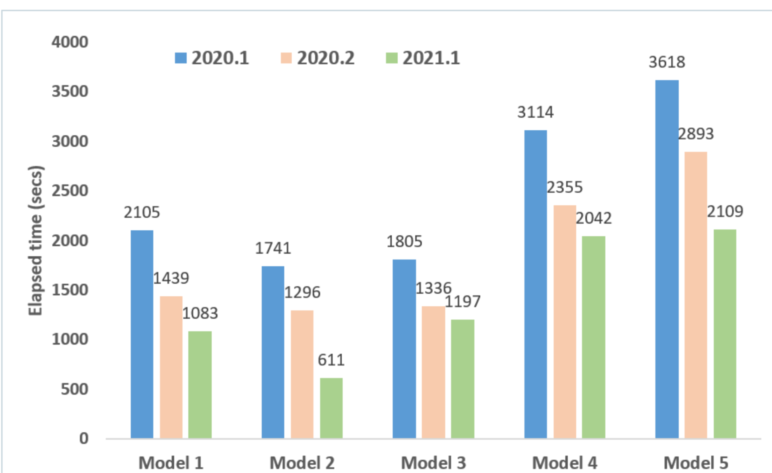


Figure 14: RDMODES performance – various models.

## 5 Modal frequency response

One of the advantages of modal frequency response is that the model is reduced to the modal degrees of freedom. Modal frequency response consists of two phases: computation of modes and determining the response. Since the modal computation part of the frequency response solution is the same as that for normal modes, the discussion here will focus mainly on response computation.

When there is no damping in the model or when the damping is defined by modal damping, the equations in modal degrees of freedom are all uncoupled. The solution of uncoupled equations is inexpensive, that is, significantly less compared to the overall elapsed time. On the other hand, when structural viscous damping is present, or vibro-acoustic coupling exists in the model, the equations in the modal domain are coupled. The degree of coupling is defined by the extent of damping – the more spatial the damping, the more the degree of coupling of the modal degrees of freedom.

Simcenter Nastran has fast frequency response solution algorithms implemented to handle low-rank and high-rank coupling – the best method is automatically selected for the user. Significant algorithmic performance improvements have been made for frequency response performance in the last few releases for high-rank damping to make it as competitive as for low-rank damping, for which performance improvements were made a few releases ago.

Figure 15 depicts the frequency performance of a trim body model with approximately 16.8 million DOFs.  Since structural damping is defined on the entire model (high-rank), the rank of the structural damping matrix is the same as the number of modes.

The performance can be further improved by distributing the frequencies via DMP (refer to Section 2 above on direct frequency response). Since the problem is smaller compared to SOL108 (modal degrees of freedom versus physical degrees of freedom), the use of SMP will not scale as much as direct frequency response does. The rule of thumb is to use SMP in a SOL111 run with RDMODES as modal computation is the dominant portion of the total computation time.

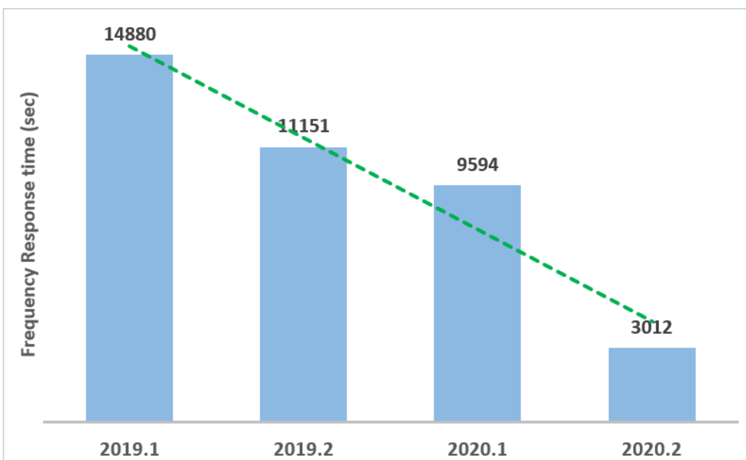


Figure 15: Frequency response performance improvements.

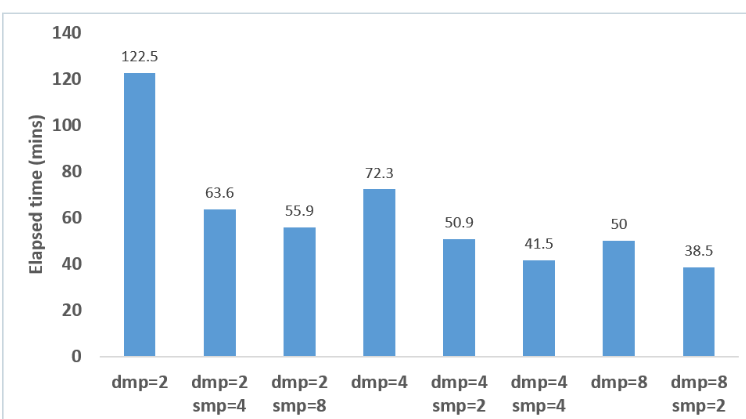


Figure 16: Elapsed time for hybrid SMP and DMP using a single compute node.

## 6 Recommendations for hardware configuration

To maximize the performance of Simcenter Nastran, it is helpful to understand how hardware choices can affect performance. Hardware with a proper balance of available memory, disk and compute resource will result in reduced elapsed time compared to other hardware that is not properly tuned for Nastran simulation.

### 6.1 Understanding memory usage

Memory is needed for the following tasks (figure 17):

- Nastran as an application – providing an appropriate amount of memory to help Nastran perform many operations in memory (in-core) as opposed to out-of-core.
- Operating system I/O cache – operating systems typically cache the disk reads and writes (I/O) to avoid kernel and hardware overheads for individual I/O operations. The fewer the read/writes, the better the application throughput.
- Operating system memory – memory for the operating system to perform its essential or critical operations.

With Simcenter Nastran the user can subscribe to a maximum of 80 percent of physical memory, leaving 20 percent of physical memory for the operating system as well as for I/O cache. The amount of memory to be used by Nastran can be specified in either the command line or in the Nastran runtime configuration file (RCF) using the "mem" keyword.

Memory specified for Nastran by "mem" keyword is internally split, as shown in figure 18.

- Core memory (also known as open core in Nastran) – available to all modules to perform computations.
- Buffer pool memory – used to temporarily buffer I/O (read and writes). For some large problems where lot of data is held in temporary buffers, the buffer pool memory helps reduce the elapsed time. Buffer pool memory can be set by "buffpool"
- Scratch memory – used to buffer all scratch read/writes. Scratch memory can be set by "smem" keyword.

Numeric solutions such as matrix factorization in Nastran require open core memory. The I/O operations utilize buffer pool and scratch memory – this is analogous to how the OS performs its own I/O cache. By default, scratch and buffer pool are set to 20 percent of total memory specified using the "mem" keyword. For problems requiring large matrix factorizations, increasing the open core at the expense of scratch and buffer pool memory might be necessary. Finding the right balance between scratch memory, buffer pool and open core in cases where both the model size and data recovery are large is often challenging. It is recommended to use the defaults as set in Nastran. In cases where numeric computations require more memory, the scratch and buffer pool memory can be reduced, paving the way for more memory to core modules.

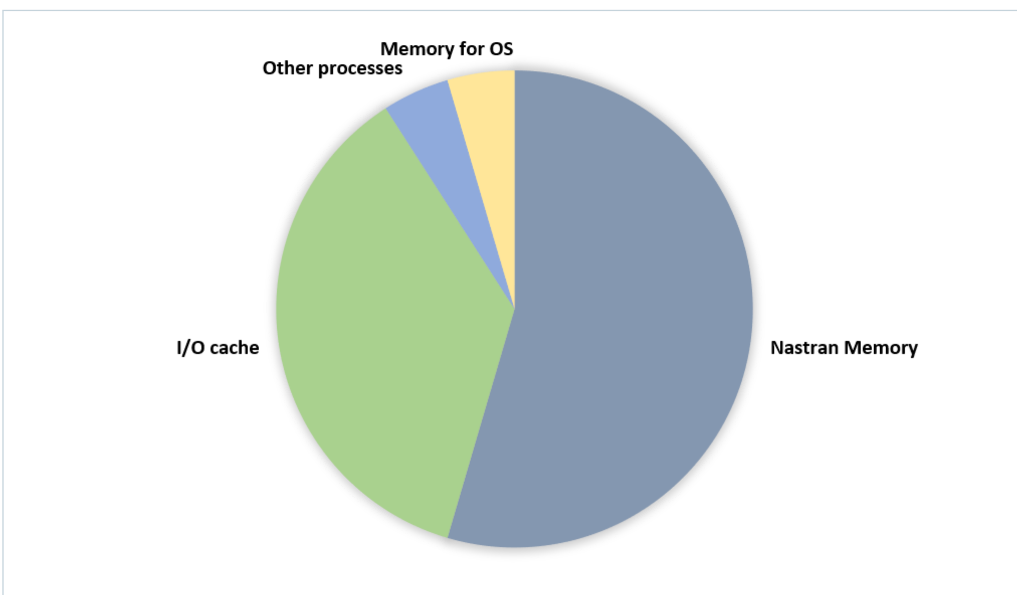


Figure 17: Typical memory distribution

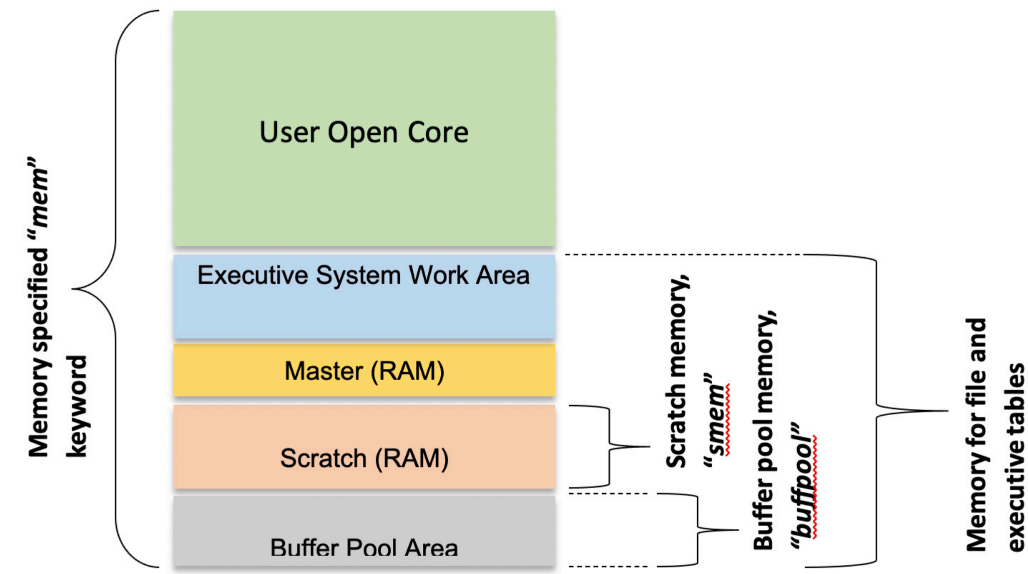


Figure 18: Nastran internal memory allocation.

Figure 19 shows the "USER INFORMATION MESSAGE 4157" output to the "F04" file. This message contains two important pieces of data:

- Minimum (absolute minimum) amount of memory required to factorize a given matrix
- Minimum amount of memory required to avoid disk spillage.

Figure 19 also shows the effect of memory on matrix factorization for different models. Three different memory ranges are used:

- Memory less than the memory required for disk spillage. In this range, any additional memory specified goes to reduce the disk spillage, thereby improving the performance.
- Memory greater than the threshold to avoid disk spillage. In this range, any additional memory specified doesn't impact the elapsed time. The additional memory is not used by Nastran for matrix factorization.
- Memory much greater than that required for the disk spill. In this case, much more memory than what is required is specified. In some cases, this can lead to increased elapsed times as it is possible that the operating system is being starved of memory that can be used for caching the read/writes.

```
*** USER INFORMATION MESSAGE 4157 (DFMSYN)
    PARAMETERS FOR SPARSE DECOMPOSITION OF DATA BLOCK  KLL     ( TYPE=RDP ) FOLLOW
                     MATRIX SIZE =   70345 ROWS            NUMBER OF NONZEROES =  2701957 TERMS
          NUMBER OF ZERO COLUMNS =       0              NUMBER OF ZERO DIAGONAL TERMS =       0
                CPU TIME ESTIMATE =   78216 SEC               I/O TIME ESTIMATE =      25 SEC
       MINIMUM MEMORY REQUIREMENT =    1364 K WORDS         MEMORY AVAILABLE =   32615 K WORDS
     MEMORY REQR'D TO AVOID SPILL =   12305 K WORDS        MEMORY USED BY BEND  =    3651 K WORDS
     EST. INTEGER WORDS IN FACTOR =    87006 K WORDS          EST. NONZERO TERMS =   174758 K TERMS
```

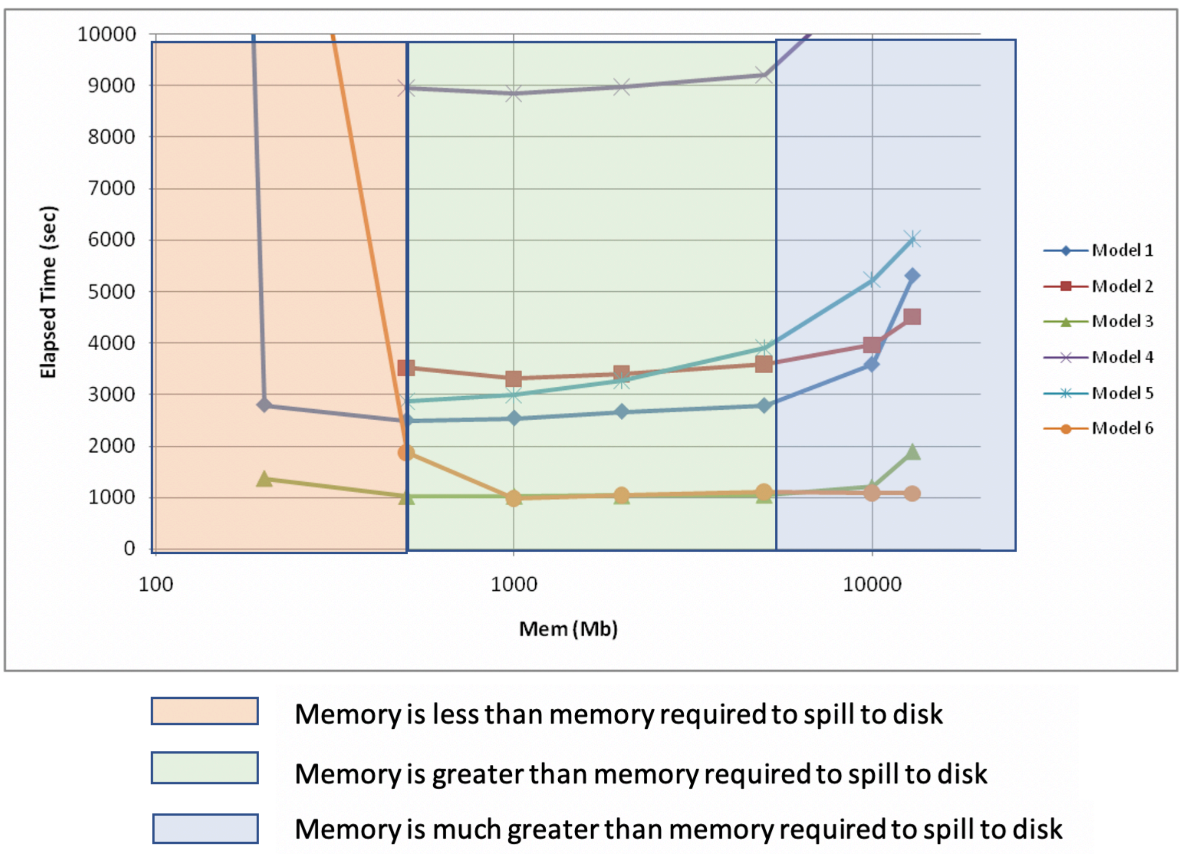


Figure 19: Nastran matrix factorization performance - various memory settings.

### 6.2 RAM recommendations

The most important factor in determining performance is the amount of random access memory (RAM) on the system. Increasing dynamic RAM (DRAM) allows a larger portion of the problem to be solved in-core where memory access speeds are orders of magnitude faster than the I/O accesses required for out-of-core solution. A good rule of thumb is that the system should have at least 6 GB (and preferably 8 GB or more) of DRAM per physical core for optimal performance. For large models, at least 256 GB of total system memory is recommended.

### 6.3 I/O and (scratch) disk recommendations

The second most important determiner of Nastran performance (after memory) is I/O subsystem performance. When a Nastran computation runs out of RAM memory, then Nastran reads and writes (scratch) data to disk, and since Nastran is very I/O intensive it is important to select an appropriate disk drive for Nastran scratch files.

Two important I/O system choices will affect Nastran performance: disk type and disk configuration.

Disks can be either mechanical, for example, SATA, SCSI or SAS drives, or non-mechanical, such as solid-state disk drives and optical disks. Non-mechanical drives typically allow higher throughput and have lower disk latency for accessing data and hence are preferred for Nastran scratch file storage.  The best choice for disk type is an SSD using an NVMe interface, followed by SSD with SAS interface. Higher throughput can also be achieved if the disks are set up in a RAID configuration, preferably RAID 0 configuration.

Some additional recommendations:

- Avoid using network disks for Nastran scratch files.
- If mechanical drives are to be used, choose the ones that have higher rotation speed (e.g., 15K RPM) and large buffer size models.
- For Linux systems, use second extended file (EXT2) or XFS file system for scratch disks. If EXT4 must be used, turn off journaling. Journaling file systems replicate I/O to provide fail-safe or fault-tolerant systems and hence must be turned off for scratch file systems.

### 6.4 CPU recommendation

Processor choice is the final consideration for hardware decisions. Some of the Nastran modules are computationally intensive. For large dynamics analyses, Nastran can generally scale to at least 16 processors, so a hardware configuration with two 12-core sockets is a general recommendation.

The major contributors to Nastran performance are the processor speed (base frequency), the core count and the cache size. In general, multiple moderate core count processors – e.g., two sockets, each with 20 cores running at 2.5 GHz – represent a better choice than a single high core count processor – e.g., one socket with 48 cores running at 2.3 GHz.

Further, when faced with a choice between a better processor and additional memory, then additional RAM is usually the right decision.

Some additional recommendations:

- Hyperthreading is not supported by Nastran and hence this should be turned off in the BIOS settings. If it is enabled, there is a risk of oversubscribing the processors.
- Some of the operating systems throttle the CPU clock speed to conserve energy. This should be turned off either in the BIOS or using the OS settings.

### 6.5 Recommended configurations

Hardware configurations well-suited to Nastran can vary based on the analyses performed. Below are some recommended configurations, roughly based on currently available processors and cloud instance types.

| Description | Sockets/Cores | Clock Speed (GHz) | RAM (GB) | Scratch Disk (GB) | Comparable AWS Instance Type |
|---|---|---|---|---|---|
| Small | 1/8 | 3.0 | 64 | 1 × 600 | r5d.2xlarge |
| Medium | 1/16 | 2.8 | 128 | 2 × 600 | r5d.4xlarge |
| Large | 2/24 | 2.8 | 192 | 2 × 900 | r5d.12xlarge |
| Server/Cluster | 2/32 | 2.8 | 512 | 4 × 900 | r5d.16xlarge |

**Siemens Digital Industries Software**

**Headquarters**
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

**Americas**
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

**Europe**
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

**Asia-Pacific**
Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

## About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. The Xcelerator portfolio helps companies of all sizes create and leverage digital twins that provide organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on LinkedIn, Twitter, Facebook and Instagram. Siemens Digital Industries Software – Where today meets tomorrow.

**siemens.com/software**



83542-C4 3/21 A