

Product Design Process

MSC SimManager, Oracle Database & SPARC T5-4

SPARC T5-4 Consolidates Database, SimManager and WebServer

Reported: 03/26/2014

Significance of Results

Using Oracle's SPARC T5-4 server to run MSC's SimManager software system and an Oracle Database 12c instance provides a means for reliable and efficient management of simulations and associated data. An automotive design workload is used to demonstrate how the SPARC T5-4 server can be used to consolidate the work of multiple servers and deliver better overall performance.

- The SPARC T5-4 server outperformed the multiple x86 server solution in all 12 workloads considered. The performance was based on the individual total elapsed times to complete each of the simulations, all 3 phases. All 12 workloads were run simultaneously. Overall, the SPARC T5-4 server showed a 1.4 times advantage over the multiple x86 server solution.
- The SPARC T5-4 server solution produced up to a 1.7x advantage over the multiple x86 server solution. The SPARC T5-4 server was only 15% utilized for this workload, allowing for future workload growth.
- The SPARC T5-4 server solution allows for nearly 20 similar product design scenarios to be analyzed in a 24 hour period.

The workload is comprised of twelve simulations. Each simulation consists of an assembly of a model, the analysis phase and post-processing the results. This study highlights the advantages of using the SPARC T5-4 server for the assembly and post-processing phases rather than the multiple x86 server solution. Both solutions used the same compute cluster for the analysis phase.

Performance Landscape

Automotive Product Design Simulation Workload			
Case Study Analysis Type	Elapsed Time (sec)		SPARC Advantage
	SPARC T5-4	Multiple x86	
Car Interior Noise Acoustics via Fluid/Solid, Sol: 448 roots / 96 solves, Fl: 34 Roots / 22 solves	1152	1723	1.5x
Gear Strength Linear Contact	3101	4875	1.6x
Mini Van Modal Frequency Response w/ in-core factorization	1021	1763	1.7x
Car Body Vibrational Normal Modes, 1073 roots / 358 solves Sparse Lanczos	3385	4133	1.2x
Car Body 2 Design Optimization Cycles, 180 roots	1670	2366	1.4x
Car Body 3 Design Optimization Cycles, 250 roots	4347	5549	1.3x
Engine Gasket Nonlinear contact	816	1059	1.3x
Wheel Assembly Nonlinearities: material, deformation, contact	1360	1734	1.3x

Car - Frontal Barrier NCAC "Neon" Nonlinear Explicit Crash Analysis	1729	2357	1.4x
Vehicle Collision Nonlinear Explicit Crash Analysis - Large NCAC Model	4276	6479	1.5x
Car - Frontal Barrier via MSC Nastran "700" (LS-DYNA) Solver - Nonlinear Explicit Crash Analysis	2499	3244	1.3x
Car Body - Large Dynamic Analysis via Automated Component Mode Synthesis (ACMS)	3534	3788	1.1x
Ratio Geometric Means (12 Simulations)			1.38x
<p>Legend</p> <p>Case Study – corresponds to an MSC SimManager Simulation Definition</p> <p>Analysis Type – pertains to the solution phase of the Case Study</p> <p>Elapsed Time – the total elapsed run times (in seconds) for execution of all three phases (assembly/solution/post-processing) of each Simulation Definition</p> <p>SPARC Advantage – x86 time divided by SPARC T5-4 time, greater than 1 means SPARC is faster.</p>			

Configuration Summary

SPARC System Under Test:

SPARC T5-4 server
 4 x 3.6 GHz SPARC T5 processors
 1 TB memory
 8 x 300 GB 10K RPM SAS2 HDDs
 Oracle Solaris 11.1
 Oracle Database 12c Release 1
 MSC SimManager V2012.2.1 for Solaris/SPARC

Multiple x86 Systems Under Test:

Sun Fire X4270 M2 server
 2 x 3.47 GHz Intel Xeon X5690, Hyper-Threading disabled
 48 GB memory (12 x 4 GB)
 3 x 300 GB 10K RPM SAS2
 Oracle Linux 6.1
 MSC SimManager V2012.0.1

Sun Fire X4270 M2 server
 2 x 3.47 GHz Intel Xeon X5690, Hyper-Threading disabled
 48 GB memory (12 x 4 GB)
 4 x 300GB 10K RPM SAS2 HDDs (3 used in ZFS raidz1 pool for database)
 Oracle Solaris 11 Express (snv_151a X86_sru09)
 Oracle Database 11g Release 2

Compute Servers:

6 x Sun Fire X4170 M2 servers each with
 2 x 3.07 GHz Intel Xeon X5675, Hyper-Threading disabled
 48 GB memory (12 x 4 GB)
 3 x 300 GB 10K RPM SAS2 HDDs
 Oracle Linux 6.1
 MD Nastran V2012.1.0

Benchmark Description

This is a performance evaluation of a system solution consisting of hardware and software that forms the infrastructure

for computer-based product design. The system included a database server, a web application server, and a cluster of compute nodes.

MSC SimManager, a simulation data and process management system, orchestrated the execution of a set of analyses used to simulate response to various service conditions anticipated during the lifetime of the proposed product. It runs on the web application server where analysis models are assembled, simulations are deployed, key results are extracted, and reports are generated.

The database server retrieves product data and ultimately archives key results reports and selected (analysis) files. This structured management of information facilitates the determination of optimal product design(s).

Multiple simulation case studies form a single product design scenario. These are run concurrently on a compute node cluster. The simulations are performed by various analysis modules each capable of representing some significant facet of conditions under which the product will be subjected. The benchmark measures the total elapsed time required to process a throughput workload representative of a product design scenario found in a manufacturing industry.

In this particular study a simulation workload was developed that is typical of a product design scenario found in the automotive industry.

Automotive Product Design Workload Distribution								
Simulation Definition	Case Study Analysis Type	Model Type DOF	Cores Used Per Compute Node					
			#1	#2	#3	#4	#5	#6
1	Car Body Acoustics(1)	(L) 654,560	0	0	0	1	0	0
2	Gear Linear Contact	(L) 2,450,320	1	0	0	0	0	0
3	MiniVan Modal Response	(L) 468,223	0	0	0	0	1	0
4	Car Body Normal Modes(2)	(L) 1,315,562	0	0	0	0	1	0
5	Car Body Design Optimize(3)	(L) 486,573	1	0	0	0	0	0
6	Car Body Design Optimize(4)	(L) 486,573	0	0	0	1	0	0
7	Engine Gasket Nonlin Contact(5)	(N) 720,059	2	2	2	2	0	0
8	Wheel Assembly Nonlin Contact(6)	(N) 729,264	2	2	2	2	0	0
9	Car - Rigid Barrier Crash Analysis(7)	(C) 874,000	2	2	2	2	2	2
10	3 Car Collision Crash Analysis(8)	(C) 2,483,000	0	2	4	4	4	2
11	Car - Rigid Barrier Crash Analysis(9)	(C) 851,736	4	4	0	0	4	6
12	Car Body Modal Synthesis(10)	(L) 3,799,278	0	0	2	0	0	2
Total Cores Loaded Per Compute Node (12 maximum)			12	12	12	12	12	12

All simulation model files are taken from established benchmark test cases previously used extensively by industry engineers analysts and designers

Legend

(L) large scale models - modal/frequency response + optimization

(N) complex nonlinear component analysis
(C) crash analysis

- (1) Acoustics via Fluid/Solid, Sol: 448 roots / 96 solves, FI: 34 Roots / 22 solves
- (2) Normal Modes, 1073 roots / 356 solves Sparse Lanczos
- (3) 2 Design Optimization cycles, 180 roots
- (4) 3 Design Optimization cycles, 250 roots
- (5) mildly nonlinear static analysis simulates bolting a cylinder head onto an engine block. The nonlinearity is from changes in the contact conditions and yielding of the gasket material 20 incrs. 27 iters
- (6) strongly nonlinear static analysis determines the footprint of an automobile tire. 3-step analysis: wheel mount, tire inflation, road conditions nonlinearity from large displacements, sliding contact, and hyperelastic material behavior 41 incrs 177 iters
- (7) widely used frontal crash Neon car model created by National Crash Analysis Center (NCAC) ~ 535,000 els 291206 nodes (see topcrunch.org)
- (8) larger 3 Vehicle NCAC model: van rear-ends a compact car that crashes into a midsize car. - 795,000 els 828,000 nodes (see topcrunch.org)
- (9) MSC Neon equiv model demo'ing SimMgr implementation + MSC Dytran simulation 271,000 els 283,912 nodes
- (10) large ACMS (Automated Component Mode Synthesis) module - enables engineers to perform very large-scale dynamic analyses Simulation Manager Automotive Workload Distribution

Key Points and Best Practices

The product design benchmark testbed consists of a database server, a simulation management server and a cluster of compute nodes. For each test set, the cluster of compute nodes remains constant. For the SPARC T5-4 server test, the database server and the simulation management server are both on the SPARC T5-4 server. For the multiple x86 servers test, the database server and the simulation management server are on separate Sun Fire X4270 M2 servers.

Shared Temp Directory (interface to compute nodes)

When running on remote nodes, particularly multiple compute nodes, it is best to establish a SimManager shared temp directory that is available to both SimManager and the compute nodes. The location of this directory is specified by defining the SMTempDirectory parameter in the file, Portal.properties as follows:

```
#####
# This property defines the "TEMP" shared file scratch area
# intended to be seen by all compute nodes as well as from
# the SimMgr server where the job will be initiated
#####
SMTempDirectory = /ctmp0/oledog/tmp
#####
```

The location of Portal.properties file depends on how the Portal for SimManager was configured on the web application server.

```
cd /msc/apache-tomcat-6.0.32/webapps/SimManager/init/SimManager/portalInstances/EnterpriseEdition
%ls -l
Portal.properties
VaultConfig.properties
```

When SimManager spawns a job it starts by creating a job dependent "Simulation" subdirectory.

(e.g. /...../tmp/SimMan/Prod08_V1_scen08_Job801/Simulation/

Under that are typically found subdirectories:

- Input typically contains the application input files copied or derived from files archived in the data base

- Output where all final output files are stored following execution of the application associated with the job

- PPO if post-processing is specified as part of the entire job the "output" files are stored here

When the job is complete, files are archived in the database per instructions specified in the application context, as well as, by the job Simulation Definition.

Different ISV Applications and Application Contexts

SimManager is set up to run (the multi-modular) MD Nastran and a variety of other MSC applications including MSC pre- (e.g. PATRAN) and post-processors (e.g. MSC Fatigue). However, it is possible to also run other applications. One way of doing this is to create a new application context. If this is done one should first create a corresponding application and finally also create a corresponding set of application parameters.

The format of the application context is as follows:

```
/msc/runScript.sh /msc/runSol4cpJob.js -DinputDeck=${inputDeck}
-DoutputDirectory=${outputDirectory} -Dmemory=${memory}
-DmakeFile=${makeFile} -Dcpu=${numberOfCpus} >
${outputDirectory}/solver.stdout 2> ${outputDirectory}/solver.stderr
```

The application context uses a Java run script (e.g here "runSol4cpJob.js") where, for example, when running Nastran the application command line entry might be as follows:

```
//NASTRAN
// commandLine = "/ctmp0/oledog/mscv2012/MS_C_Nastran/20121/bin/nast20121 "
+ inputdeck +"scr=yes bat=no app=no sdir=/ptmp/tmp";
```

Additional command line parameters can be added to the Java run scripts for these various application contexts. For example, memory allocations for a Nastran job might be specified as follows:

```
//MEMORY
// if (imemory != null)
// commandLine = commandLine + " mem="+imemory;
// else
// commandLine = commandLine + " mem=1000mb smem=400mb";
```

Multiple Jobs via SimGenerator

The capability to use a variety of different applications can be extended further to run these different applications all concurrently on a cluster of compute nodes. This is readily done via the SimGenerator module within SimManager. This module is normally set up to run a series of different simulations, using the same application with various components or assemblies of the same product model. In this arrangement the Simulation Definition for each job includes Assemble and Simulate steps as well as a Post-Processing phase.

However SimGenerator can be set up to run a variety of entirely different applications each one doing a significantly different type of analysis. This is done by establishing a "scenario" that includes a set of different Simulation Definitions where each one would reference a different solver application with its own associated application context and Java run script and with its own appropriate post-processing application that can also be defined. In this instance conventional Simulation Definition assembly procedures are also bypassed by modifying the Java scripts associated with the particular Assembly "Setup" specified on the "Application" tab for the particular Simulation Definition. The conventional (component) model files are then replaced by complete (already "assembled") input files that are imported and stored as (mesh) models files associated with "Model Definitions" and Items and "Item Definitions" all of which comprise the Product that is currently the focus of the current "Simulation Study".

See Also

- MSC Software – SimManager
- MSC Nastran

- SPARC T5-4 Server
oracle.com OTN
- Oracle Solaris
oracle.com OTN
- Oracle Database
oracle.com OTN

Disclosure Statement

Copyright 2014, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Results as of March 26, 2014.

[Back To Top](#)

Copyright 1994-2014, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.