WHITE PAPER

SGI® Technology Guide for CD-adapco™ STAR-CCM+® Analysts

March, 2014

Author

Dr. Ayad Jassim†

Abstract

STAR-CCM+ is a process oriented CAE software used to solve multi-disciplinary problems within a single integrated environment. STAR-CCM+ is designed to utilize hardware resources as effectively independent of physical location and local computer resources. In general CFD software has experienced significant effects to their compute environment in recent history. These effects were governed by hardware and software features such as the introduction of "cores" and "sockets" as processor components along with memory speed, I/O sub-systems, interconnect fabrics and communications software such as InfiniBand and MPI.

This SGI Technology guide provides an analysis of the parallel performance of two STAR-CCM+ solvers, namely the Segregated and the Coupled solvers using the Intel x86-64 architectural features of Intel®, Hyper-Threading technology and Intel® Turbo Boost technology on SGI computer systems running the Intel® Xeon® Processor E5-2600/E5-4600 product families (code named Sandy Bridge) and the Intel® Xeon® processor E5-2600 v2 product families (code named Ivy Bridge). This work is based on SGI hardware architectures, specifically, the SGI® ICE™ X System, SGI® Rackable® Standard Depth C2112-4RP4 cluster solutions as well as the SGI® UV™ 2000 shared memory system. The main objective is to provide STAR-CCM+ users with a qualitative understanding of the benefits gained from these two features when executing them on SGI hardware platforms.

† SGI Applications Engineering

TABLE OF CONTENTS

1.0 About SGI Systems

SGI systems were used to perform the benchmarks outlined in this paper, including the SGI® Rackable® standard-depth cluster; SGI® ICE™ X integrated blade cluster and the SGI UV 2000 shared memory system. For a comprehensive list of references for the hardware used in this paper see [1] through [5].

1.1 SGI® Rackable® Standard-Depth Cluster

SGI Rackable standard-depth, rackmount C2112-4RP4 servers support up to 512GB of memory per node in a dense architecture with up to 96 cores per 2U with support for up to 56 Gb/s. FDR and QDR InfiniBand, twelve-core Intel® Xeon® processor E5-2600 v2 series and DDR3 memory running SUSE® Linux® Enterprise Server or Red Hat® Enterprise Linux Server for a reduced TCO. (Fig. 1)

SGI Rackable Configurations used in this paper:

Benchmark System 1

- Product name: SGI Rackable C2112-4RP4
- Intel® Xeon® 8-core 2.6 GHz E5-2670
- IB QDR interconnect
- 4 GB of Memory/core Memory Speed 1600MHz
- Altair® PBS Professional Batch Scheduler v11
- SLES or RHEL with latest SGI Performance Suite™

Benchmark System 2

- Product name: SGI Rackable C2112-4RP4
- Intel® Xeon® 12-core 2.7 GHz E5-2697 v2
- IB FDR interconnect
- 4 GB of Memory/core Memory Speed 1867MHz
- Altair® PBS Professional Batch Scheduler v11
- SLES or RHEL with latest SGI Performance Suite

Figure 1: *Overhead View of SGI Rackable Server with the Top Cover Removed*

1.2 SGI® ICE™ X System

The SGI® ICE™ X is the world's fastest distributed memory commercial supercomputer for over four years running. This performance leadership is proven in the lab, and at customer sites including the largest and fastest pure compute InfiniBand cluster in the world. The system can be configured with compute nodes comprising Intel® Xeon® processor E5-2600 v2 series exclusively or with compute nodes comprising both Intel® Xeon® processors and Intel® Xeon Phi™ coprocessors or Nvidia® compute GPU's running on SUSE Linux Enterprise Server and Red Hat Enterprise Linux. SGI ICE X can be architected in a variety of topologies with choice of switch and single or dual plane FDR InfiniBand interconnect topology. The integrated bladed design offers rack-level redundant power and cooling via air, warm water or cold water and is also available with storage and visualization options (Fig.2).

Configurations used in this paper:

Benchmark System 1
Product name: SGI ICE X

- Intel® Xeon® 8-core 2.9 GHz E5-2690
- Integrated IB FDR interconnect Hypercube/Fat Tree
- 4 GB of Memory/core memory speed 1600 MHz
- Altair® PBS Professional Batch Scheduler v11
- SLES or RHEL with latest SGI Performance Suite

Benchmark System 2
Product name: SGI ICE X

- Intel® Xeon® 10-core 3.0 GHz E5-2690 v2
- Integrated IB FDR interconnect Hypercube/Fat Tree
- 4 GB of Memory/core memory speed 1867MHz
- Altair® PBS Professional Batch Scheduler v11
- SLES or RHEL with latest SGI Performance Suite

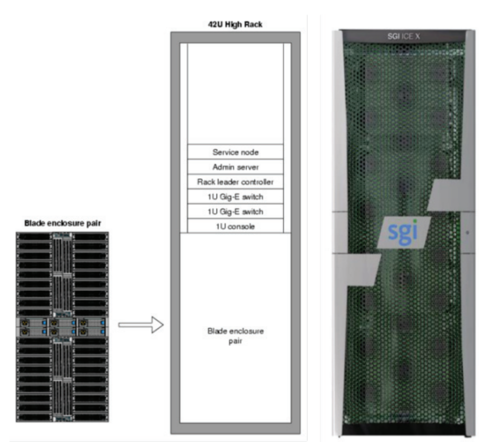

Figure 2: *SGI ICE X Cluster with Blade Enclosure*

1.3 SGI® UV™ 2000

The SGI UV 2000 is a scalable cache-coherent shared memory architecture. SGI UV 2 product family can scale a single system image (SSI) to a maximum of 2,048 cores (4,096 threads) due to its SGI NUMAflex®, blade-based architecture. The SGI UV 2 includes the Intel® Xeon® processor E5-4600 and the latest Xeon® processor E5-4600 v2 product family. This system can operate unmodified versions of Linux such as SUSE Linux Enterprise Server and Red Hat Enterprise Linux. The SGI UV also supports scalable graphics accelerator cards, including NVIDIA® Quadro®, NVIDIA® Tesla® K20 GPU computing accelerator and Intel® Xeon Phi™. Job memory is allocated independently from cores allocation for maximum multi-user, heterogeneous workload environment flexibility. Whereas on a cluster, problems have to be decomposed and require many nodes to be available, the SGI UV can run a large memory problem on any number of cores and application license availability with less concern of the job getting killed for lack of memory resources compared to a cluster. Fig. 3.

Figure 3: *SGI UV 2000 with door open*

Configurations used in this paper:

Benchmark System 1
Product name: SGI UV 2000

- 64 sockets (512 cores) per rack
- Intel® Xeon® 8 core 2.7 GHz E5-4650
- SGI NUMALink® 6 Interconnect
- 4 GB of Memory/core
- Altair PBS Professional Batch Scheduler with CPUSET MOM v11
- SLES or RHEL with latest SGI Performance Suite with Accelerate

Benchmark System 2
Product name: SGI UV 2000

- 32 sockets (320 cores)
- Intel® Xeon® 10 core 2.4 GHz E5-4650 v2
- SGI NUMALink 6 Interconnect
- 4 GB of Memory/core
- Altair® PBS Professional Batch Scheduler with CPUSET MOM v11
- SLES or RHEL with latest SGI Performance Suite with Accelerate

1.4 STAR-CCM+ MPI Selection

STAR-CCM+ parallel solvers implement a number of MPI (Message Passing Interface) libraries. The most relevant to SGI hardware are MPICH, IBM Platform MPI, OPEN MPI and SGI MPI. Users can select a particular MPI by passing the mpi name argument to the –mpidriver flag of the starccm+ command, e.g.

starccm+ -mpidriver <name> … other options …

where <name> can be any of mpich, platform, openmpi or sgi.

Note that the SGI MPI is highly recommended for use of SGI UV systems due to its ability to work within cpu sets and flexibility of pinning mpi threads to cores. To instruct the STAR-CCM+ solver to use the SGI MPI, users need to set the STAR-CCM+ environment variable SGI_MPI_HOME to point to the path of the 'SGI Message Passsing Tool Kit' installation (accessible to all compute nodes) in addition to setting the option '-mpidriver sgi' in the starccm+ command line. Two more useful SGI MPI environment variables are MPI_DSM_CPULIST="<selected-cores>:allhosts" (to pin mpi threads to selected cores) and MPI_SHARED_NEIGHBORHOOD=HOST (in the case of simulations with relatively large number of domains/threads). See 'man mpi' in the 'SGI Message Passing Tool Kit.'

1.5 Resource & Workload Scheduling

Resource and workload scheduling allows you to manage large, complex applications, dynamic and unpredictable workloads, and optimize limited computing resources. SGI offers several solutions that our customers can choose from to best meet their needs.

Altair Engineering PBS Professional®

Altair PBS Professional® is SGI's preferred workload management tool for technical computing scaling across SGI's clusters and servers. PBS Professional is sold by SGI and supported by Altair Engineering and SGI.

Features:

- Policy-driven workload management which improves productivity, meets service levels, and minimizes hardware and software costs
- Integrated operation with SGI Management Center for features such as workload driven, automated dynamic provisioning

Adaptive Computing Moab HPC Suite Basic Edition

Adaptive Computing Moab® HPC Suite enables intelligent predictive scheduling for workloads on scalable systems.

Features:

- Policy-based HPC workload manager that integrates scheduling, managing, monitoring and reporting of cluster workloads
- Includes TORQUE resource manager

2.0 STAR-CCM+ Overview

STAR-CCM+ includes an extensive range of validated physical models that provide the user with a toolset capable of tackling the most complex multi-disciplinary engineering problems. The software is deployed as a client that handles the user interface and visualization, and a server which performs the compute operations. The client/server approach is designed to facilitate easy collaboration across organizations, simulations can be accessed independently of physical location and local computer resources. STAR-CCM+ recently became the first commercial Computer Fluid Dynamics (CFD) package to mesh and solve a problem with over one billion cells. Much more than just a CFD solver, STAR-CCM+ is an entire engineering process for solving problems involving flow (of fluids or solids), heat transfer, and stress. It provides a suite of integrated components that combine to produce a powerful package that can address a wide variety of modeling needs. These components are:

3D-CAD Modeler

The STAR-CCM+ 3D-CAD modeler is a feature-based parametric solid modeler within STAR-CCM+ that allows geometry to be built from scratch. The 3D-CAD models, can subsequently be converted to geometry parts for meshing and solving. A major feature of 3D-CAD is design parameters, which allow you to modify the models from outside of the 3D-CAD environment. These allow you to solve for a particular geometry, change the size of one or more components, and quickly rerun the case.

CAD Embedding

STAR-CCM+ simulations can be set up, run and post-processed from within popular CAD and PLM environments such as SolidWorks, CATIA V5, Pro/ENGINEER, SpaceClaim, and NX. STAR-CCM+'s unique approach gets you from CAD model to an accurate CFD solution quickly and more reliably. CFD results are linked directly to the CAD geometry (a process called associativity). The STAR-CCM+ CAD clients have bi-directional associativity so that geometry transferred across may be modified directly in STAR-CCM+ with the underlying CAD model updated.

Surface Preparation Tools

At the heart of STAR-CCM+ is an automated process that links a powerful surface wrapper to CD-adapco unique meshing technology. The surface wrapper significantly reduces the number of hours spent on surface clean-up and, for problems that involve large assemblies of complex geometry parts, reduces the entire meshing process to hours instead of days.

The surface wrapper works by 'shrink-wrapping' a high-quality triangulated surface mesh onto any geometrical model, closing holes in the geometry and joining disconnected and overlapping surfaces, providing a single manifold surface that can be used to automatically generate a computational mesh without user intervention.

STAR-CCM+ also includes a comprehensive set of surface-repair tools that allow users to interactively enhance the quality of imported or wrapped surfaces, offering the choice of a completely automatic repair, user control, or a combination of both.

Automatic Meshing Technology

Advanced automatic meshing technology generates either polyhedral or predominantly hexahedral control volumes at the touch of a button, offering a combination of speed, control, and accuracy. For problems involving multiple frames of reference, fluid-structure interaction and conjugate heat transfer, STAR-CCM+ can automatically create conformal meshes across multiple physical domains.

An important part of mesh generation for accurate CFD simulation is the near-wall region, or extrusion-layer mesh. STAR-CCM+ automatically produces a high-quality extrusion layer mesh on all walls in the domain. In addition, you can control the position, size, growth-rate, and number of cell layers in the extrusion-layer mesh.

Physics Models

STAR-CCM+ includes an extensive range of validated physical models that provide the user with a toolset capable of tackling the most complex multi-disciplinary engineering problems.

Time
Steady-state, unsteady implicit/explicit, harmonic balance

Flow
Coupled/segregated flow and energy

Motion
Stationary, moving reference frame, rigid body motion, mesh morphing, large displacement solid stress, overset meshes

Dynamic Fluid Body Interaction (DFBI)
Fluid-induced motion in 6 degrees of freedom or less, catenary and linear spring couplings

Material
Single, multiphase and multi-component fluids, solids

Regime
Inviscid, laminar, turbulent (RANS, LES, DES), laminar-turbulent transition modeling. Incompressible through to hypersonic non-Newtonian flows

Sensitivity analysis
Adjoint solver with cost functions for pressure drop, uniformity, force, moment, tumble and swirl. Sensitivities with respect to position and flow variables.

Multi-Domain
Porous media (volumetric and baffle), fan and heat exchanger models

Heat Transfer and Conjugate Heat Transfer
Conducting solid shells, solar, multi-band and specular thermal radiation (discrete ordinates or surface-to-surface) convection, conjugate heat transfer

Multi-component Multiphase
Free surface (VOF) with boiling, cavitation, evaporation & condensation, melting & solidification, Eulerian multiphase with boiling, gas dissolution, population balance, granular flow, Lagrangian, droplet breakup, collision, evaporation, erosion and wall interaction as well as discrete element modeling (DEM) composite and clumped particles, non-spherical contact, particle deformation and breakup, fluid film with droplet stripping, melting & solidification, evaporation & boiling, dispersed multiphase for soiling and icing analysis fluid film.

Multi-Discipline
Finite Volume stress (small and large displacements, contacts), fluid structure interaction, electromagnetic field, Joule heating, electro-deposition coating, electrochemistry, casting

Combustion and Chemical Reaction
PPDF, CFM, PCFM, EBU, progress variable model (PVM), thickened flame model, soot moments emission, and DARS CFD complex chemistry coupling interphase reactions for Eulerian multiphase

Aeroacoustic Analysis

Fast Fourier transform (FFT) spectral analysis, broadband noise sources, Ffowcs-Williams Hawkings (FWH) sound propagation model, wave number analysis

Post-processing

STAR-CCM+ has a comprehensive suite of post-processing tools designed to enable you to obtain maximum value and understanding from your CFD simulation. This includes scalar and vector scenes, streamlines, scene animation, numerical reporting, data plotting, import, and export of table data, and spectral analysis of acoustical data.

CAE Integration

Several third-party analysis packages can be coupled with STAR-CCM+ to further extend the range of possible simulations you can do. Co-simulation is possible using Abaqus, GT-Power, WAVE, RELAP5-3D, AMESIM and OLGA, and file-based coupling is possible for other tools such as Radtherm®, NASTRAN and ANSYS®.

3.0 SGI Benchmarks of STAR-CCM+

3.1 Job Submittal Procedure

On the SGI hardware, the STAR-CCM+ job submittal procedure can be made in one of two methods:

Direct job submission

Direct job submission may be done using the following two examples:

a- Run a 32-thread SGI MPI job on a Shared memory system:
export MPI_DSM_DISTRIBUTE=1
starccm+ -np 32 –mpidriver sgi model.sim

b- Run a 32-thread SGI MPI job on two nodes in a cluster system:
export SGI_MPI_CPULIST=”0-15:allhosts”
starccm+ –mpidriver sgi node1:16,node2:16 model.sim

Using PBS Pro™ job scheduler job submission

Batch schedulers/resource managers dispatch jobs from a front-end login node to be executed on one or more compute nodes. To achieve the best runtime in a batch environment disk access to input and output files should be placed on the high performance shared parallel file system. The following is the synopsis of a job submission script.

The following is an example of a PBS Pro job script that submits a STAR-CCM+ job by selecting any of SGI MPI, PLATFORM MPI or OPEN MPI: (users will need to edit parts of this script to suit their environment)

```
#!/bin/csh
#
### PBS ###
#PBS -S /bin/csh -j oe -k o -r n
### SGE ###
#####$ -S /bin/csh -j y -cwd
#PBS -l walltime=0:15:00
#
# run this script using the PBSpro command:
#
# qsub -l select=<no.Nodes>:ncpus=<no.cores.per.node>:mpiprocs=<no.mpi.threads.per.node> \
#     -v NPROC=<no.mpi.threads.for.the.job>,CASE=<name> \
#        MPITYPE=OPENMPI|PMPI|SGIMPI,MACHINE=snb2,cam,cy013   ./this_script
#
setenv CCMVER “STAR-CCM+8.02.011”
setenv PATH  /apps /ccm+/${CCMVER}/${CCMVER}/star/bin:$PATH
setenv CDLMD_LICENSE_FILE 1999@appsengr.engr.sgi.com
setenv RAMFILES 1
cd $PBS_O_WORKDIR
if ( $MPITYPE == “OMPI” ) then
    (time starccm+ -power -np ${NPROC} \
     -mpidriver openmpi \
                -mpiflags “--bind-to-core \
                          --slot-list 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19” \
                -machinefile $PBS_NODEFILE \
                -rsh ssh \
                -batch Benchmark25.java \
                ./${CASE}.sim ) \
                 >& ${NPROC}core_${CASE}_${MPITYPE}_${CCMVER}_${MACHINE}.log
else if ( $MPITYPE == “PMPI” ) then
    setenv REMSH ssh
    setenv MPI_REMSH ssh
    # setenv MPI_GLOBAL_MEMSIZE 2957168
    (time starccm+ -power -np ${NPROC} \
     -mpidriver platform:” -cpu_bind=\
      MAP_CPU:{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19} \
     -e MPI_REMSH=ssh -e MPI_MAX_REMSH=2048 -e REMSH=ssh “ \
     -machinefile $PBS_NODEFILE \
     -rsh ssh \
     -batch Benchmark25.java \
     ./${CASE}.sim ) \
```

```
  >& ${NPROC}core_${CASE}_${MPITYPE}_${CCMVER}_${MACHINE}.log
else if ( $MPITYPE == "MPT" ) then
  setenv MPI_DSM_CPULIST "0-19:allhosts"
  setenv MPTVER "2.09-ga"
  #SGI_MPI_HOME is a STAR_CCM+ environment variable that contains the PATH to
  #the SGIMPI installation directory
  setenv SGI_MPI_HOME /sw/sdev/mpt-x86_64/${MPTVER}
  # use the env var below for UV only
  #setenv MPI_SHARED_NEIGHBORHOOD HOST
   setenv MPI_IB_RAILS 2
   setenv MPI_VERBOSE 1
#
  (time starccm+ -power -np ${NPROC} \
   -mpidriver sgi \
   -batchsystem "pbs" \
   -rsh ssh \
   -batch Benchmark25.java \
   ./${CASE}.sim ) \
   >& ${NPROC}core_${CASE}_${MPITYPE}${MPTVER}_${CCMVER}_${MACHINE}.log
else
  echo "Unkown MPI type $MPITYPE "
endif
```

3.2 Definitions and Performance metrics

The Intel® Xeon® E5-2600 processor family comes with two features, namely Intel® Turbo Boost 2.0 and Intel® Hyper-Threading Technology, where Turbo Boost is a feature first introduced in the Intel® Xeon® 5500 series for increasing performance by raising the core operating frequency within controlled limits depending on the sockets' thermal envelope. The mode of activation is a function of how many cores are active at a given moment which may be the case when OpenMP threads or MPI processes are idle under their running parent. For example, for a base frequency of 3.0GHz, with 1-2 cores active, their running frequencies will be throttled up to 3.5GHz, but with 3-4 cores active only to 3.3 GHz. For most computations, utilizing Turbo Boost technology can result in improved runtimes, but the overall benefit may be mitigated by the presence of performance bottlenecks other than pure arithmetic processing.

Intel® Hyper-Threading Technology (Intel® HT Technology) is an Intel patented technique for simultaneous multithreading (SMT). In this technique some execution elements are duplicated, specifically elements that store the executional state, where as elements that actually do the execution are not duplicated. This means that only one processor is physically present but the operating system sees two virtual processors, and shares the workload between them.

Note those units such as L1 and L2 cache, and the execution engine itself are shared between the two competing threads. Hyper-threading requires both operating system and CPU support. Intel's Xeon® 5600 series processor reintroduced a form of Intel® HT Technology where 6 physical cores effectively scale to 12 virtual cores thus executing 12 threads. Thus, for example, in the Intel® Xeon® 10-core 3.0 GHz E5-2690 v2 based compute node there is a total of 40 virtual cores allowing one to execute 40 threads. In practice, an executing thread may occasionally be idle waiting for data from main memory or the completion of an I/O or system operation. A processor may stall due to a cache miss, branch misprediction, data dependency or the completion of an I/O operation. This allows another thread to execute concurrently on the same core taking advantage of such idle periods.

Thus in this guide we use the following definitions and metrics based on the two features above:

'S_n' denote the elapsed time for the n-thread per node job in a standard non HTT and non Turbo Boost mode of operation. Each node is configured to have 'n' physical cores.

'H_{2n}' denote the elapsed time for a 2n-thread per node job under HTT mode of operation where each node is configured to have 'n' physical and 'n' hyper-threaded cores.

'T_n' denote the elapsed time for the n-thread per node job in a Turbo Boost mode and non HTT and non mode of operation. Each node is configured to have n physical cores.

'C_{2n}' denote the elapsed time for a 2n-thread per node job running in combined Turbo Boost and HTT mode of operation. Each node is configured to have 'n' physical and 'n' hyper-threaded cores.

'$\%H_{2n}$' denote the percentage gain of 'H_{2n}' relative to 'S_n'

'$\%T_n$' denote the percentage gain of 'T_n' relative to 'S_n'

'$\%C_{2n}$' denote the percentage gain of 'C_{2n}' relative to 'S_n'

For more details on the above definitions and metrics see [6].

4.0 Benchmark Examples

LeMansCar17m: 17m cells, turbulent flow, 500 iterations using Segregated and Coupled solvers, Fig 4.

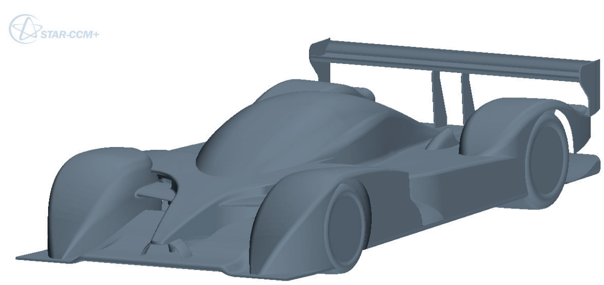

Figure 4: *LeMansCar17 Geometry*

Large Classified Model: Very large model, turbulent flow, 11 iterations using Segregated and Coupled Solvers

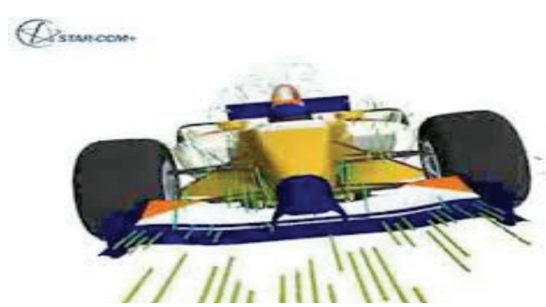

Figure 5: *Large Classified Model*

4.1 LeMansCar17m Model Benchmark Results

Figures 6, 7 and 8 present benchmark results for the LeMansCar17m model on SGI Sandy Bridge based hardware, namely SGI Rackable C2112-4TY14, SGI UV 2000 and SGI ICE X respectively.

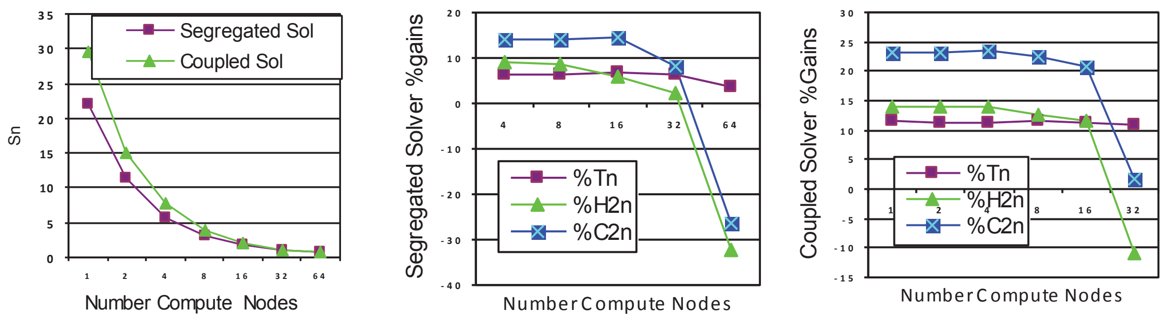

Figure 6: *LeMansCar17m Segregated and Coupled solver average elapsed times per iteration* S_n *and the corresponding percentage gains* $\%H_{2n}$, $\%T_n$ *and* $\%C_{2n}$ *on SGI Rackable C2112-4TY14 with Sandy Bridge E5-2670 @ 2.60GHz, n=16.*

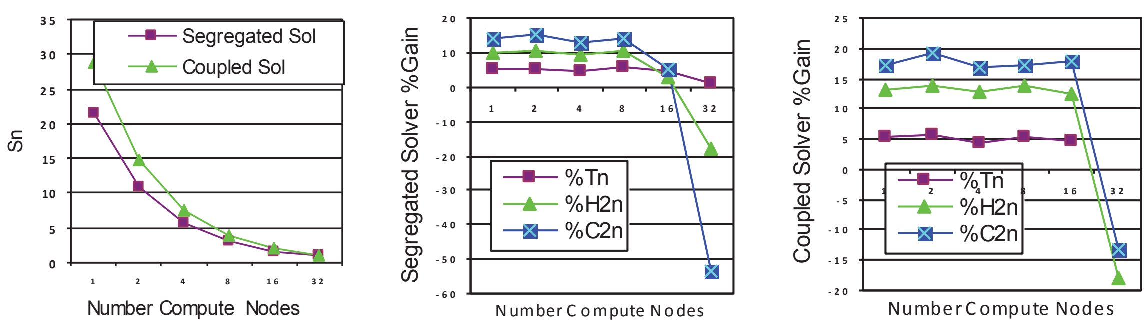

Figure 7: *LeMansCar17m Segregated and Coupled solver average elapsed times per iteration* S_n *and the corresponding percentage gains* $\%H_{2n}$, $\%T_n$ *and* $\%C_{2n}$ *on UV 2000 with Sandy Bridge E5-4600 @ 2.60GHz, n=16.*

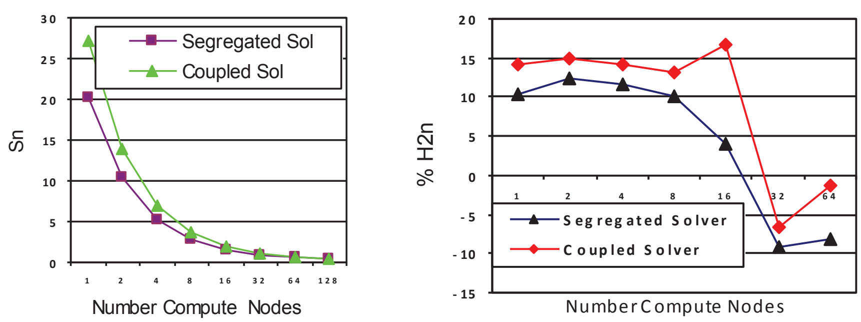

Figure 8: *LeMansCar17m Segregated and Coupled solver average elapsed times per iteration* S_n *and the corresponding percentage gains* $\%H_{2n}$ *on SGI ICE X with Sandy Bridge E5-2690 @ 2.90GHz, n=16.*

Figures 9, 10 and 11 present benchmark results for the LeMansCar17m model on SGI Ivy Bridge based hardware, namely SGI Rackable C2112-4TY14, SGI UV 2000 and SGI ICE X respectively.

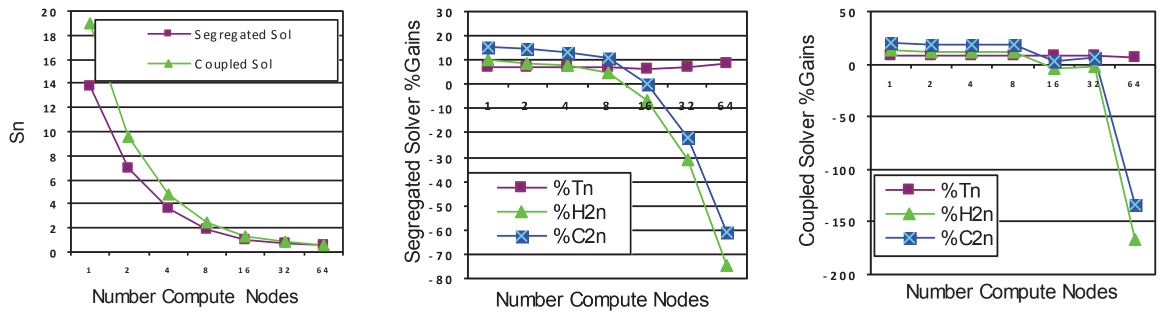

Figure 9: *LeMansCar17m Segregated and Coupled solver average elapsed times per iteration S_n and the corresponding percentage gains $\%H_{2n}$, $\%T_n$ and $\%C_{2n}$ on SGI Rackable C2112-4TY14 with Ivy Bridge E5-2697v2 @ 2.70GHz FDR cluster, n=24.*

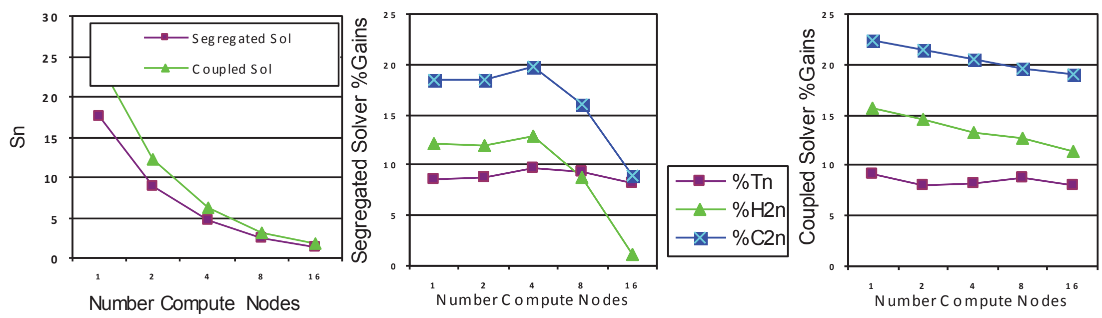

Figure 10: *LeMansCar17m Segregated and Coupled solver average elapsed times per iteration S_n and the corresponding percentage gains $\%H_{2n}$, $\%T_n$ and $\%C_{2n}$ on UV 2000 with Ivy Bridge E5-4650 v2 @ 2.40GHz, n=20.*

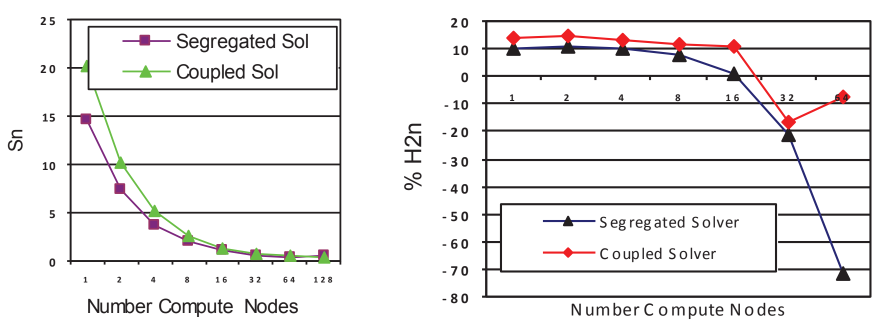

Figure 11: *LeMansCar17m Segregated and Coupled solver average elapsed times per iteration S_n and the corresponding percentage gains $\%H_{2n}$ on SGI ICE X with Ivy Bridge E5-2690v2 @ 3.0GHz cluster, n=20.*

4.2 Large Classified Model Benchmark Results

Figures 12, 13 and 14 present benchmark results for the Large Classified model on SGI Sandy Bridge based hardware, namely SGI Rackable C2112-4TY14, SGI UV 2000 and SGI ICE X respectively.

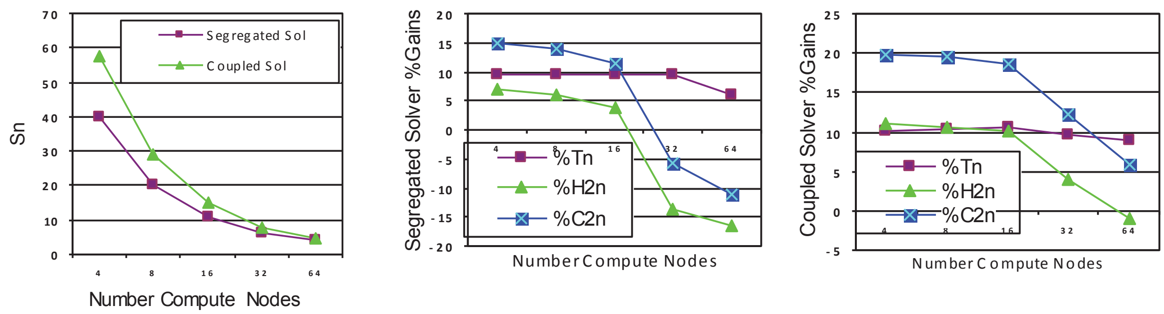

Figure 12: *Large Classified Segregated and Coupled solver average elapsed times per iteration S_n and the corresponding percentage gains $\%H_{2n}$, $\%T_n$ and $\%C_{2n}$ on SGI Rackable C2112-4TY14 with Sandy Bridge E5-2670 @ 2.60GHz, n=16.*

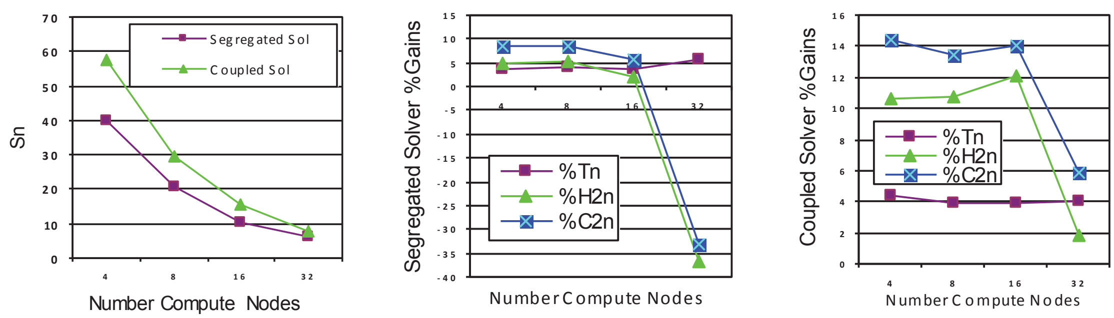

Figure 13: *Large Classified Segregated and Coupled solver average elapsed times per iteration S_n and the corresponding percentage gains $\%H_{2n}$, $\%T_n$ and $\%C_{2n}$ on SGI UV 2000 with Sandy Bridge E5-4600 @ 2.60GHz, n=16.*

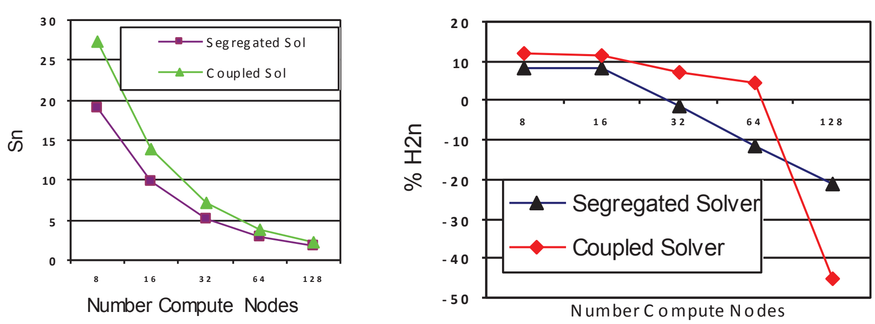

Figure 14: *Large Classified Segregated and Coupled solver average elapsed times per iteration S_n and the corresponding percentage gains $\%H_{2n}$ on SGI ICE X with Sandy Bridge E5-2690 @ 2.90GHz cluster, n=16.*

Figures 15,16 and 17 present benchmark results for the Large Classified model on SGI Ivy Bridge based hardware, namely SGI Rackable C2112-4TY14, SGI UV 2000 and SGI ICE X respectively.

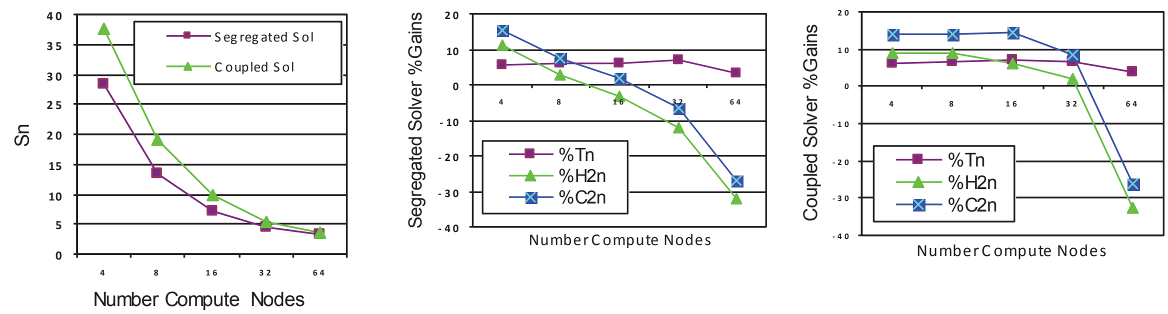

Figure 15: *Large Classified Segregated and Coupled solver average elapsed times per iteration* S_n *and the corresponding percentage gains* $\%H_{2n}$*,* $\%T_n$ *and* $\%C_{2n}$ *on SGI Rackable C2112-4TY14 with Ivy Bridge E5-2697 v2 @ 2.7GHz FDR, n=24.*

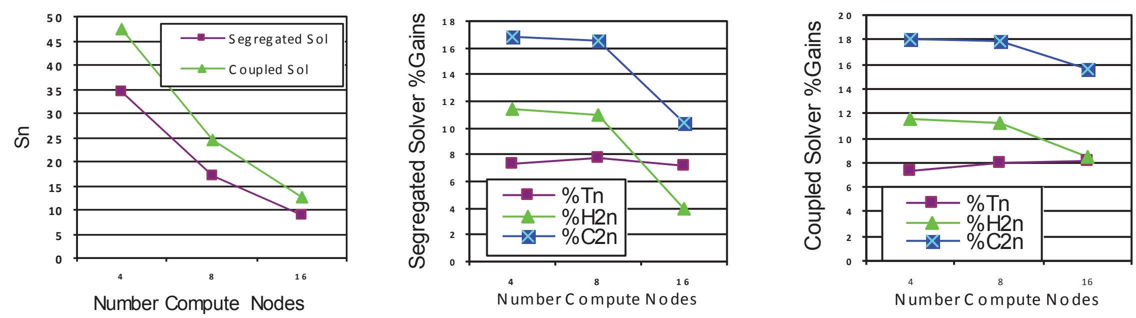

Figure 16: *Large Classified Segregated and Coupled solver average elapsed times per iteration* S_n *and the corresponding percentage gains* $\%H_{2n}$*,* $\%T_n$ *and* $\%C_{2n}$ *on SGI UV 2000 with Ivy Bridge E5-4600 v2 @ 2.4GHz, n=20.*

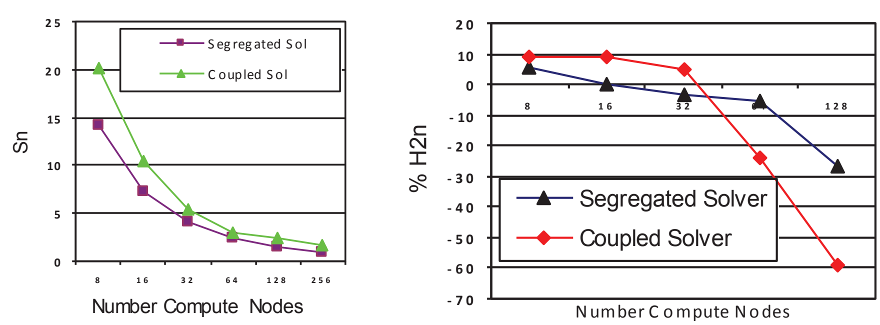

Figure 17: *Large Classified Segregated and Coupled solver average elapsed times per iteration* S_n *and the corresponding percentage gains* $\%H_{2n}$ *on SGI ICE X with Ivy Bridge E5-2690v2 @ 2.90GHz cluster, n=20.*

4.3 Comparisons

Figures 18 (a, b, c and d) indicate that the H_{2n} values in the case of ICE X Ivy Bridge, n=20, drop to negative values faster than in the case of ICE X Sandy Bridge, n=16 (for both Segregated and Coupled Solvers) due to the larger number of cores per socket. Similar observations apply also to the SGI Rackable and UV 2000.

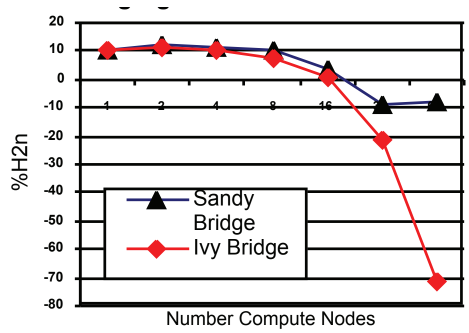

Figure 18a: LeMansCar %H_{2n} on ICE X Segregated Solver.

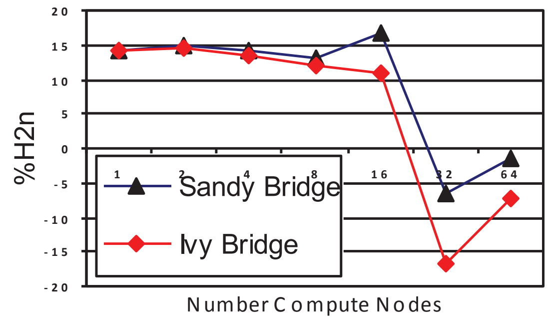

Figure 18b: LeMansCar %H_{2n} on ICE X Coupled Solver.

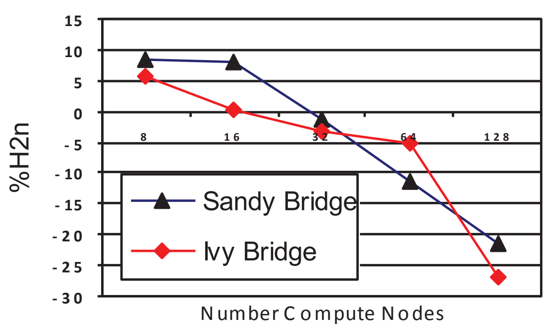

Figure 18c: Large Classified %H_{2n} on ICE X Segregated Solver.

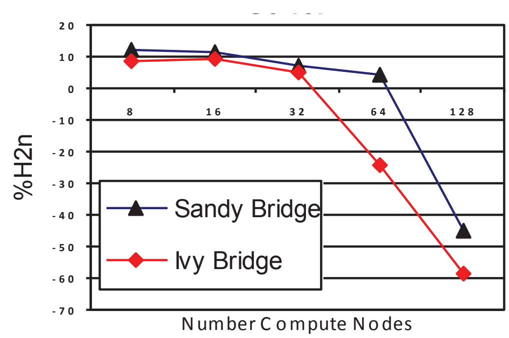

Figure 18d: Large Classified %H_{2n} on ICE X Coupled Solver.

Figures 19 (a, b, c and d) present plots of Hyper-threading gain, %H_{2n}, values for four different size models generated on an SGI ICE-X Sandy Bridge (n=16) and Ivy Bridge (n=20) using the Segregated and Coupled solvers. The four models are the Mercedes A Class 5M cell model, the LeMansCar 17M cell model, the LeMansCar94M cell model refined from the 17M model and the Large Classified model. These figures describe the trends of Hyper-threading gain based on the size of the model. Comparing the trends for the four cases it seems that Hyper-threading gains are lowest for the A Class model which is the smallest size model (approximately 5M cells). However for the other three models, namely the LeMansCar17M, LeMansCar94M and the Large Classified, the corresponding gains appear to be of similar trends despite the significant differences in the sizes of the three models. This indicates that Hyper-threading gains are limited by the compute node's resources such as memory bandwidth and cache sizes.

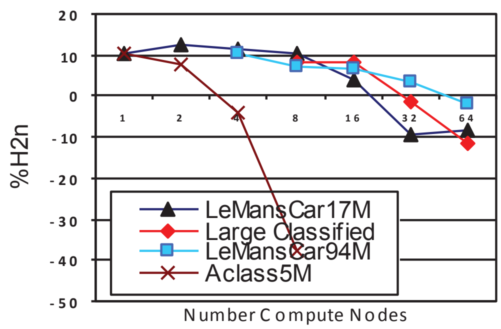

Figure 19a: *Plot of Segregated %H_{2n} gains on ICE X Sandy Bridge for four different size models.*

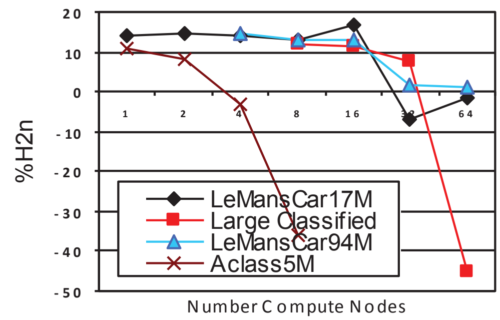

Figure 19b: *Plot of Coupled Solver %H_{2n} gains on ICE X Sandy Bridge for four different size models.*

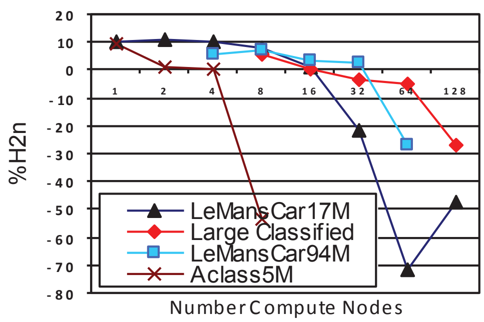

Figure 19c: *Plot of Segregated %H_{2n} gains on ICE X Ivy Bridge for four different size models.*

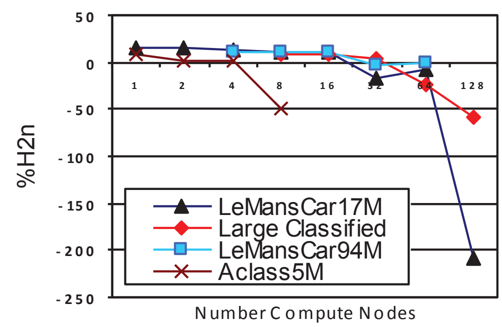

Figure 19d: *Plot of Coupled Solver %H_{2n} gains on ICE X Ivy Bridge for four different size models.*

In fact the Large Classified model gains appear to be slightly below those of the two LeMansCar models. This observation indicates that extremely large size models can incur heavier demands on the Hyper-threading resources of the compute node. Thus Hyper-threading gain values have a limit beyond which there are no further gains irrespective of how large the model size may be.

A further comparison may be made on the effects of FDR–to-QDR interconnects on the performance of the Segregated and Coupled solvers. Figures 20a and 20b present plots of percentage gains of S_n FDR to S_n QDR on the SGI Rackable cluster with Intel E5-2697 v2, n=24, for both solvers in the case of the LeMansCar17m and the Large Classified models respectively.

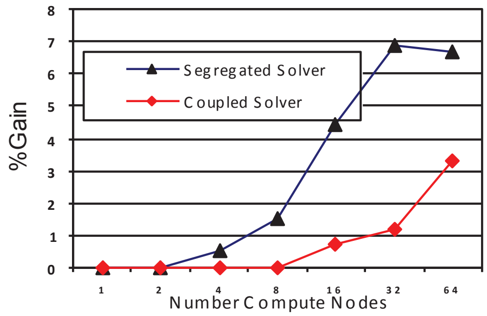

Figure 20a: *LeMansCar* %S_n *FDR/QDR Solvers gain on SGI Rackable with Intel E5-2697 v2 Cluster.*

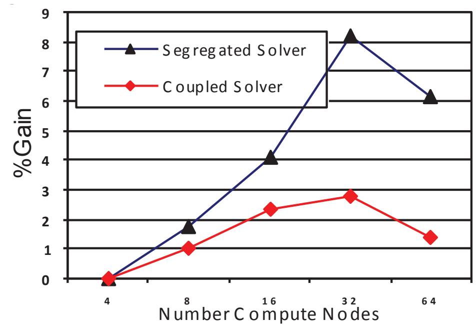

Figure 20b: *Large Classified* %S_n *FDR/QDR Solvers gain on SGI Rackable with Intel E5-2697 v2 Cluster.*

The plots show that FDR-to-QDR percentage gains are higher in the case of the Segregated solver than for the Coupled solver for both models. The gains were observed to be approximately 6-8% and 2-3% for the Segregated and Coupled Solvers, respectively, for tests involving 32 to 64 nodes of SGI Rackable E5-2697 v2 cluster. Note that at the time of writing this paper, this cluster was configured to a maximum of 64 nodes, thus we will expect that these gains to be relatively higher for tests using larger number of nodes, for example 96, 128 and 256 nodes.

5.0 Summary and Conclusions

In this paper an analysis of the parallel performance of the commercial CFD application STAR-CCM+ from CD-adapco has been presented. The analysis has been based on experiments performed on the SGI computer hardware powered by the Intel Xeon E5-2670, E5-2960, E5-2650 (code named Sandy Bridge) and the Xeon E5-2697 v2 and E5-2690 v2 (code named Ivy Bridge) processor families. Three SGI compute hardware platforms were considered, namely the SGI Rackable C2112-4TY14 with Sandy Bridge E5-2670 @ 2.6GHz/8-core, the SGI ICE X with Sandy Bridge E5-2690 @ 2.90GHz/8-core and the SGI UV2000 with Sandy Bridge E5-4650 @ 2.70GHz/8-core. Also used are Ivy Bridge based hardware, namely ICE X with IP113 (Dakota) blades/E5-2690V2 3.0GHz/10-core Ivy Bridge and a C2112-4TY14 Rackable Cluster with Ivy Bridge E5-2697 v2 12-core 2.7GHz.

The analysis is primarily based on two features of the Intel Sandy/Ivy Bridge processors, namely the Turbo boost mode and Hyper-Threading Technology (HTT). For these two features we defined modes of operation denoted by the definitions of 'S_n', 'H_{2n}' ,'T_n' and 'C_{2n}' to correspond to the Standard, Hyper-threaded, Turbo boost and the Combined Hyper-threaded with Turbo boost execution modes respectively.

On the other hand, on the application side, two STAR-CCM+ input models were considered, the LeMansCar17m standard benchmark and a Classified Large model. Experiments were executed using two STAR-CCM+ solvers, namely the Segregated flow solver and the Coupled flow solver. STAR-CCM+ appears to significantly benefit from the Turbo Boost 2.0 feature of the Sandy/Ivy processors family. Experiments have shown that the application gain from turbo boost feature ranges between 6% to 14% depending on the model size and number of compute nodes in a test. Overall experiments have shown that the turbo boost gains are relatively constant with respect to the number of nodes within the parallel scalable part of the model. On the other hand, experiments have also shown that turbo boost mode seem to gain performance in most cases and at high node counts.

For the Hyper-threading feature, experiments in this paper have shown that for STAR-CCM+ this feature appears to show a useful effect on all the above hardware platforms up to a limited number of compute nodes per experiment. Experiments have also shown that hyper-threading gains appears to be larger in the case of the Coupled solver as opposed to the Segregated solver, however this may require more observations to properly validate on a wider scale. In fact, we have observed some models which benefited positively from hyper-threading for only up to 4 or 6 nodes. Hyper-threading feature percentage gains tend to drop gradually to a threshold value relative to an increase in the number of compute nodes beyond which gains drop significantly to negative values. This drop depends also on the number of cores per socket where larger number of cores per socket result in further acceleration in the drop of hyper-threading percentage gains. Thus hyper-threading gains tend to drop at a faster rate in the case of Ivy Bridge processors due to the larger number of cores per socket as opposed to Sandy Bridge. Overall our experiments have shown that the hyper-threading feature may be impaired by the following:

- Hyper-threading will no longer gain performance if the executing model parallel scalability had reached its threshold for that number of domains decomposition. e.g. if a model will normally not scale beyond 128 cores, then executing it as a Hyper-threaded task on 128 cores (or more) will not result in any significant gain.
- Hyper-threading will not gain performance if the hyper-threaded threads data access requirements totally saturate the bandwidth per compute node. This may result in performance degradation.

Experiments have also shown that the application may benefit from the combined use of features of hyper-threading and Turbo Boost for a range of a number of compute nodes involved in the test. The two input model experiments have shown that the combined two-feature effect can provide significant performance gains for up to 16 compute nodes in the case of the segregated solver and up to 32 nodes in the case of the Coupled solver per test. Note that a combined gain may be considered as significant when each of these two features yield a positive performance gain for a test. More importantly the gains of the two features combined can be quantified by the difference between the standard execution time S_n, and gained differences of the two features individually based on the equation:

$$C_{2n} \tilde{=} S_n - \Delta T_n - \Delta H_{2n}$$

Where

$$\Delta T_{n,} = S_{n,} - T_n \quad \text{and} \quad \Delta H_{2n,} = S_n - H_{2n}$$

This makes it possible to determine the value of C_{2n}, as an approximation, without having to run its corresponding test by simply knowing S_n, T_n and H_{2n}. Correspondingly decided if a C_{2n} test will provide a positive result based on the criteria that a C_{2n} test is worth performing if both ΔT_n and ΔH_{2n}, are positive.

A useful approach for gaining the maximum benefit of Hyper-threading and Turbo Boost features may be achieved by implementing the following algorithm. Given an input model and based on a relatively small number of iterations/time-steps execute the model using the following steps:

1. Start with an N number of nodes.
2. Run the model to obtain S_n, T_n and H_{2n}, values.
3. If $\Delta T_n > 0$ and $\Delta H_{2n} > 0$ then increase N and repeat step 2.
4. Else if $\Delta T_n < 0$ or $\Delta H_{2n} < 0$ then stop iterating and use the previous iteration values of S_n, T_n and H_{2n} to calculate the corresponding C_{2n} value using the equations above.

The above algorithm enables users to approximately determine the optimal number of cores and nodes corresponding to the resulting value of C_{2n}. Thus running the full simulation using a C_{2n} mode of operation for an optimal parallel performance.

6.0 References

[1] Platform Manager on SGI Altix® ICE Systems Quick Reference Guide: Chapter 1. SGI Altix ICE 8200 Series System Overview Hardware-Books (document number: 007-5450-002)

[2] SGI ICE X System Hardware User Guide (document number: 007-5806-001, published 2012-03-28)

[3] SGI ICE X Installation and Configuration Guide (document number: 007-5917-002, published: 2013-11-20).

[4] Technical Advances in the SGI® UV™ Architecture. SGI White paper. June, 2012

[5] SGI® Rackable® C2112-4TY14 System User's Guide, 2012. http://techpubs.engr.sgi.com/library/manuals/5000/007-5685-002/pdf/007-5685-002.pdf

[6] A.JASSIM, "STAR-CCM+ Using Parallel Measurements from Intel® Sandy Bridge / Ivy Bridge x86-64 based HPC Clusters. A Performance Analysis", 2014

7.0 About SGI

SGI, the trusted leader in high performance computing (HPC), is focused on helping customers solve their most demanding business and technology challenges by delivering technical computing, Big Data analytics, cloud computing, and petascale storage solutions that accelerate time to discovery, innovation, and profitability.

For more information please contact an SGI sales representative at 1-800-800-7441 or visit www.sgi.com/contactus.

Global Sales and Support: sgi.com/global