

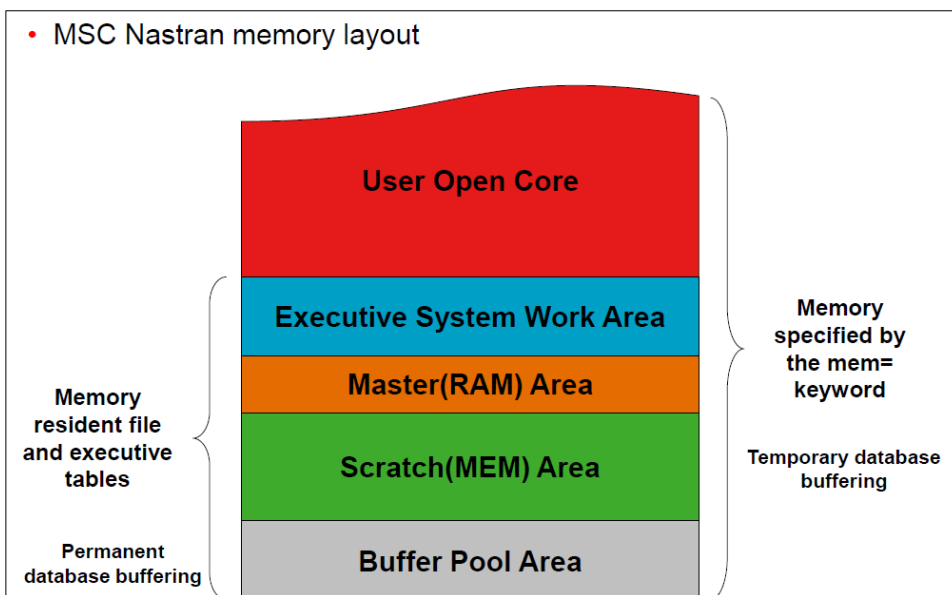


## MSC Nastran

Performance in statischen Analysen steigern

### Welche Möglichkeit gibt es, die Performance von MSC Nastran in einer statischen Analyse zu steigern?

Es gibt gemeinsame aber auch unterschiedliche Einstellungen der MSC Nastran Parameter für statische und dynamische Analysen. Dabei ist es zuerst wichtig die Memory (Hardware RAM) Einteilung von MSC Nastran zu verstehen:



Diese Einteilung wird im .f04 File folgend angezeigt:

```

** MASTER DIRECTORIES ARE LOADED IN MEMORY.
USER OPENCORE (HICORE)      = 1037375360 WORDS
EXECUTIVE SYSTEM WORK AREA   = 1169591 WORDS
MASTER (RAM)                 = 200000 WORDS
SCRATCH (MEM) AREA           = 6553700 WORDS ( 100 BUFFERS)
BUFFER POOL AREA (GINO/EXEC) = 3277350 WORDS ( 50 BUFFERS)
TOTAL MD NASTRAN MEMORY LIMIT = 1048576000 WORDS
    
```

Alle Vorbelegungen sind mit default Werten eingestellt, die aber bei Bedarf überschrieben werden können. In diesem Beispiel wurde die Buffsize (Nastran system(1)) auf 65537 gesetzt und der maximal zulässige RAM auf 4 Gb bei einem 32 bit System oder 8Gb bei einem 64 bit Betriebssystem eingestellt. Diese Werte können während der MSC Nastran Installation gleich im Konfiguration File gesetzt werden.

Es gibt 2 Module die hauptsächlich den benötigten RAM bestimmen: 1.) DCMP in der Stastik und 2.) READ in der Modalanalyse. Deshalb empfiehlt es sich, besonders bei wiederkehrenden Jobs mit ähnlicher Größe einen Blick wiederum in das .f04 File zu werfen. Am Ende des Files steht folgende Statistik:

```

*** TOTAL MEMORY AND DISK USAGE STATISTICS ***
+-----SPARSE SOLUTION MODULES -----+
HIWATER SUB_DMAP DMAP
(WORDS) DAY_TIME NAME MODULE
2884693 14:09:30 SEKRRS 73 DCMP
    
```



## MSC Nastran

### Performance in statischen Analysen steigern

Hiwater ist der maximale RAM Bedarf für diesen Job. Außerdem wird noch der Zusatz:

```
** MASTER DIRECTORIES ARE LOADED IN MEMORY.
USER OPENCORE (HICORE) = 13969712 WORDS
```

mit dem für MSC Nastran zur Verfügung stehenden RAMs ausgegeben. Sollte nun dieser Wert kleiner als der Hiwater sein, wäre zusätzlicher RAM für eine optimalere Performance hilfreich.

Eine detailliertere Ausgabe zeigt das DCMP Modul ebenfalls im .f04 File:

```
8:05:48      2:54      77621.0          0.0          170.3          0.0      SEKRRS  83      DCMP      BEGN
*** USER INFORMATION MESSAGE 4157 (DFMSYM)
PARAMETERS FOR PARALLEL SPARSE DECOMPOSITION OF DATA BLOCK KLL      ( TYPE=RDP ) FOLLOW
MATRIX SIZE = 4287361 ROWS          NUMBER OF NONZEROES = 122689156 TERMS
NUMBER OF ZERO COLUMNS = 0          NUMBER OF ZERO DIAGONAL TERMS = 0
SYSTEM (107) = 32772          REQUESTED PROC. = 4 CPUS
CPU TIME ESTIMATE = 8627 SEC          I/O TIME ESTIMATE = 24 SEC
MINIMUM MEMORY REQUIREMENT = 122150 K WORDS      MEMORY AVAILABLE = 1037364 K WORDS
MEMORY REQR'D TO AVOID SPILL = 188241 K WORDS      MEMORY USED BY BEND = 122151 K WORDS
EST. INTEGER WORDS IN FACTOR = 685274 K WORDS      EST. NONZERO TERMS = 1273023 K TERMS
ESTIMATED MAXIMUM FRONT SIZE = 7056 TERMS          RANK OF UPDATE = 24
8:06:08      3:14      82366.0          4745.0          190.1          19.9      SPDC BGN TE=8627
8:09:50      6:56      97452.0          15086.0          1918.3          1728.1     SPDC END
*** USER INFORMATION MESSAGE 6439 (DFMSA)
ACTUAL MEMORY AND DISK SPACE REQUIREMENTS FOR SPARSE SYM. DECOMPOSITION
SPARSE DECOMP MEMORY USED = 188241 K WORDS          MAXIMUM FRONT SIZE = 7056 TERMS
INTEGER WORDS IN FACTOR = 46130 K WORDS      NONZERO TERMS IN FACTOR = 1273023 K TERMS
SPARSE DECOMP SUGGESTED MEMORY = 120393 K WORDS
```

Hier sind wiederum 3 Angaben besonders von Interesse:

- 1.) Memory available
- 2.) Memory to avoid spill und
- 3.) Sparse Decomp suggested memory.

Solange das Memory available größer ist als der Sparse Decomp suggested memory haben Sie bezüglich des RAMS eine optimale Performance. Sollte aber das Verhältnis schlechter sein, geht die Performance quadratisch zurück.

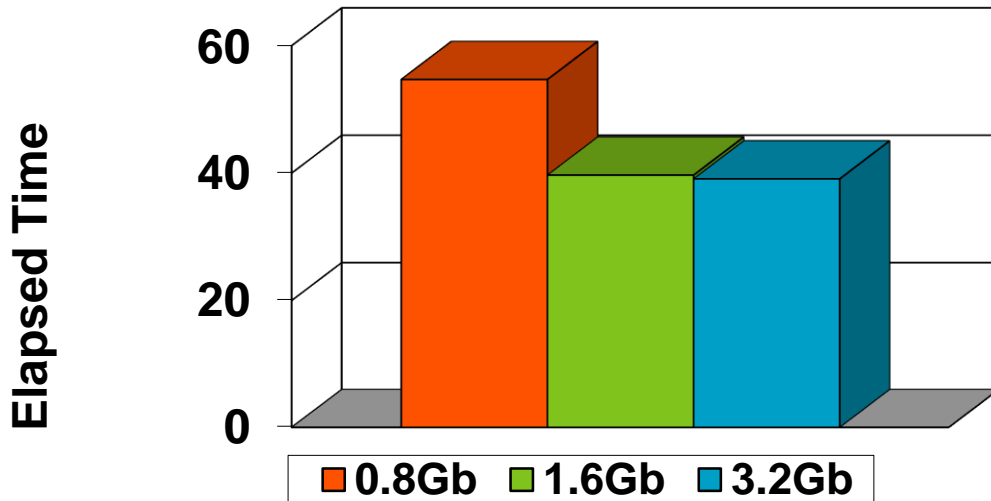
Beispiel: ein Lauf mit folgenden Angaben im f04 File:

- Memory Reqr'd to Avoid Spill = 2652 Mb
- Sparse Decomp Suggested Memory = 1560 Mb
- Abgeschickt mit folgenden 3 mem settings, 800Mb, 1.6Gb und 3.2Gb führt zu folgenden Rechenzeiten:



## MSC Nastran

Performance in statischen Analysen steigern

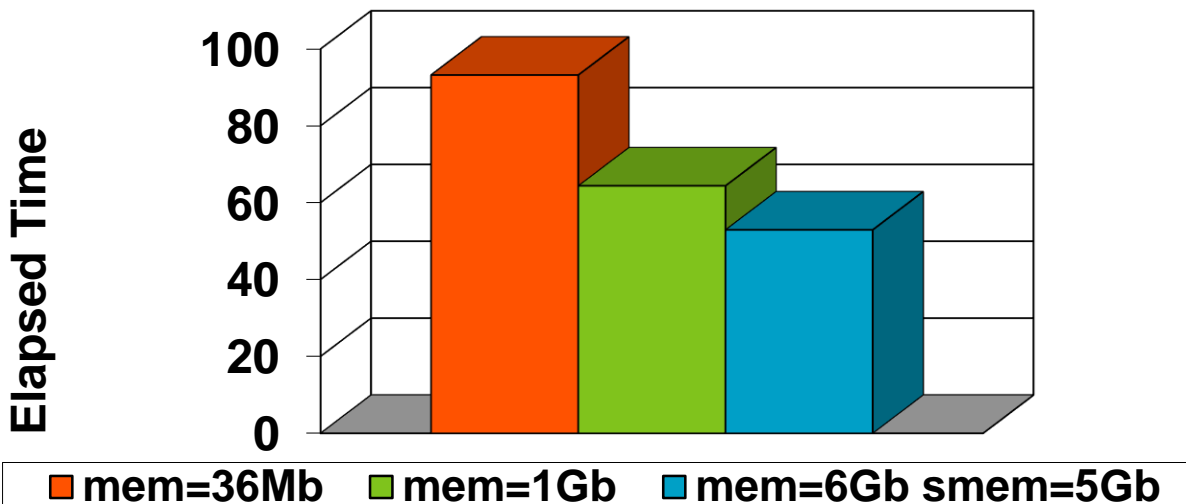


Wenn Sie Ihr Modell nun so gut kennen, gibt es die Möglichkeit zur Nutzung des „large Memories“. Hierbei wird zum Beispiel der SMEM Bereich (siehe oben) erweitert. Dies geht mit dem Keyword smem beim Abschicken des Jobs:

**Nastran *jobname* mem=6Gb smem=5Gb**

Jetzt reserviert MSC Nastran 5Gb von den maximalen 6Gb für den internen Scratch bevor auf die Platte geschrieben wird. Dies ist besonders vorteilhaft bei einem schlechtem I/O des Computers. Aber Vorsicht ! Der zur Rechnung nutzbare RAM ist jetzt nur noch 1 GB.

- Beispiel:
1. Zu wenig RAM, out-of-core solve with spill
  2. Sparse decomp suggested memory
  3. Sparse decomp suggested memory with 5Gb smem file





## MSC Nastran

### Performance in statischen Analysen steigern

Ab der MSC Nastran Version 2013 wurde dieser Schritt automatisiert wenn MSC.Nastran mit dem Keyword `mem=max` abschickt wird. Außerdem empfiehlt es sich, mit der 64 bit Version zu arbeiten, um mehr als 8 Gb RAM allokkieren zu können (keyword `mode=i8`).

Ab MSC.Nastran 2014 können diese Einstellungen direkt im Konfigurations File während der Installation gesetzt werden.

Nachdem sich bis jetzt auf die optimale Ausnutzung der Hardware beschränkt wurde, kommen wir nun zu einigen MSC Nastran Lösungsoptionen, die gezielt je nach Modelbeschaffenheit eingesetzt werden können. Es gibt in der Statik den sparse und den iterativen Solver. Der sparse Solver ist im Allgemeinen schneller bei Schalen- und Balkenmodellen während der iterative Solver bei 3D-Elementen vorzuziehen ist.

Für gemischte Modelle gibt es leider keine generelle Empfehlung. Solange aber mehr als 80% der Elemente 3D-Elemente sind, kann der iterative Solver herangenommen werden.

Der sparse Solver ist default, der iterative Solver muss im Case Control mit `SMETHOD=ELEMENT` angezogen werden. Dieser Solver ist auch in der nichtlinearen Statik SOL 400 unterstützt, allerdings nicht in der SOL 106. Beide Solver (iterativer Solver ab MSC Nastran 2014.2) können shared Memory parallel betrieben werden (keyword `smp=Anzahl der CPUs`). Bis 6 CPUs zeigt sich eine vernünftige Performancesteigerung. In den kommenden MSC Nastran Version wird ein neuer Solver mit einer besseren Skalierung verfügbar sein.

Ein SOL 101 Beispiel:

MODEL SUMMARY			
NUMBER OF GRID	POINTS	=	213943
NUMBER OF CGAP	ELEMENTS	=	565
NUMBER OF CHEXA	ELEMENTS	=	192667
NUMBER OF CPENTA	ELEMENTS	=	4718
NUMBER OF CROD	ELEMENTS	=	3
NUMBER OF CTETRA	ELEMENTS	=	407
NUMBER OF RBE2	ELEMENTS	=	8

