Dell HPC System for Manufacturing—System Architecture and Application Performance

This Dell technical white paper describes the architecture of the Dell HPC System for Manufacturing and discusses performance benchmarking results and analysis for ANSYS Fluent, ANSYS Mechanical, CD-adapco, a Siemens Business, STAR-CCM+, LSTC LS-DYNA, NICE DCV, and EnginFrame.

Authors

Joshua Weage
Garima Kochhar

A Dell Technical White Paper

Revisions

Date	Description
July 2016	Initial release

Contents

Executive Summary

This technical white paper describes the architecture of the Dell HPC System for Manufacturing, which consists of building blocks configured specifically for applications in the manufacturing domain. Detailed performance results with sample CFD and FEA applications, and power characteristics of the system are presented in this document. Virtual Desktop Infrastructure (VDI) capability is also described and validated.

1   Introduction

This technical white paper describes the Dell HPC System for Manufacturing. The Dell HPC System for Manufacturing is designed and configured specifically for customers in the manufacturing domain. This system uses a flexible building block approach, where individual building blocks can be combined to build HPC systems which are ideal for customer specific work-loads and use cases. The individual building blocks are configured to provide good performance for specific application types and workloads common to this industry. This whitepaper discusses the types of applications commonly used in the manufacturing domain, describes the building blocks that work well for each application, and provides measured, quantified performance results for the system.

The architecture of the Dell HPC System for Manufacturing, a description of the building blocks and the rationale behind the design choices are presented in Section 2. Section 3 describes the system configuration, software and application versions, and the benchmark test cases that were used to measure and analyze the performance of the Dell HPC System for Manufacturing. Section 4 quantifies the capabilities of the system and presents benchmark performance from four CAE applications. The applications used for benchmarking include ANSYS® Fluent®, ANSYS Mechanical™, CD-adapco® STAR-CCM+® and LSTC LS-DYNA®. Section 5 documents the power requirements for the solver building blocks and Section 6 discusses the VDI capabilities available in the system.

2   System Building Blocks

The Dell HPC System for Manufacturing is assembled by using preconfigured building blocks. The available building blocks are infrastructure servers, storage, networking, Virtual Desktop Infrastructure (VDI), and application specific compute building blocks. These building blocks are preconfigured to provide good performance for typical applications and workloads within the manufacturing domain. The building block architecture allows for a custom HPC system for specific end-user requirements, while still making use of standardized, domain-specific building blocks. This section describes the available building blocks along with the rationale of the system configurations.

2.1   Infrastructure Servers

The infrastructure servers are used to administer the system and enable user access. They are not actively involved in computation or storage, but they provide services that are critical to the overall HPC system. Typically these servers are the master nodes and the login nodes.

For small sized clusters, a single physical server can provide all these functions. It can also be used for storage, by using NFS, where it must be configured with additional disk drives or an external storage array.

One master node is mandatory and is required to deploy and manage the system. If high-availability (HA) functionality is required, two master nodes are necessary. Login nodes are optional and one login server per 30-100 users is recommended.

The recommended configuration for infrastructure servers is:

- Dell PowerEdge R630 server
- Dual Intel® Xeon® E5-2680 v4 processors
- 128 GB of memory, 8 x 16 GB 2400 MT/s DIMMs
- PERC H730 RAID controller
- 2 x 600 GB 10K SAS drives in RAID 1
- Dell iDRAC8 Enterprise
- 2 x 1100 W power supply units (PSUs)
- EDR InfiniBand™ (optional)

The recommended configuration for the infrastructure server is described here. The PowerEdge R630 server is suited for this role. A cluster will have only a small number of infrastructure servers; therefore, density is not a concern, but manageability is more important. The Intel Xeon E5-2680 v4 processor is a mid-bin 120 W, 14c CPU which is sufficient for this role. 128 GB of memory by using 8x16 GB DIMMs provides sufficient memory capacity, with minimal cost/GB, while also providing good memory bandwidth. These servers are not expected to perform much I/O. Therefore, two drives are sufficient and RAID 1 is selected to protect the OS on these servers. For small systems (four nodes or less), a Gigabit Ethernet network may provide sufficient performance. For most other systems, EDR InfiniBand is likely to be the data interconnect of choice, which provides a high throughput, low latency fabric for node-node communications, or access to a Dell NFS Storage Solution (NSS) or Dell Intel Enterprise Edition for Lustre (IEEL) storage solution.

2.2 Explicit Building Blocks

Explicit Building Block (EBB) servers are typically used for Computational Fluid Dynamics (CFD) and explicit Finite Element Analysis (FEA) solvers such as Abaqus/Explicit, Altair® RADIOSS™, ANSYS Fluent, CD-adapco STAR-CCM+, ESI PAM-CRASH, Exa PowerFLOW®, LSTC LS-DYNA, and OpenFOAM®. These software applications typically scale well across many processor cores and multiple servers. The memory capacity requirements are typically modest and these solvers perform minimal disk I/O while solving. In most HPC systems for manufacturing, the large majority of servers are EBBs.

The recommended configuration for EBBs is:

- Dell PowerEdge C6320 server
- Dual Intel Xeon E5-2697A v4 processors
- 128 GB of memory, 8 x 16 GB 2400 MT/s DIMMs
- Mezzanine SAS RAID controller
- 2 x 600 GB 10K SAS drives in RAID 0
- Dell iDRAC8 Express
- 2 x 1600 W PSUs in the chassis
- EDR InfiniBand (optional)

The recommended configuration for the EBB servers is described here. Because the largest percentage of servers in the majority of systems will be EBB servers, a dense solution is important; therefore, the PowerEdge C6320 server is selected. The Intel Xeon E5-2697A v4 processor is a 16-core CPU running at 2.6 GHz with a maximum all-core turbo frequency of 3.1 GHz. 32 cores per server provides a dense compute solution, with good memory bandwidth per core, and a power of two quantity of cores. The maximum all-core turbo frequency of 3.1 GHz is important because EBB applications are typically CPU bound. This CPU model provides the best balance of CPU cores and core speed. 128 GB of memory using 8x16GB DIMMs provides sufficient memory capacity, with minimal cost per GB, while also providing good memory bandwidth. Relevant applications typically perform limited I/O while solving. Therefore, the system is configured with two disks in RAID 0 by using the SAS mezzanine card, leaving the PCIe slot available for the EDR InfiniBand HCA. The compute nodes do not require extensive out-of-band (OOB) management capabilities and therefore an iDRAC8 Express is sufficient. For small systems (four nodes or less), a Gigabit Ethernet network may provide sufficient performance. For most other systems, EDR InfiniBand is likely to be the data interconnect of choice, which provides a high throughput, low latency fabric for node-node communications, or access to an NSS or IEEL storage solution.

The Intel Xeon E5-2697A v4 processor is a good choice for many FEA and CFD solvers; however, the specific processor model used for the EBB system can be selected based on application requirements. The Intel Xeon E5-2667 v4 processor is a good choice when maximum per core performance is important. This processor may also be appropriate for minimizing ISV software licensing costs. This would be an untested EBB configuration; however, the performance should be similar to the Implicit Building Block for non-I/O intensive applications.

2.3 Implicit Building Blocks

Implicit Building Block (IBB) servers are typically used for implicit FEA solvers such as Abaqus/Standard, Altair OptiStruct®, ANSYS Mechanical™, MSC™ Nastran™, and NX® Nastran. These applications typically have large memory requirements and do not scale to as many cores as the EBB applications. They also often have a large drive I/O component.

The recommended configuration for IBBs is:

- Dell PowerEdge R630 server
- Dual Intel Xeon E5-2667 v4 processors
- 512 GB of memory, 16 x 32 GB 2400 MT/s DIMMs
- PERC H730 RAID controller
- 8 x 300 GB 15K SAS drives in RAID 0
- Dell iDRAC8 Express
- 2 x 1100 W PSUs
- EDR InfiniBand (optional)

The recommended configuration for the IBB servers is described here. Typically, a smaller percentage of the system will be comprised of IBB servers. Because of the memory and disk drive requirements explained here, a 1U PowerEdge R630 server is a good choice. A dense PowerEdge C6320 is not needed, particularly when a cluster may have fewer than 4 IBB servers to fill a C6300 chassis. The Intel Xeon E5-2667 v4 processor is an eight-core CPU running at 3.2 GHz with a max all-core turbo frequency of 3.5 GHz. The E5-2667 v2 was a very popular choice for IBBs in the Ivy Bridge time frame and the v4 Broadwell version of this SKU addresses many of the deficiencies of the v3 Haswell version. 16 cores in the server is sufficient for IBB workloads that do not scale to many cores and the 3.5 GHz maximum all-core turbo frequency is one of the fastest available for Broadwell processors. The E5-2667 v4 provides sufficient memory bandwidth per core (117 GBps per CPU). 32 GB DIMMs are used to provide the larger memory capacities needed for these applications in a 2 DIMM per channel configuration to maximize memory bandwidth. IBB applications have large disk drive I/O requirements and eight 15K SAS disk drives in RAID 0 are used to provide fast local I/O. The compute nodes do not require extensive OOB management capabilities and therefore an iDRAC8 Express is recommended.

InfiniBand is not typically necessary for IBBs because most uses cases require only running applications on a single IBB; however, an InfiniBand HCA can be added to enable multi-server analysis or access to an NSS or IEEL storage solution.

2.4 Implicit GPGPU Building Blocks

Some IBB-relevant applications are GPU enabled. Therefore, to support customers using GPU accelerated applications, a GPU version of the IBB is available. Similar to IBB servers, Implicit GPGPU Building Blocks (IGPGPUBB) are typically used for implicit FEA solvers such as Abaqus/Standard, Altair OptiStruct, ANSYS Mechanical, MSC Nastran, and NX Nastran. The IGPGPUBB servers have similar requirements to IBBs with the addition of GPUs. IGPGPUBB's may also be appropriate when a general purpose compute infrastructure is desired to support both GPU accelerated and non-GPU accelerated workloads.

The recommended configuration for IGPGPUBBs is:

- Dell PowerEdge R730 server
- Dual Intel Xeon E5-2680 v4 processors
- 256 GB of memory, 16 x 16 GB 2400 MT/s DIMMs
- PERC H730 RAID controller
- 8 x 300 GB 15K SAS drives in RAID 0
- Dell iDRAC8 Express
- 2 x 1100 W PSUs
- One NVIDIA® Tesla® K80
- EDR InfiniBand (optional)

The recommended configuration for the IGPGPUBB servers is described here. A Dell PowerEdge R730 server is required to support the NVIDIA Tesla K80. Relevant GPU-enabled applications typically do not scale beyond two GPUs. Therefore, one K80 is sufficient. The Intel Xeon E5-2680 v4 processor is a 14-core CPU running at 2.4 GHz with a maximum all-core turbo frequency of 2.9 GHz. Because the GPU is providing significant computing power for the IGPGPUBB, a high-end CPU is not required. The E5-2680 v4 is a cost effective, mid-bin processor, which is sufficient for this server.

The IBB used 32 GB DIMMs, but for the IGPGPUBB, 16 GB DIMMs are sufficient for GPU-optimized workloads. 16 GB DIMMs in a 2 DIMM per channel configuration provides balanced memory capacity (256 GB) and memory bandwidth. Eight 15K SAS disk drives in RAID 0 are used to support the large disk drive I/O requirements. Compute nodes do not require extensive OOB management capabilities and therefore an iDRAC8 Express is recommended.

InfiniBand is not typically necessary for IGPGPUBBs because most uses cases require only running applications on a single server; however, an InfiniBand HCA can be added to enable multi-server analysis or access to an NSS or IEEL storage solution.

2.5   Virtual Desktop Infrastructure Servers

Virtual Desktop Infrastructure (VDI) servers provide remote desktop services that are co-located with the compute resources. This allows for centralized management of visualization resources and reduces the movement of data out of the datacenter. Users can be located in the same building as the datacenter or anywhere with network access. The goal is to provide the user with a visual and interactive experience as if they were working on a local workstation, while providing direct access to the resources of the HPC system. If VDI is required, an HPC System for Manufacturing can be configured with multiple VDI servers, with a recommendation to include one VDI server per 1-4 power users or 1-16 standard users.

The recommended configuration for VDI servers is:

- Dell PowerEdge R730 server
- Dual Intel Xeon E5-2680 v4 processors
- 512 GB of memory, 16 x 32 GB 2400 MT/s DIMMs
- PERC H730 RAID controller
- 2 x 600 GB 10K SAS drives in RAID 1 (for operating system)

- 4 x 600 GB 10K SAS drives in RAID 0 (for local temporary storage)
- QLogic 10 GbE Network Daughter Card (NDC)
- Dell iDRAC8 Express
- 2 x 1100 W PSUs
- One NVIDIA GRID® K2
- EDR InfiniBand (optional)

The recommended configuration for the VDI server is described here. A PowerEdge R730 is required to support the NVIDIA GRID K2 which is used to provide hardware accelerated 3D graphics. The Intel Xeon E5-2680 v4 with 14 cores at 2.4 GHz (maximum all-core turbo of 2.9 GHz) is a cost effective processor for VDI and provides uniformity with the infrastructure building block servers. VDI systems require a large amount of memory in order to support multiple users, and therefore 32GB DIMMs are recommended. A 2 DIMM per channel configuration maximizes available memory bandwidth. 600 GB 10K drives provide sufficient performance and capacity, and allow part commonality between infrastructure, VDI and EBB servers. A 10 GbE NDC is recommended to provide sufficient network bandwidth for multiple simultaneous users. The management capabilities provided by an iDRAC8 Express are sufficient for VDI.

InfiniBand is optional on the VDI system; however, an InfiniBand HCA is required to access an NSS or IEEL storage solution.

2.6   Dell NSS-HA Storage

The Dell NFS Storage Solution (NSS) provides a tuned NFS storage option that can be used as primary storage for user home directories and application data. The current version of NSS is NSS7.0-HA with options of 240 TB or 480 TB raw disk space. NSS is an optional component and a cluster can be configured without NSS.

NSS-HA is the high performance computing network file system (NFS) storage solution of Dell, optimized for performance, availability, resilience, and data reliability. The best practices used to implement this solution result in better throughput compared to non-optimized systems. A high availability (HA) setup, with an active-passive pair of servers, provides a reliable and available storage service to the compute nodes. The HA unit consists of a pair of Dell PowerEdge R730 servers. A Dell PowerVault MD3460 dense storage enclosure provides 240 TB of storage for the file system with 60 x 4 TB, 7.2K near-line SAS drives. This unit can be extended with a PowerVault MD3060e to provide an additional 240 TB of disk space for the 480 TB solution. Each of the PowerVault arrays is configured with 6 virtual disks (VDs). Each VD consists of 10 hard drives configured in RAID6 (8+2).

The NFS server nodes are directly attached to the dense storage enclosures by using 12 Gbps SAS connections. NSS7.0-HA provides two network connectivity options for the compute cluster: EDR InfiniBand and 10 Gigabit Ethernet. The active and passive NFS servers run Red Hat Enterprise Linux (RHEL) 7.2 with Red Hat's Scalable File System (XFS) and Red Hat Cluster Suite to implement the HA feature.

2.7   Dell IEEL Storage

Dell IEEL storage is an Intel Enterprise Edition for Lustre (IEEL) based storage solution consisting of a management station, Lustre metadata servers, Lustre object storage servers, and the associated backend storage. The management station provides end-to-end management and monitoring for the entire Lustre storage system.

The Dell IEEL storage solution provides a parallel file system with options of 480 TB or 960 TB raw storage disk space. This solution is typically used for scratch space for larger clusters.
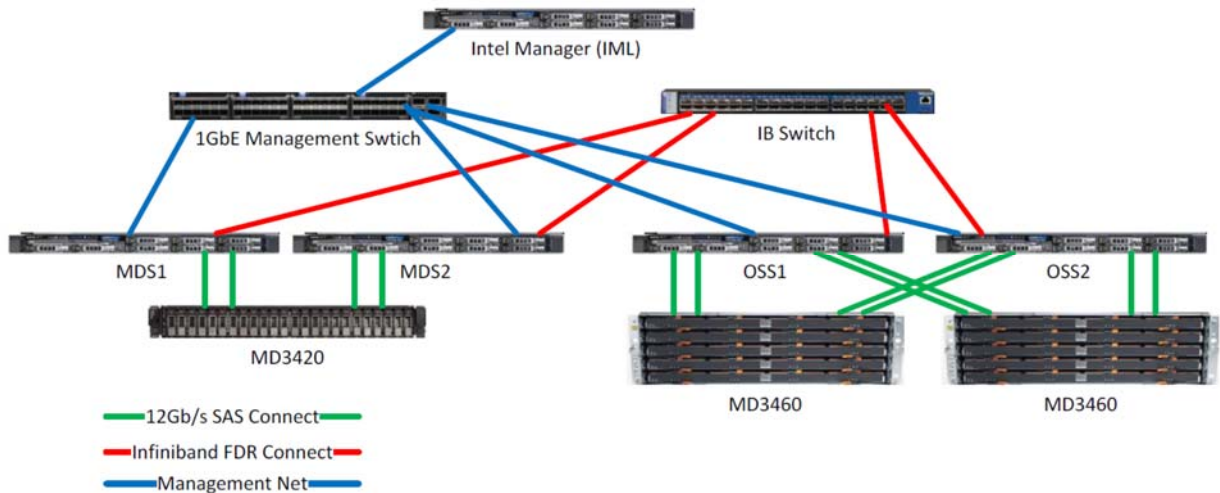
Figure 1 Overview of the Dell IEEL Components and Connectivity

2.8   System Networks

Most Dell HPC Systems for Manufacturing are configured with two networks—an administration network and a high-speed/low-latency switched fabric.

The administration network is Gigabit Ethernet that connects to the onboard LOM/NDC of every server in the cluster. This network is used for provisioning, management and administration. On the EBB, IBB, IGPGPUBB and VDI servers, this network will also be used for IPMI hardware management. For infrastructure and storage servers, the iDRAC Enterprise ports may be connected to this network for OOB server management. The heartbeat ports for NSS-HA and IEEL Gigabit management ports may also be connected to this network.

Each rack in the system contains at least one Dell Networking S3048-ON Ethernet switch. If there is more than one switch in the system, multiple switches will be stacked with 10 Gigabit Ethernet stacking cables.

A high-speed/low-latency fabric is recommended for clusters with more than four servers. The current recommendation is an EDR InfiniBand fabric. The fabric will typically be assembled by using Mellanox

SB7790 36-port EDR InfiniBand switches. The number of switches required depends on the size of the cluster and the blocking ratio of the fabric.

2.9   VDI Software

The VDI software is installed on the VDI servers and provides GPU accelerated remote visualization capabilities. NICE DCV with NICE EnginFrame is the recommended VDI software stack for the Dell HPC System for Manufacturing.

2.10   Cluster Software

The Cluster Software is used to install and monitor the system's compute servers. Bright Cluster Manager 7.2 (BCM) is the recommended cluster software.

The PBS Professional® distributed workload manager has been validated for this system. PBS Professional is the recommended workload manager for the Dell HPC System for Manufacturing.

2.11   Services and Support

The Dell HPC System for Manufacturing is available with full hardware support and deployment services, including NSS-HA and IEEL deployment services.

3 Reference System

The reference system was assembled in the Dell HPC Innovation Lab by using the system building blocks described in Section 2. The building blocks used for the reference system are listed in Table 1.

Table 1. Reference System Configuration

Building Block	Quantity
Infrastructure Server	1
Explicit Building Block with EDR InfiniBand	8
Implicit Building Block	2
Implicit GPGPU Building Block	1
VDI Server	1
Dell Networking S3048-ON Ethernet Switch	1
Mellanox SB7790 EDR InfiniBand Switch	1

The BIOS configuration options used for the reference system are listed in Table 2.

Table 2. BIOS Configuration Options

BIOS Option	Setting
Logical Processor	Disabled
Memory Snoop Mode	Opportunistic Snoop Broadcast
Node Interleaving	Disabled
System Profile	Performance Profile DAPC Profile also used for idle power measurements
IONonPostedPrefetch	Disabled

The software versions used for the reference system are listed in Table 3.

Table 3.   Software Versions

Component	Version
Operating System	RHEL 7.2
Kernel	3.10.0-327.el7.x86_64
OFED	Mellanox 3.2-2.0.0.0
Bright Cluster Manager	7.2 with RHEL 7.2 (Dell version)
ANSYS Fluent	v17.1
ANSYS Fluent Benchmarks	v15 and v16
ANSYS Mechanical	v17.1
ANSYS Mechanical Benchmarks	v17.0
CD-adapco STAR-CCM+	11.02.010 mixed precision
CD-adapco STAR-CCM+ Benchmarks	Benchmark cases as listed
HPL	xhpl_intel64 from Intel 2016.1.150 MKL v1.5.2.16 for CUDA enabled
LSTC LS-DYNA	mpp_s_r8_1_105897_avx2 Platform MPI 9.1.0.1 Intel MPI 5.1.2.150
LSTC LS-DYNA Benchmarks	car2car-ver10 with endtime=0.02 ODB10M-ver16 with endtime=0.02
NICE	DCV 2016.0-16811 EnginFrame 2015.1
NVIDIA Driver	352.79
PBS Professional	13.0.2
STREAM	5.10 compiled using Intel compiler 2016.1.150

4 System Performance

This section presents the performance results obtained from the reference system described in Section 3. Basic performance of the servers was measured first, prior to any application benchmarking. This was done to ensure that individual server sub-systems were performing as expected and that the systems were stable. The STREAM memory bandwidth test was used to check the memory performance and HPL was used to check the computational subsystem, power delivery and to stress test the individual servers. After basic system performance was verified, the Fluent, ANSYS Mechanical, STAR-CCM+, and LS-DYNA benchmark cases were measured on the system.

4.1 STREAM

The STREAM benchmark results for the computational building blocks are listed in Table 4. The results in the table are the minimum, maximum, and average memory bandwidth from the Triad test for three runs of STREAM on all of the building blocks in the reference system. These results demonstrate sustained memory bandwidth of 129 GBps for the Explicit building block, 117 GBps for the Implicit Building Block, and 129 GBps for the Implicit GPGPU building block. This performance is as expected.

The memory bandwidth for the Implicit building block is less than the other two building blocks because of the specific processor selected for this system. The Intel Xeon E5-2667 v4 has only a single memory controller; whereas, the processors used in the other two building blocks have two memory controllers. This means that the total available memory bandwidth per processor is less; however, memory bandwidth per core is high because the E5-2667 v4 is an 8-core processor.

The memory bandwidth of the master node and VDI system were also verified prior to application benchmarking.

Table 4.   STREAM Benchmark Results

Building Block	Triad MBps (min)	Triad MBps (max)	Triad MBps (avg)
Explicit	129,410	129,927	129,665
Implicit	117,528	117,918	117,712
Implicit GPGPU	129,063	129,350	129,222

4.2 High Performance Linpack (HPL)

High Performance Linpack (HPL) is a popular benchmark that is computationally intensive. It is used to rank the TOP500 fastest supercomputers in the world and is an important burn-in test. Although not usually representative of real-world application performance, as a burn-in test, it helps to quickly identify unstable components and verify the power delivery to the system.

The precompiled HPL binary from Intel MKL is used in this test. The results for individual computational building blocks are listed in Table 5. This table presents the minimum and maximum result from three runs of HPL for all of the individual building blocks in the system. The variation observed for the eight Explicit

building blocks is 11%; however, the slowest result is still within specification for the relevant processor model.

Results of running HPL in parallel on the eight Explicit building blocks are presented in Figure 2. This bar chart shows HPL performance from one to eight servers. This test demonstrates good scalability of the system with up to eight Explicit building blocks.

Table 5. HPL Performance of Individual Building Blocks

Building Block	GFLOPS (min)	GFLOPS (max)
Explicit	1,122	1,249
Implicit	763	809
Implicit GPGPU (CPU Only)	1,041	1,043
Implicit GPGPU (CUDA enabled)	1,851	1,865
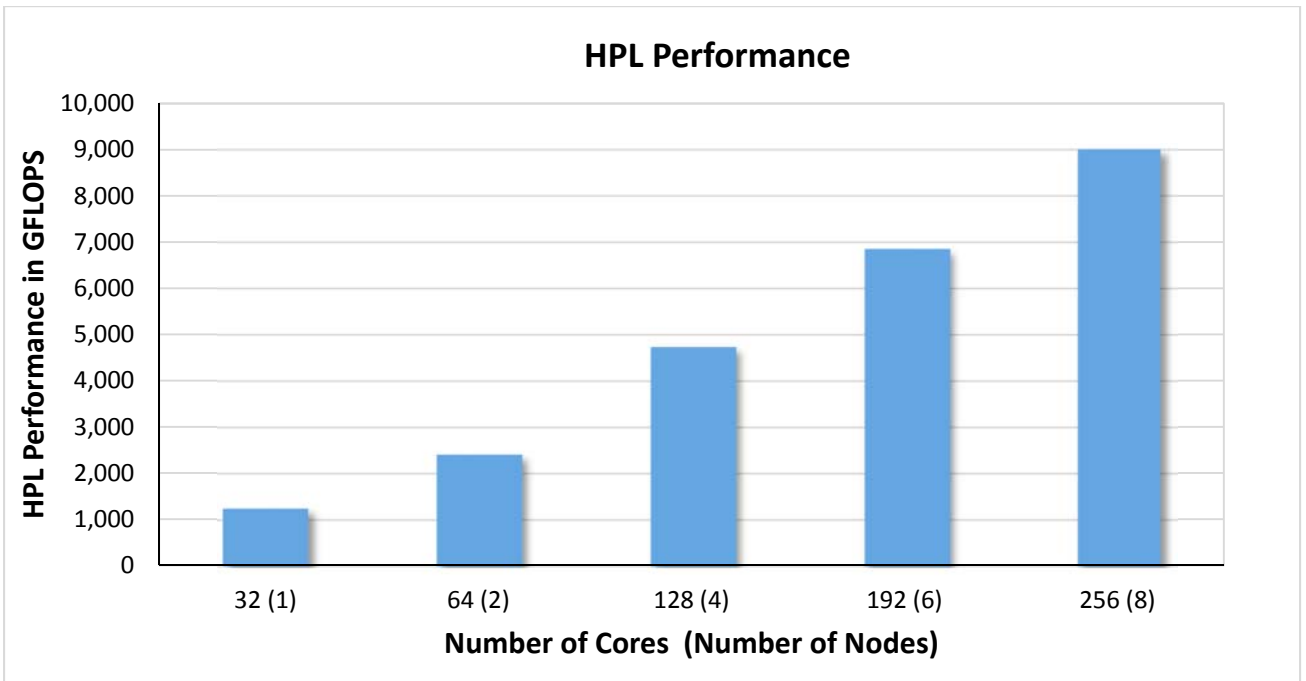
Figure 2 HPL Performance of Explicit Building Blocks

4.3 ANSYS Fluent

ANSYS Fluent is a multi-physics Computational Fluid Dynamics (CFD) software commonly used in multiple engineering disciplines. CFD applications typically scale well across multiple processor cores and servers, have modest memory capacity requirements, and perform minimal disk I/O while solving. For these types of application characteristics, the Explicit building block servers are appropriate. Fifteen benchmark problems from Fluent benchmark suite v16 were evaluated on the EBB servers in the reference system.

The results for Fluent are presented by using the Solver Rating metric which counts the number of 25 iteration solves that can be completed in a day. That is, <total seconds in a day>/<25 iteration solve time in seconds>. A higher value represents better performance.

Figure 3 shows the relative performance of the three compute building block types for eight of the ANSYS Fluent benchmarks. For this comparison, all processor cores in the individual building blocks are utilized while running ANSYS Fluent; GPUs are not used. This comparison demonstrates that for Fluent, application performance is primarily determined by processor performance. The Intel Xeon E5-2697A v4 processor used in the Explicit BB is a good choice for Fluent.
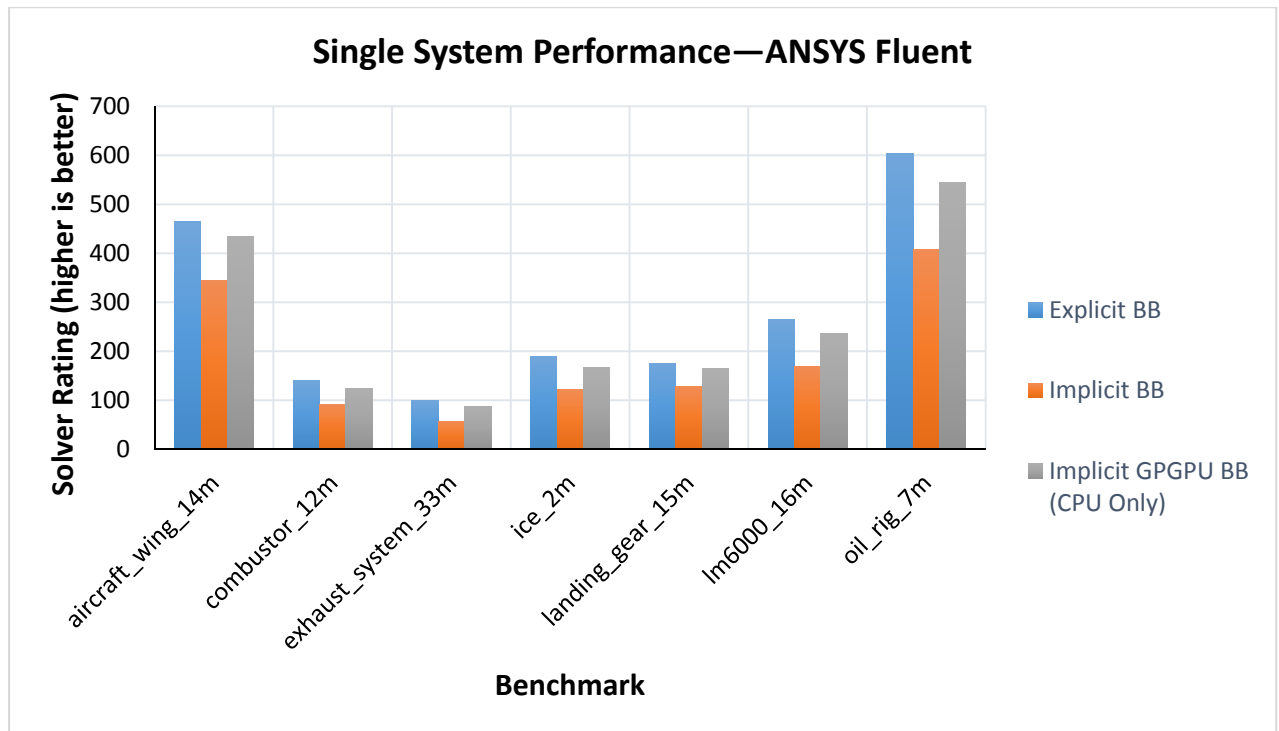
Figure 3 Performance of Individual Building Blocks—ANSYS Fluent

The bar charts in Figure 4 through Figure 7 show the measured performance of the reference system, on one to eight EBBs, by using 32 to 256 cores. Each data point on the graphs records the performance of the specific benchmark data set by using the number of cores marked on the horizontal axis in a parallel simulation. The results are divided into four charts for easy readability—the scale for Solver Rating is large

and some models run much faster than others depending on the number of cells in the model, type of solver used and physics of the problem.

Combustor_71m, f1_racecar_140m and open_racecar280m are large models that require two or more servers for sufficient memory capacity. The results for these cases start with the first valid result obtained for the specific problem.
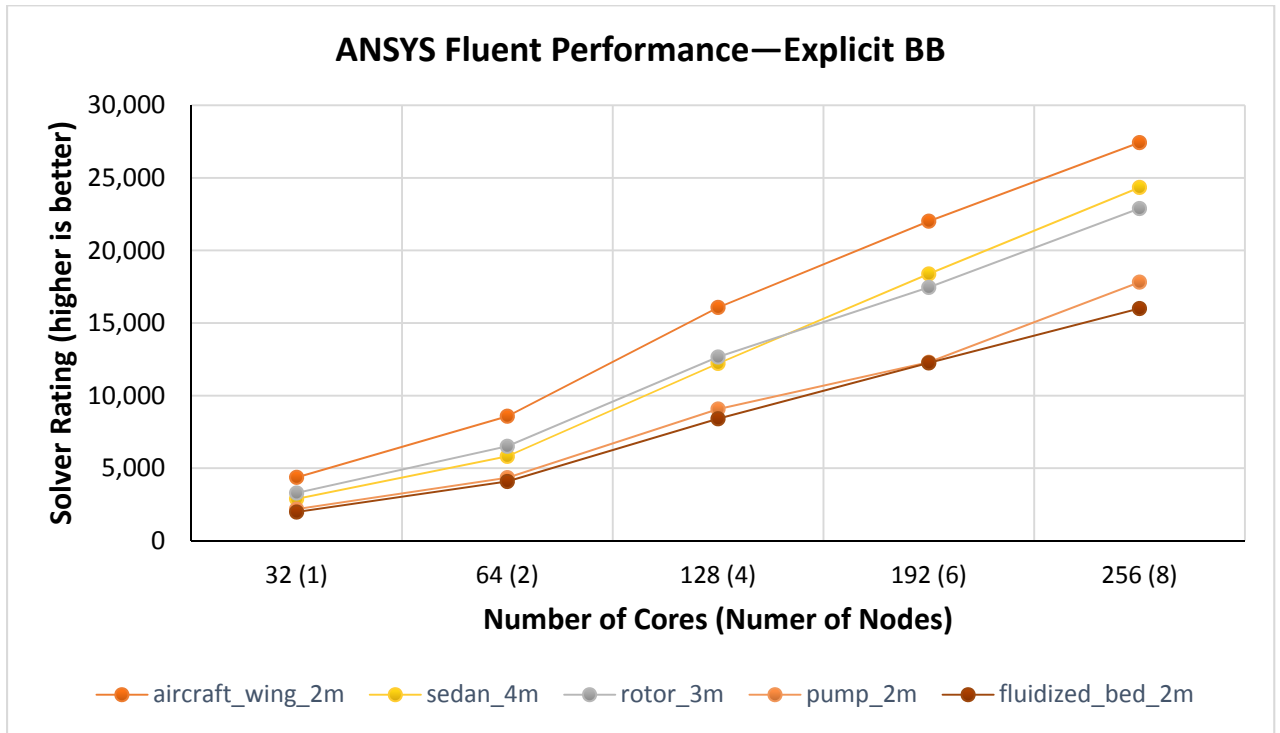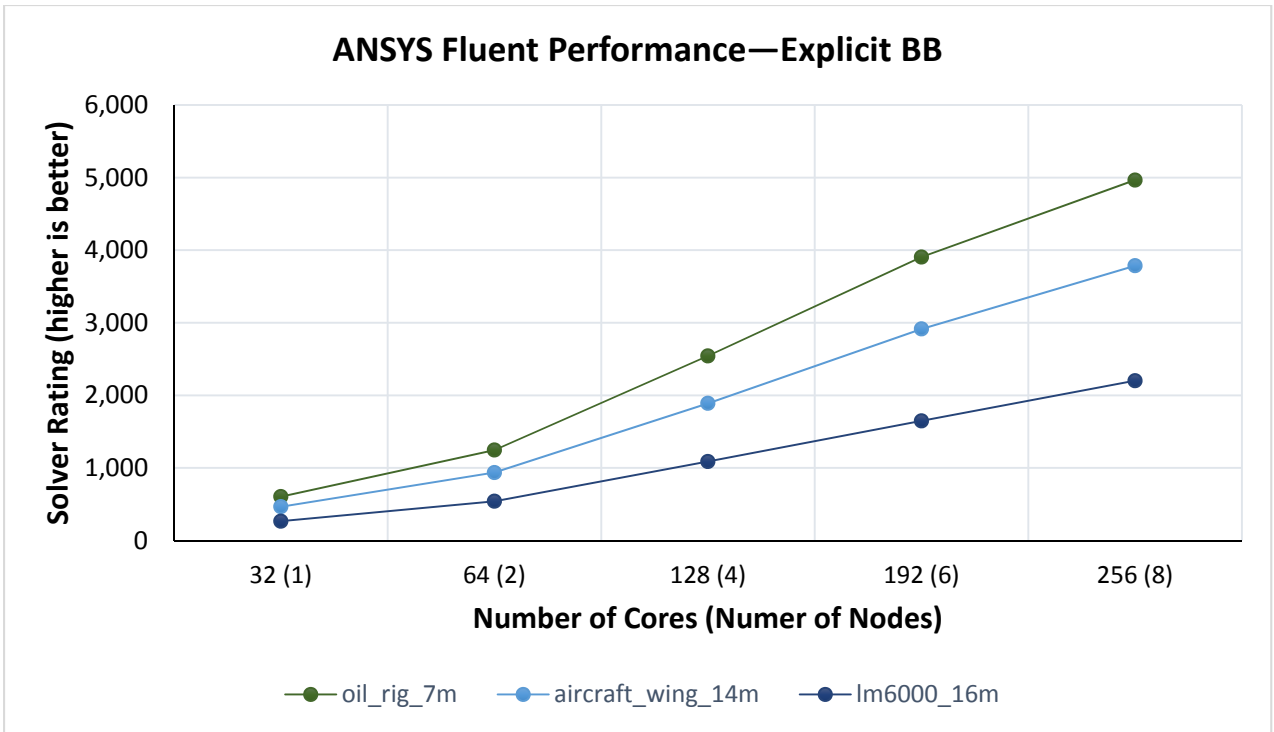
ANSYS Fluent Performance—Explicit BB
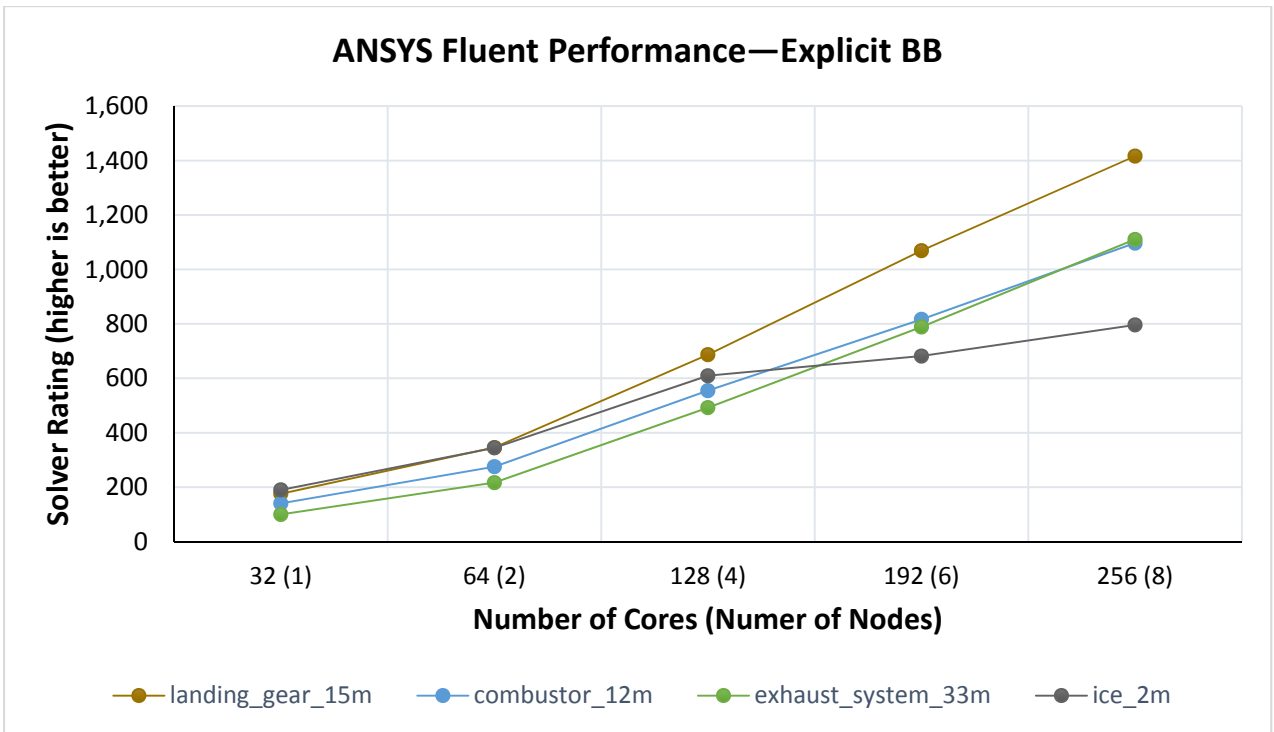

Figure 4 ANSYS Fluent Performance—Explicit BB (1)

Figure 5 ANSYS Fluent Performance—Explicit BB (2)

Figure 6 ANSYS Fluent Performance—Explicit BB (3)

ANSYS Fluent Performance—Explicit BB

ANSYS Fluent Performance—Explicit BB

Y-axis: Solver Rating (higher is better), 0 to 300

X-axis: Number of Cores (Numer of Nodes): 64 (2), 128 (4), 192 (6), 256 (8)

Legend: combustor_71m, f1_racecar_140m, open_racecar_280m
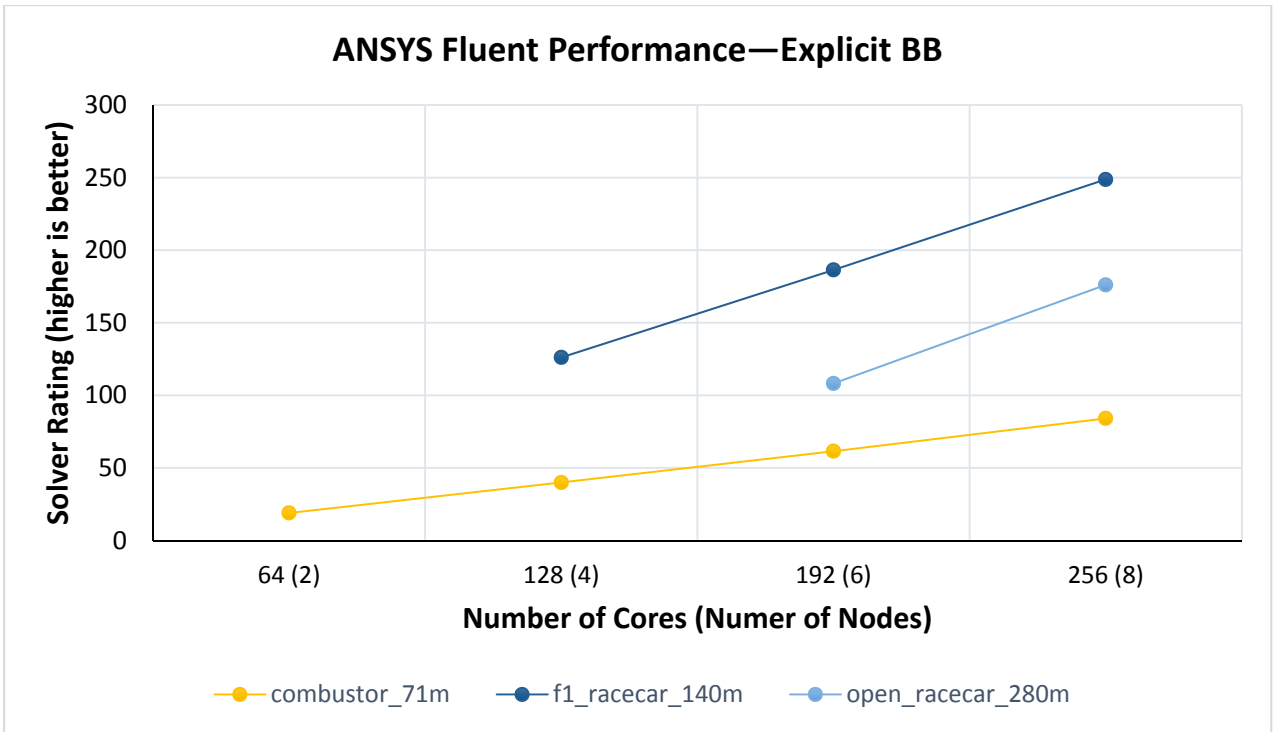
Figure 7 ANSYS Fluent Performance—Explicit BB (4)

Figure 8 through Figure 10 present the same performance data, but plotted relative to the "32-cores (1 node)" result. It makes it easy to see the scaling of the solution—the performance improvement as more cores are used for the analysis. Problem scalability depends on the cell count and physics being simulated for the specific benchmark problem. For many of the problems, scaling is nearly linear. The results shown in Figure 8 are for relatively small problems. Therefore, some of these cases scale well and some do not. In Figure 10, Ice_2m does not scale as well beyond 128 cores or 4 nodes; this is expected because of the physics of this problem.

Figure 11 presents relative scaling data for the three larger benchmark problems. For these cases, results are plotted relative to the first valid data point for each case.
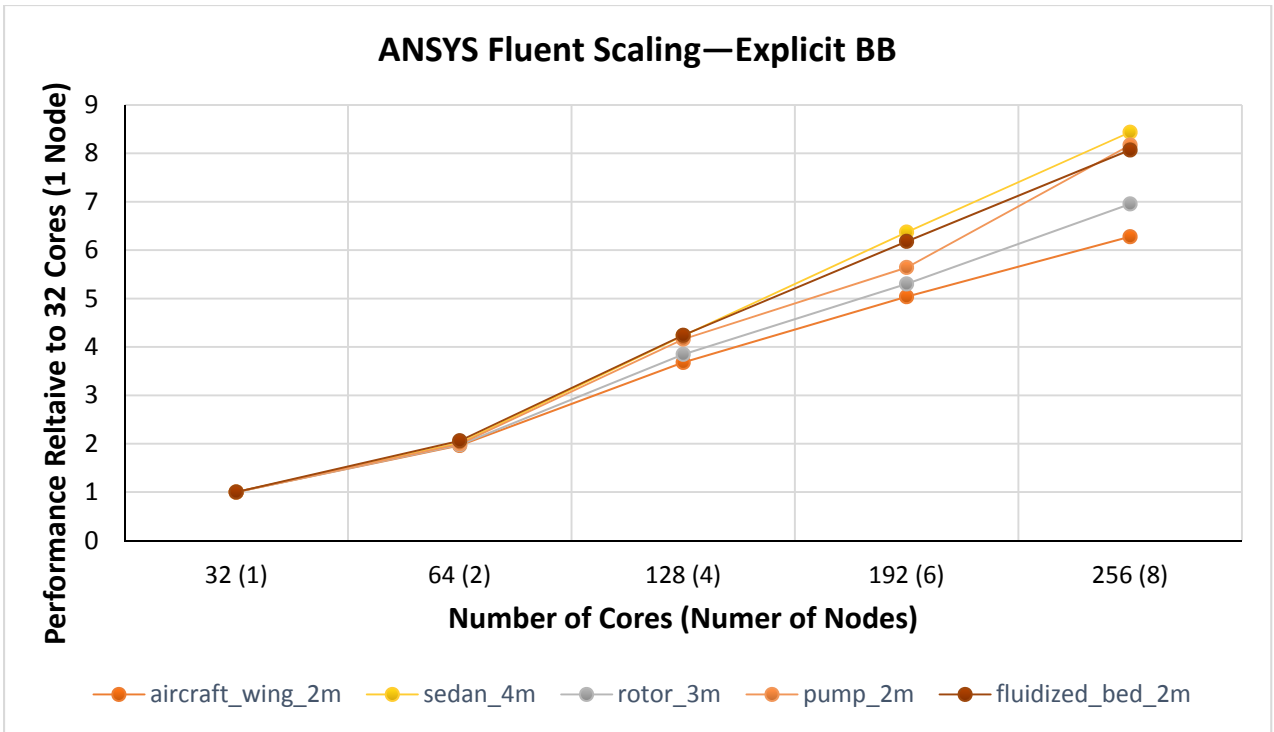
Figure 8 ANSYS Fluent Scaling—Explicit BB (1)
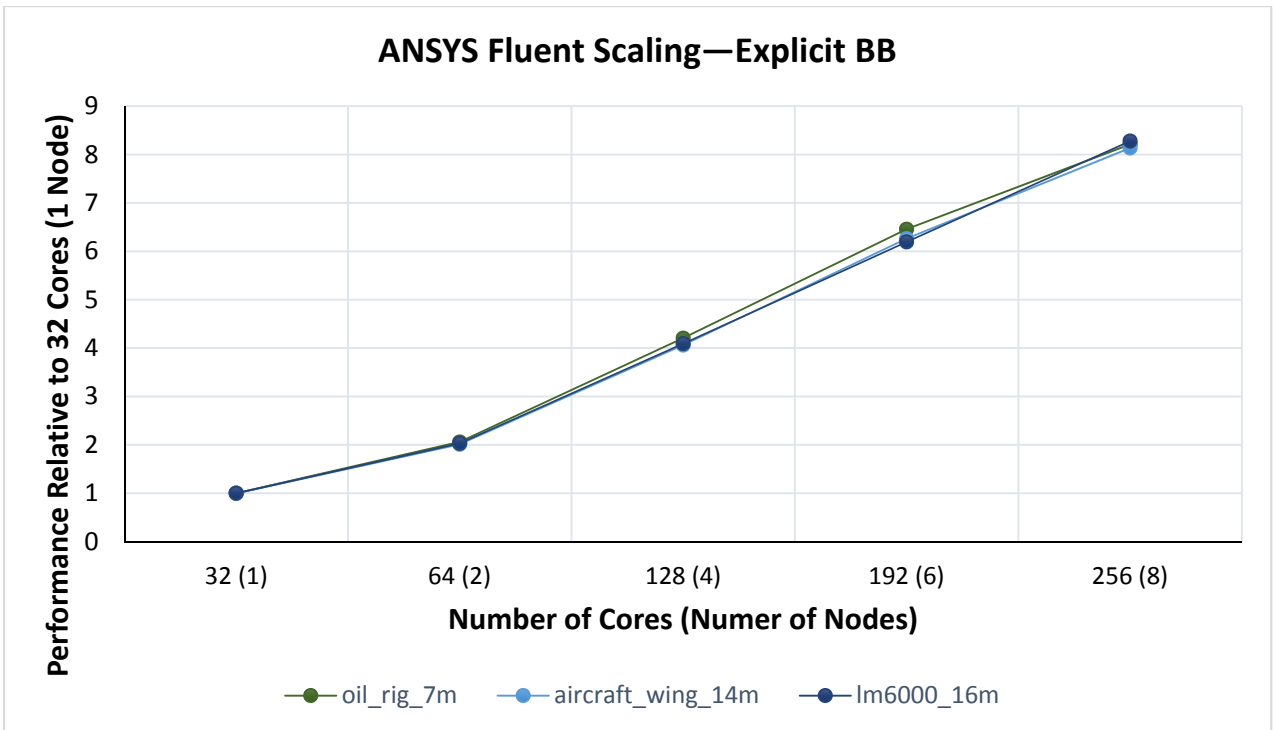
Figure 9 ANSYS Fluent Scaling—Explicit BB (2)

ANSYS Fluent Scaling—Explicit BB
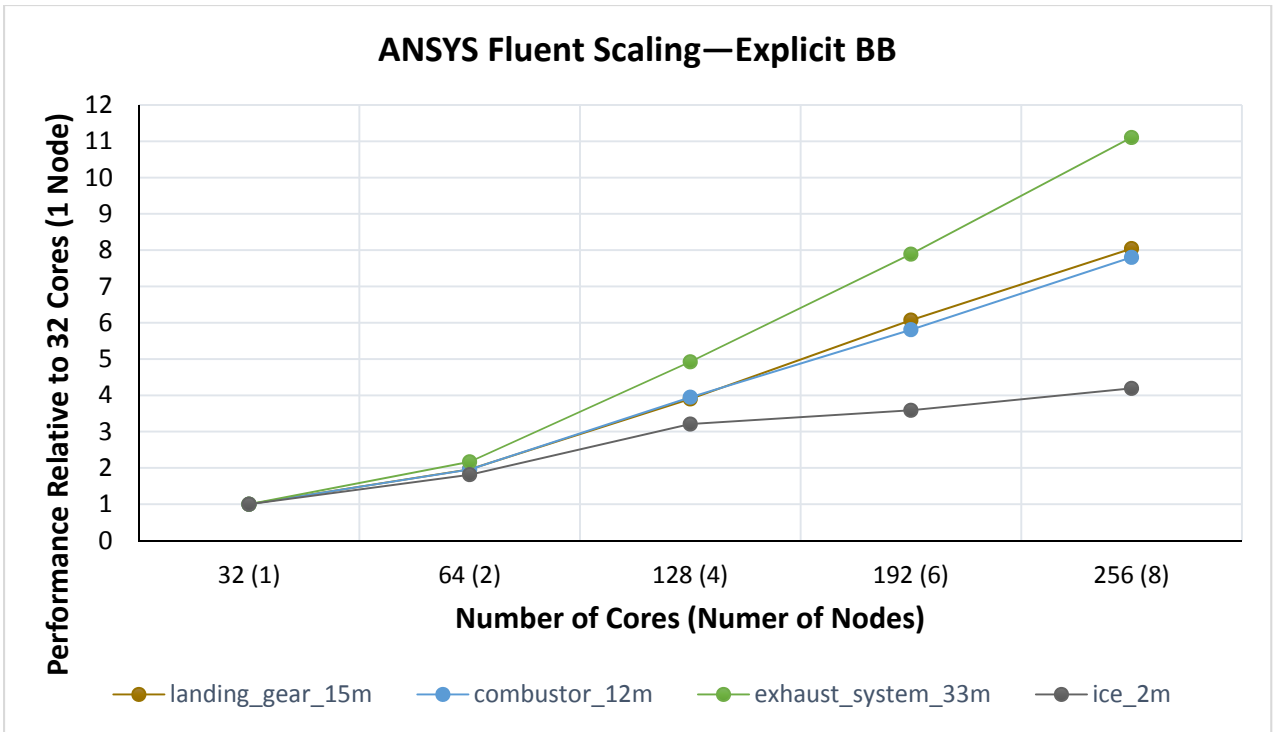
Figure 10 ANSYS Fluent Scaling—Explicit BB (3)

ANSYS Fluent Scaling—Explicit BB
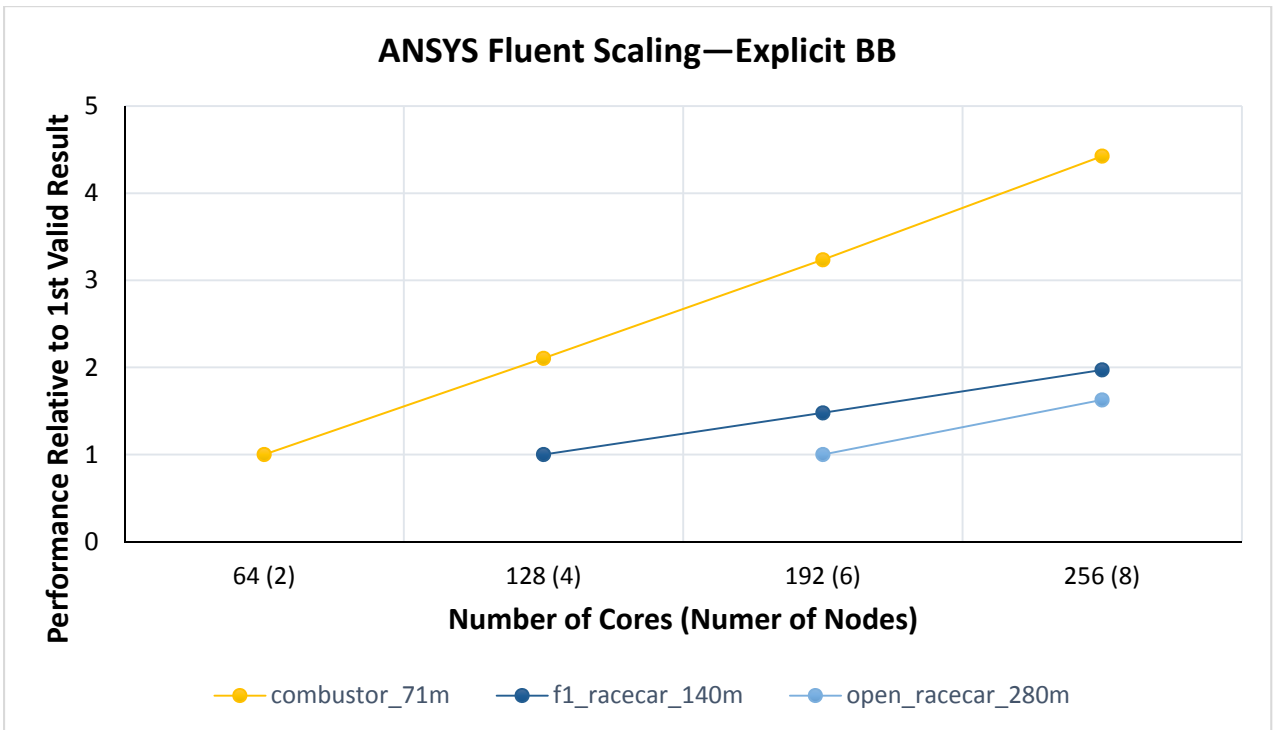
Figure 11 ANSYS Fluent Scaling—Explicit BB (4)

4.4 ANSYS Mechanical

ANSYS Mechanical is a multi-physics Finite Element Analysis (FEA) software commonly used in many engineering disciplines. Depending on specific problem types, FEA applications may or may not scale well across multiple processor cores and servers. Specific types of FEA problems will benefit from GPU acceleration, while other problems may not benefit. Implicit FEA problems often place large demands on the memory and disk I/O sub-systems. Given the varying system requirements for different types of FEA problems, benchmarking for ANSYS Mechanical was performed by using the Implicit, Implicit GPGPU, and Explicit building block systems. The ten benchmark problems from the ANSYS Mechanical v17.0 benchmark suite were evaluated on the reference system.

The performance results for each individual building block types are presented in the following sections. The results are presented by using the Core Solver Rating metric. This metric represents the performance of the solver core which excludes any job pre- and post-processing.

Figure 12 shows the relative performance of the three compute building block types for the ANSYS Mechanical benchmarks by using the DMP solver. For this comparison, all processor cores and GPUs in the individual building blocks are utilized while running ANSYS Mechanical. For the V17sp-3 benchmark, there is a known issue that results in poor performance when using GPU acceleration, so the results for this benchmark are not reported. It is clear from this comparison that there is not a single best performing server configuration for ANSYS Mechanical. The best server configuration for this application is highly dependent on the specific simulation problem.
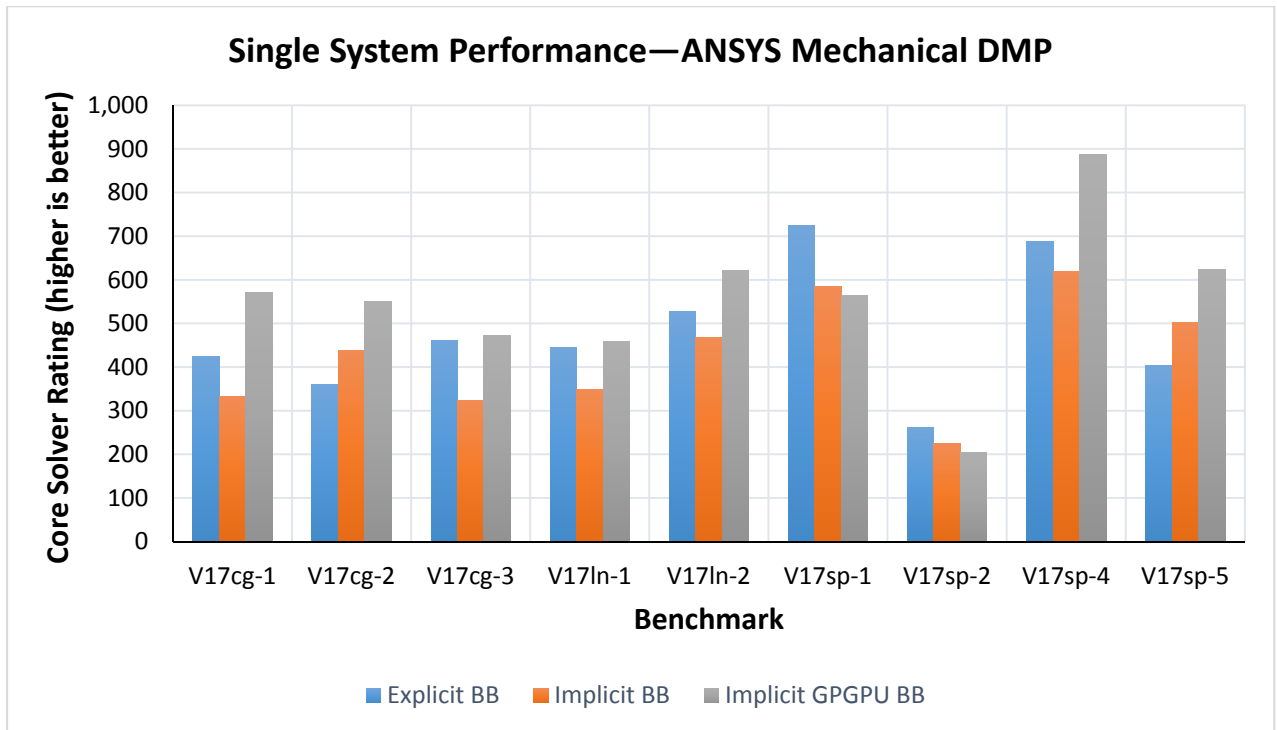

Figure 12 Performance of Individual Building Blocks—ANSYS Mechanical DMP

4.4.1   Implicit Building Block

Two types of solvers are available with ANSYS Mechanical: Distributed Memory Parallel (DMP) and Shared Memory Parallel (SMP). The performance results for these two solvers on an Implicit building block server are shown in Figure 13 and Figure 14. Each data point on the graphs records the performance of the specific benchmark data set by using the number of cores marked on the horizontal axis. In general, the DMP solver offers equivalent or better performance than the SMP solver. The DMP solver continues to scale up to 16 cores, while the SMP solver does not scale as well with more than eight cores.
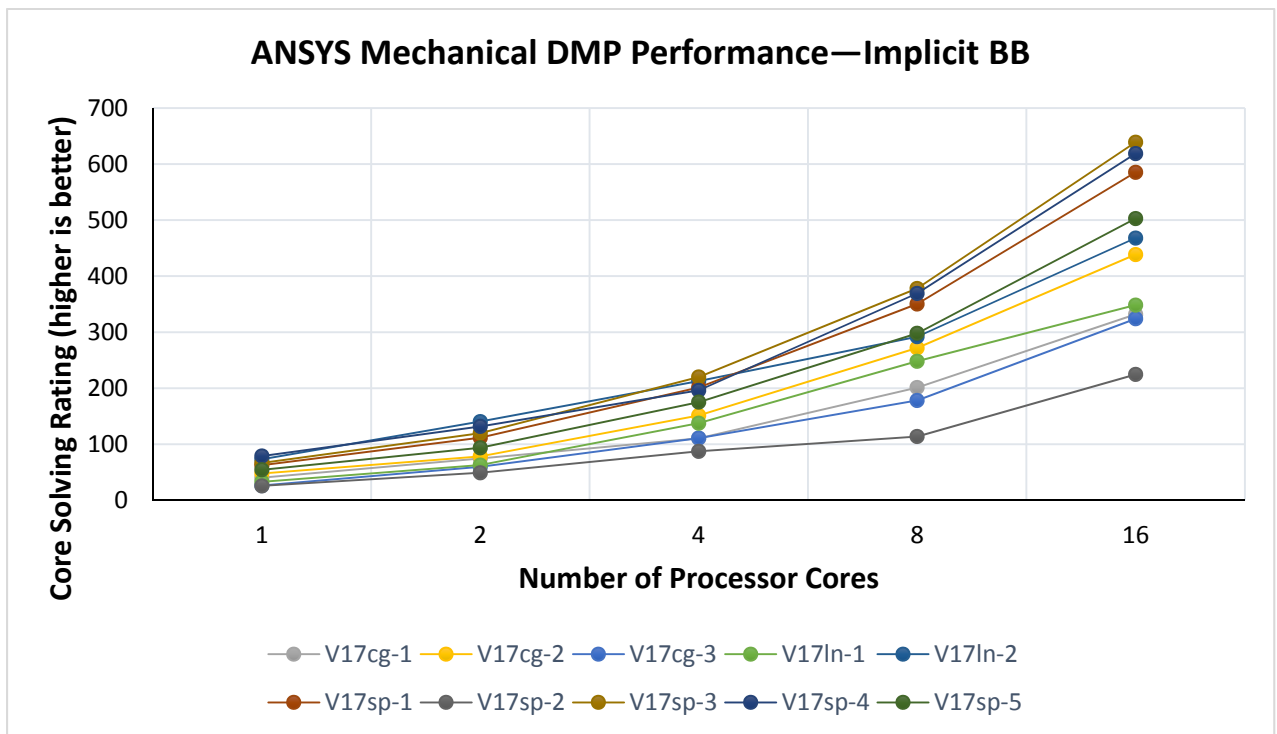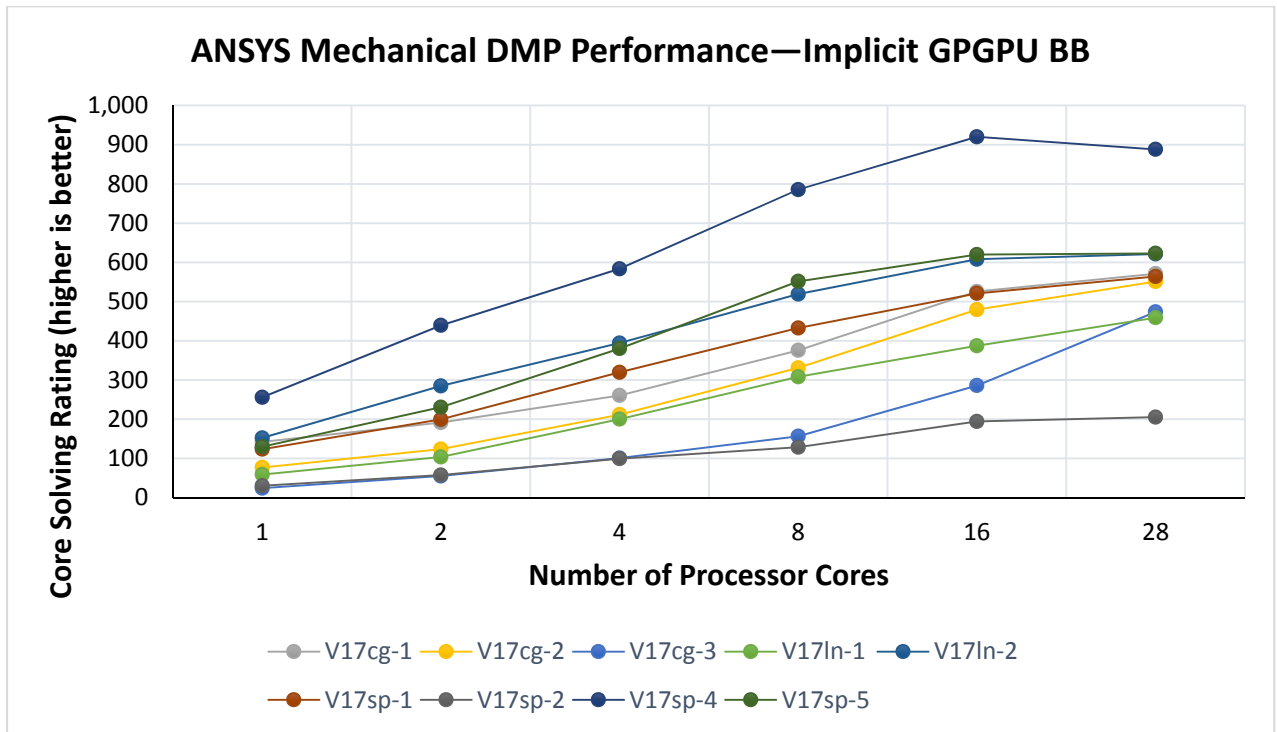

Figure 13 ANSYS Mechanical DMP Performance—Implicit BB

ANSYS Mechanical SMP Performance—Implicit BB
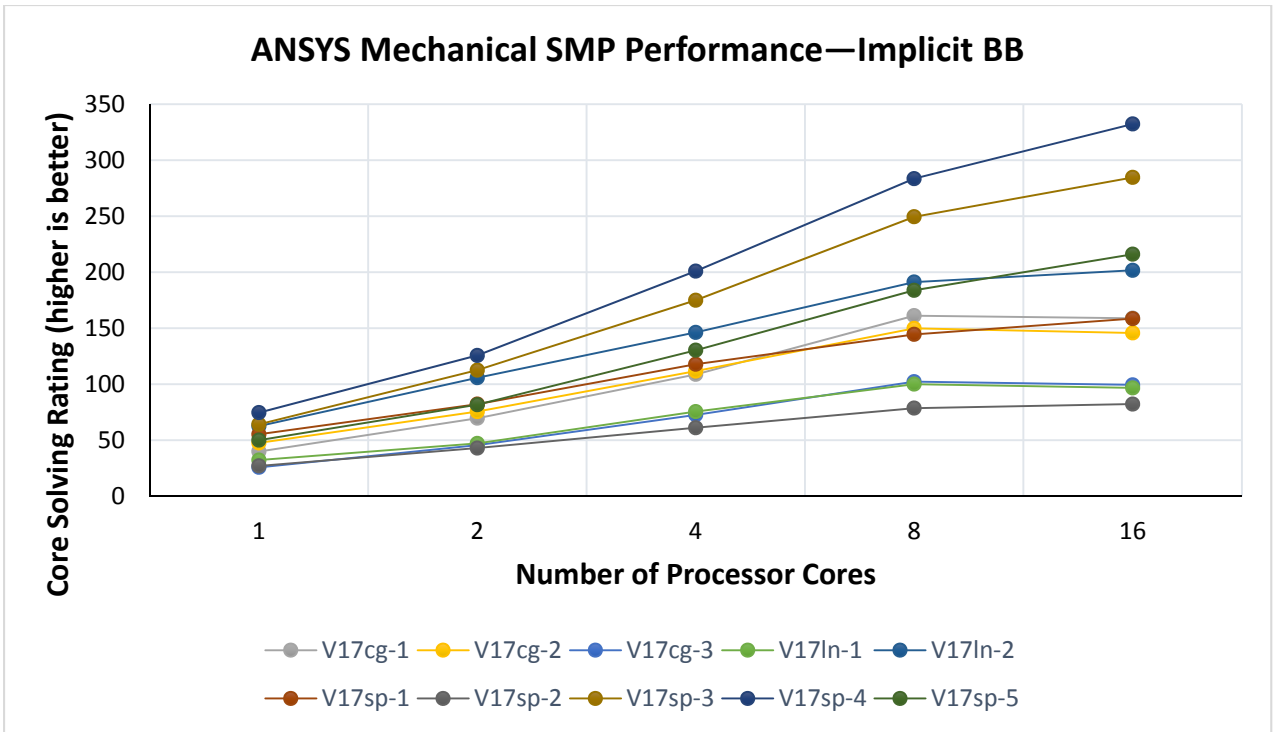
Figure 14 ANSYS Mechanical SMP Performance—Implicit BB

Figure 15 and Figure 16 present the same performance data but plotted relative to the one-core result. This makes it easy to see the scaling of the solution. That is, the performance improvement as more cores are used for the analysis. Problem scalability depends on many factors such as the number of degrees of freedom in the problem and on the particular solution type and solver being used. For the DMP solver with 16 cores, scaling for all of the benchmark problems ranges from 6.4–12.4. For the SMP solver with 16 cores, scaling ranges from 2.9–4.5 across all of the benchmark problems.
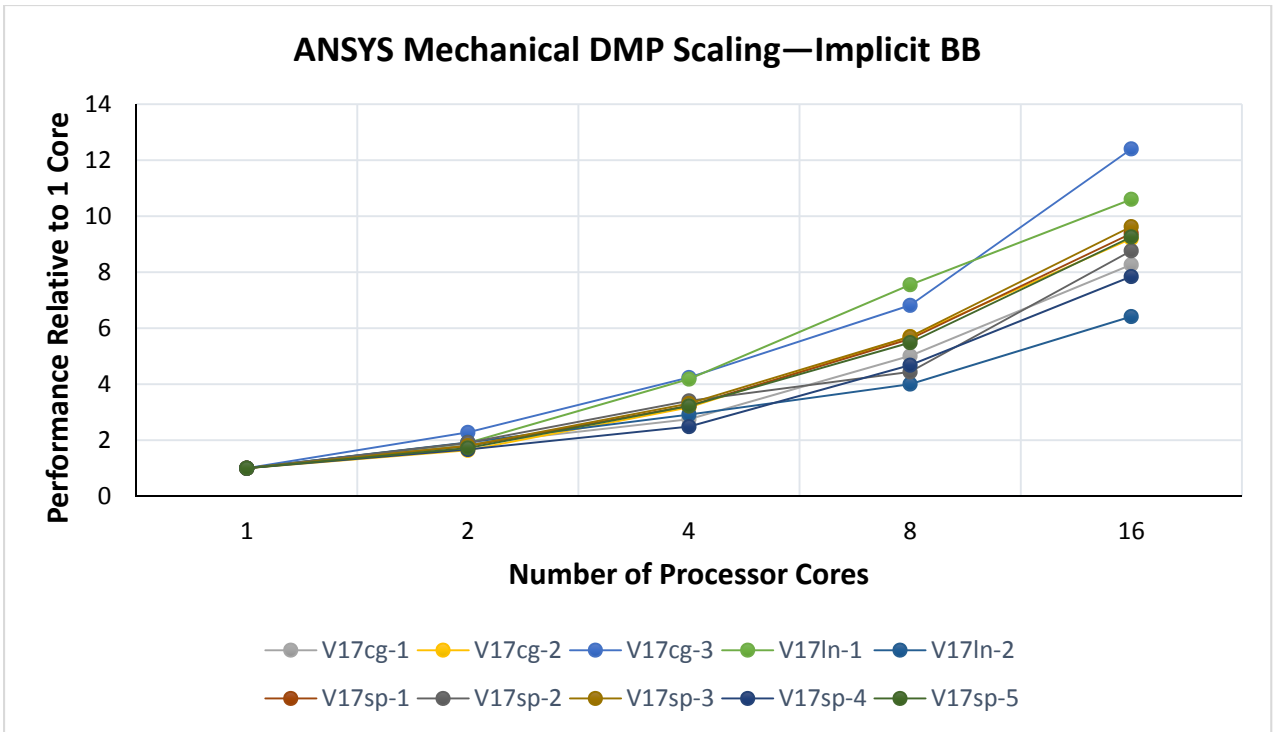
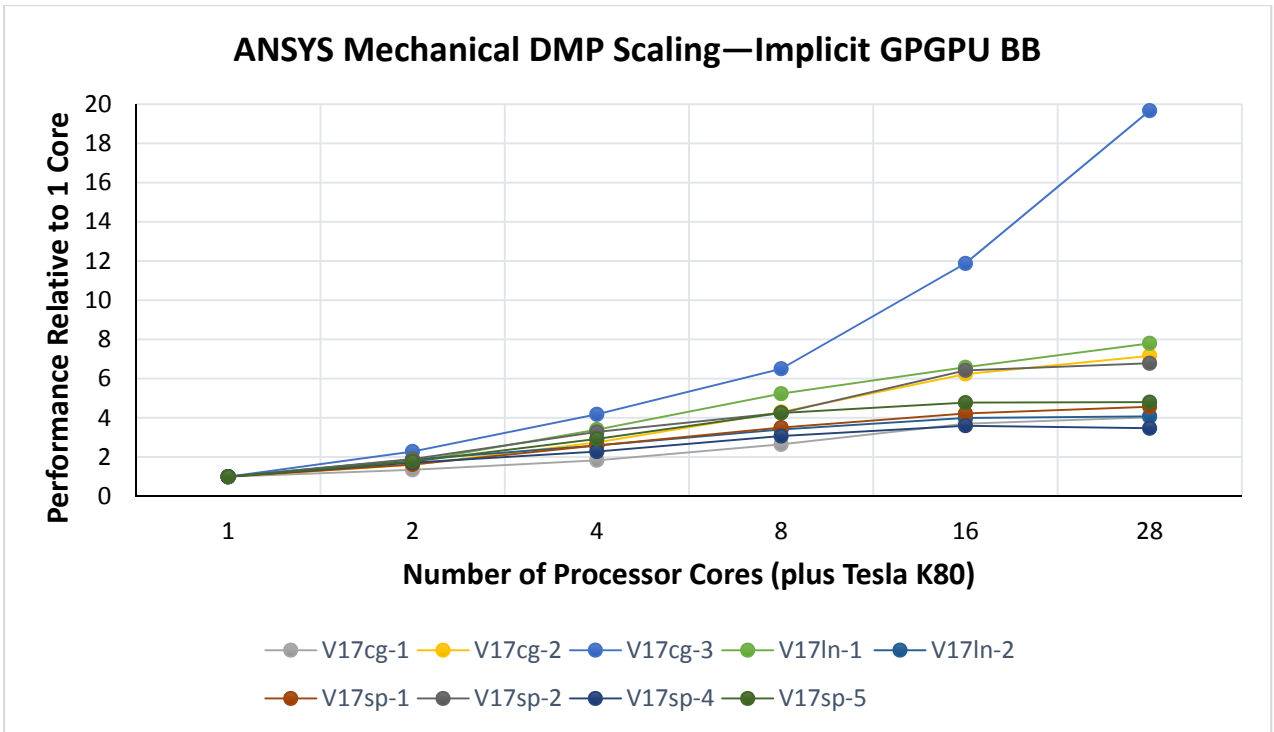Figure 15 ANSYS Mechanical DMP Scaling—Implicit BB
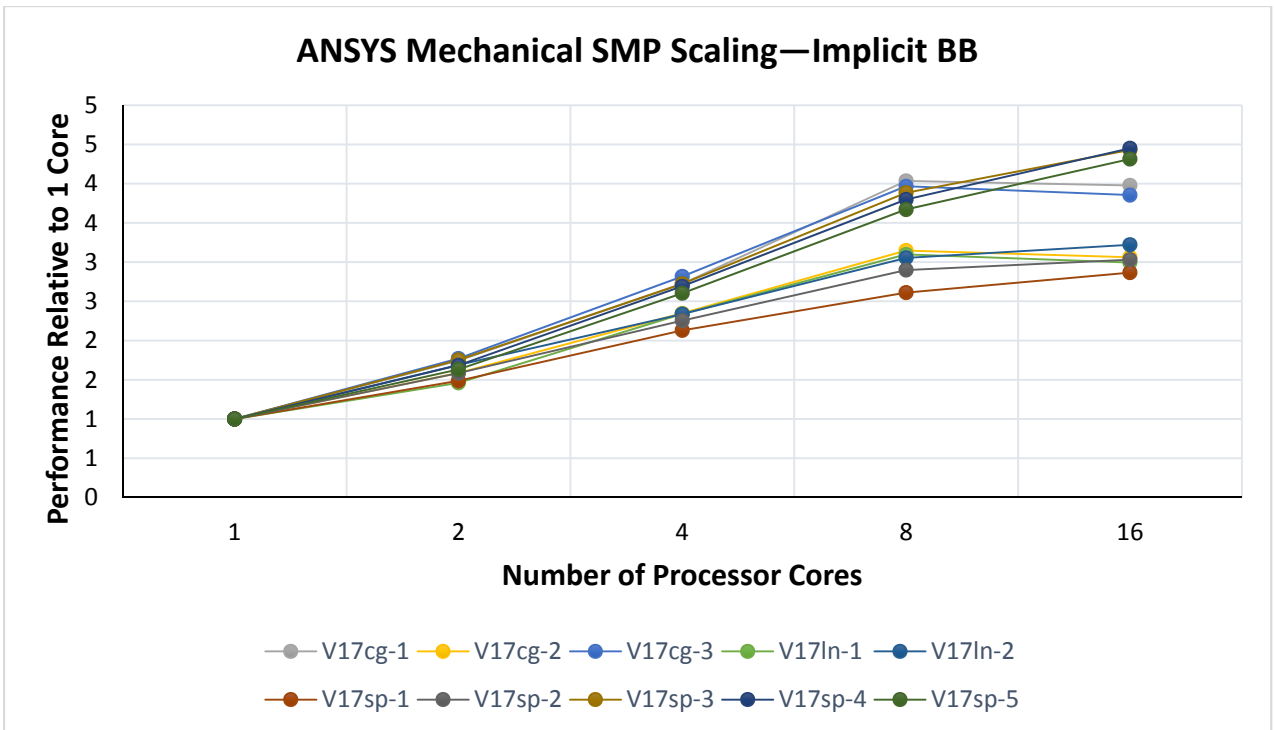
Figure 16 ANSYS Mechanical SMP Scaling—Implicit BB

4.4.2 Implicit GPGPU Building Block

The Implicit GPGPU building block includes a NVIDIA Tesla K80 which contains two GPUs. Both GPUs were used for the ANSYS Mechanical benchmarks. GPU acceleration is available with both DMP and SMP solvers. Therefore, results for both solvers are reported. The performance results for the two solvers on an Implicit GPGPU building block server are shown in Figure 17 and Figure 18. Each data point on the graphs records the performance of the specific benchmark data set by using two GPUs plus the number of processor cores marked on the horizontal axis. In general, the DMP solver offers equivalent or better performance than the SMP solver. The DMP solver continues to scale up to 16 processor cores or more for a few of the benchmarks, while the SMP solver does not scale as well with more than eight cores. For most of the benchmark cases, using GPU acceleration provides a significant performance advantage when compared against the Implicit BB results.

Note that there is a known issue with benchmark case V17sp-3 when using GPU acceleration with the DMP solver. Because of this, the results for the DMP solver for this case are not reported.

Figure 17 ANSYS Mechanical DMP Performance—Implicit GPGPU BB

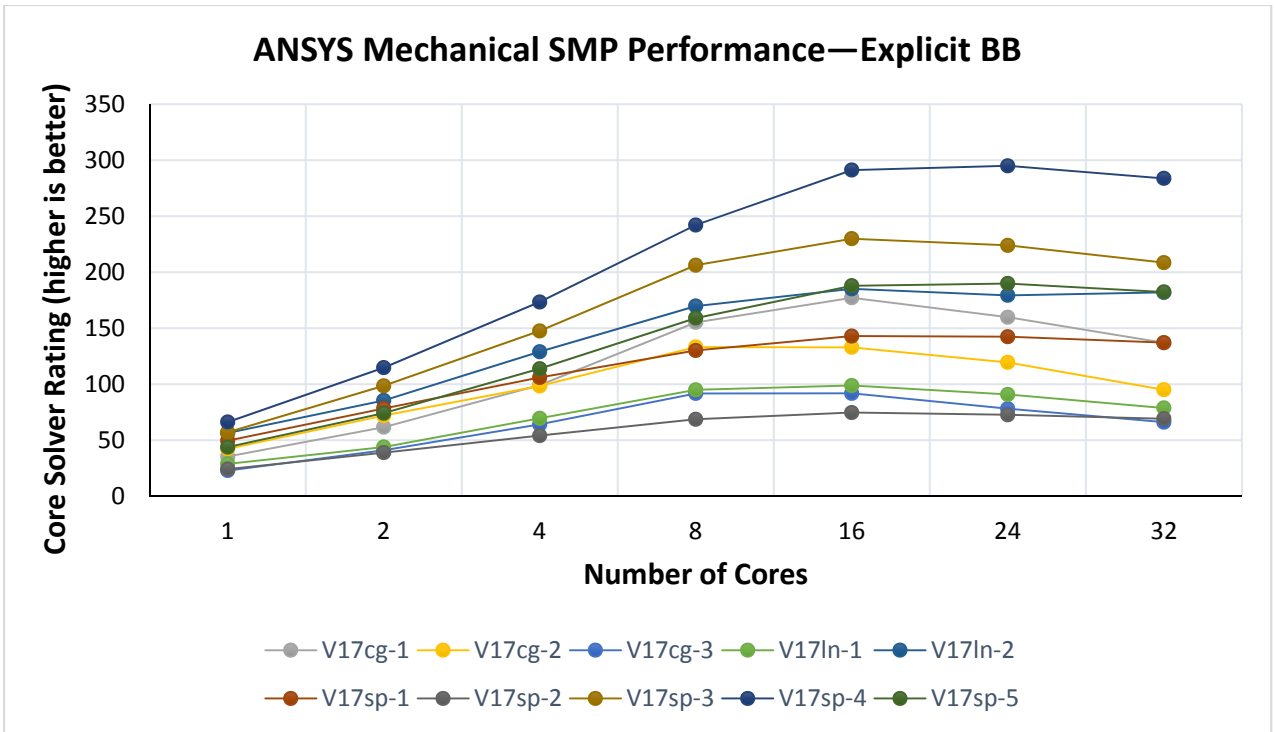**ANSYS Mechanical SMP Performance—Implicit GPGPU BB**
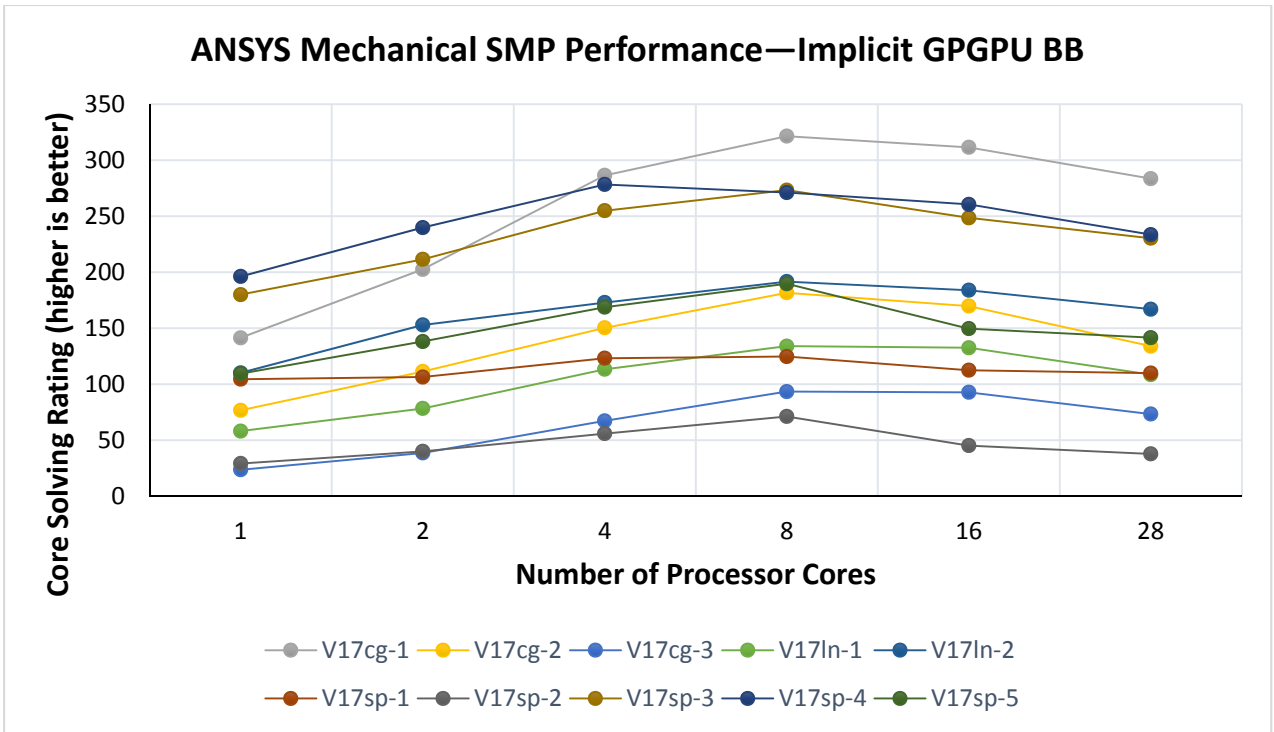
Figure 18 ANSYS Mechanical SMP Performance—Implicit GPGPU BB

Figure 19 and Figure 20 present the same performance data but plotted relative to the one core result. This makes it easy to see the scaling of the solution. That is, the performance improvement as more cores are used for the analysis. Problem scalability depends on many factors such as the number of degrees of freedom in the problem, the particular solution type, and solver being used. For the DMP solver with 28 processor cores and two GPUs, scaling for all of the benchmark problems ranges from 3.5−19.7. For the SMP solver with eight processor cores and two GPUs, scaling ranges from 1.2−3.9 across all the benchmark problems.

Figure 19 ANSYS Mechanical DMP Scaling—Implicit GPGPU BB
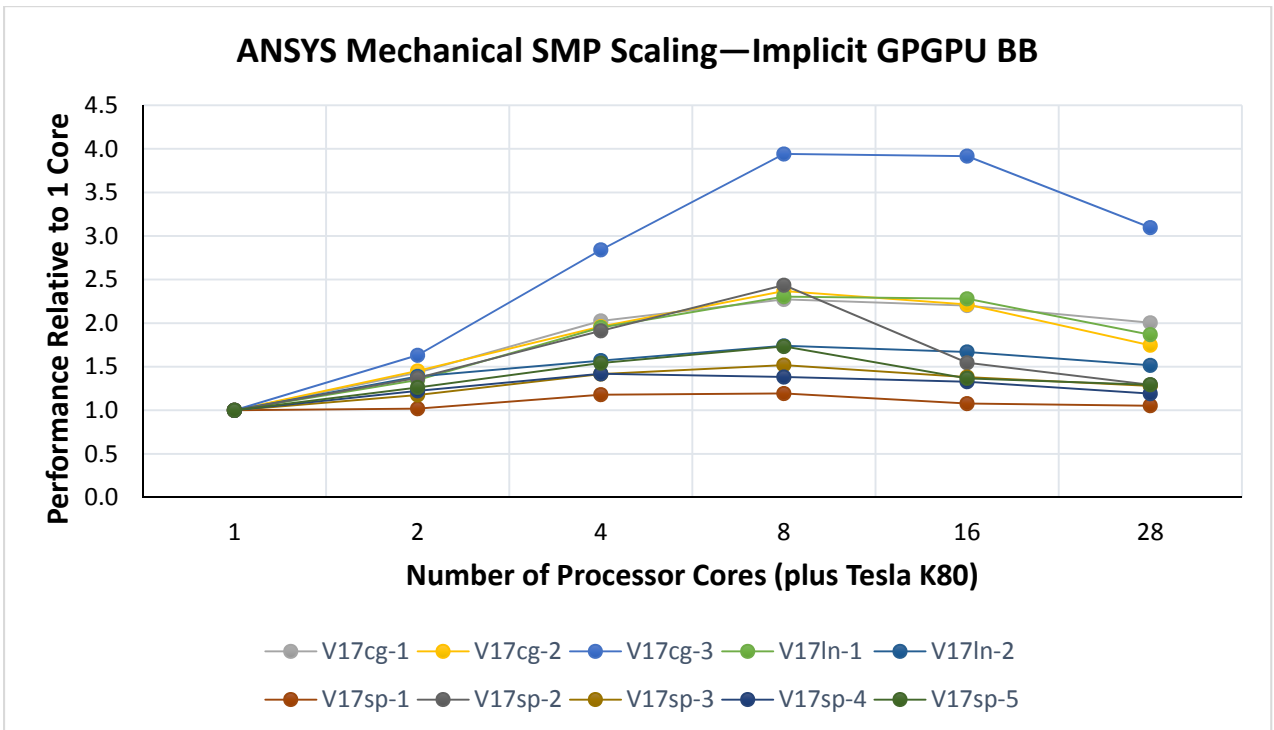
Figure 20 ANSYS Mechanical SMP Scaling—Implicit GPGPU BB

4.4.3   Explicit Building Block

The performance results for the ANSYS Mechanical DMP and SMP solvers on Explicit building blocks are shown in Figure 21 and Figure 22. For this series of benchmarks, the DMP solver was run on multiple systems and the SMP solver was run on a single system. Each data point on the graphs records the performance of the specific benchmark data set by using the number of processor cores marked on the horizontal axis. The DMP solver continues to scale up to 256 processor cores for some of the benchmarks. The SMP solver does not scale as well when more than eight cores are used.
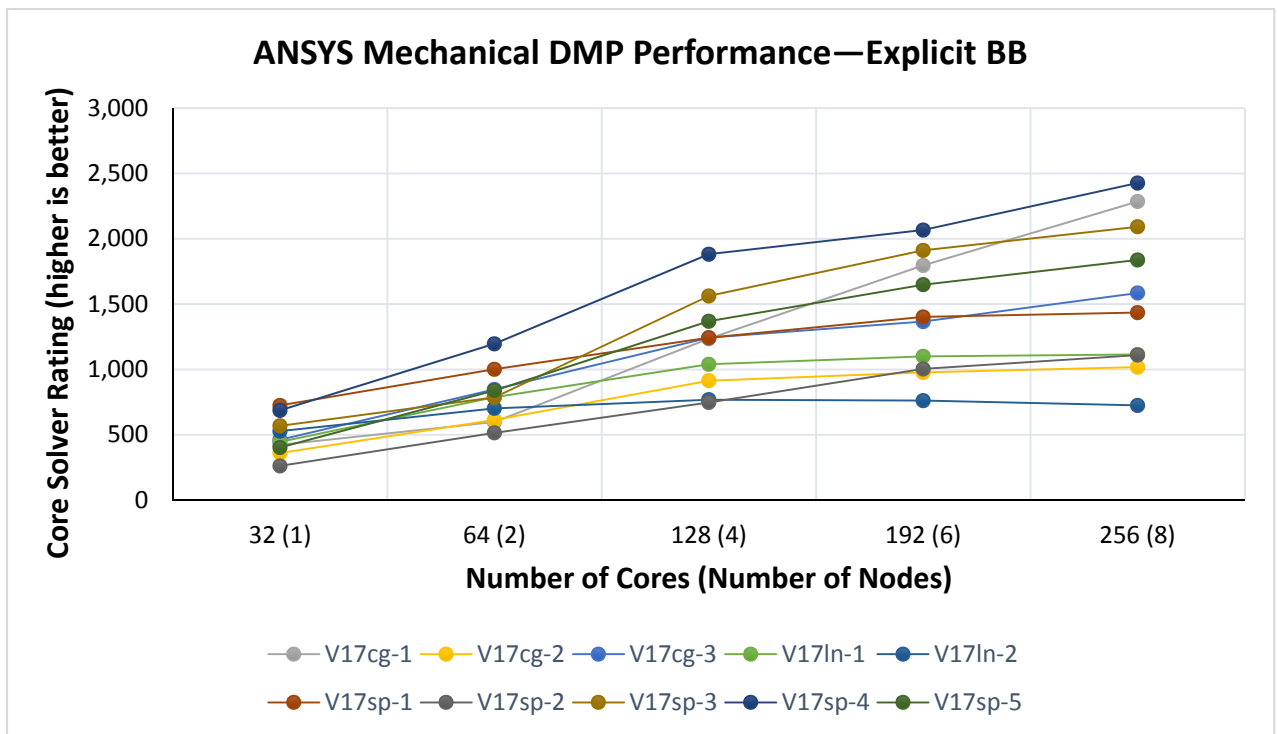
Figure 21 ANSYS Mechanical DMP Performance—Explicit BB

Figure 22 ANSYS Mechanical SMP Performance—Explicit BB

Figure 23 and Figure 24 present the same performance data but plotted relative to the one-node or one-core result. This makes it easy to see the scaling of the solution—the performance improvement when more cores are used for the analysis. Problem scalability depends on many factors including the number of degrees of freedom in the problem and on the particular solution type and solver being used. For the DMP solver with 256 cores, scaling for all of the benchmark problems ranges from 1.4–5.4. For the SMP solver with 16 processor cores, scaling ranges from 2.9–5.0 across all the benchmark problems.
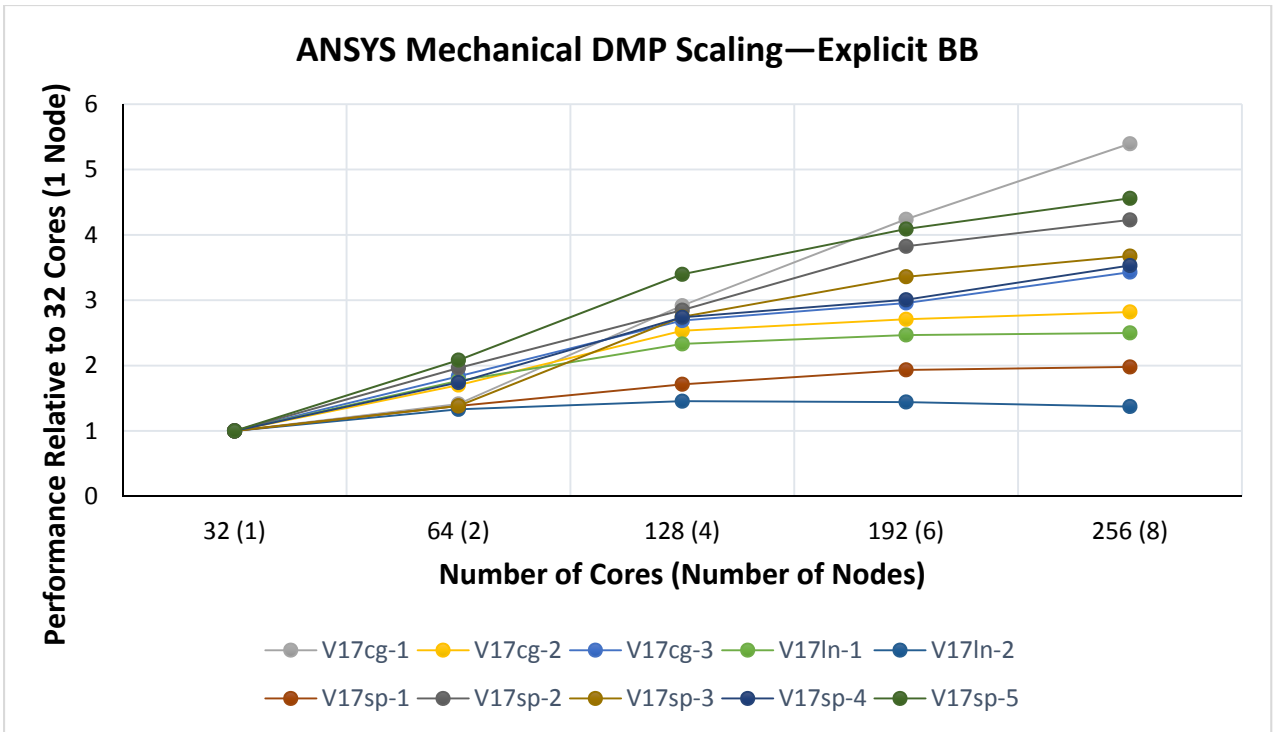
Figure 23 ANSYS Mechanical DMP Scaling—Explicit BB
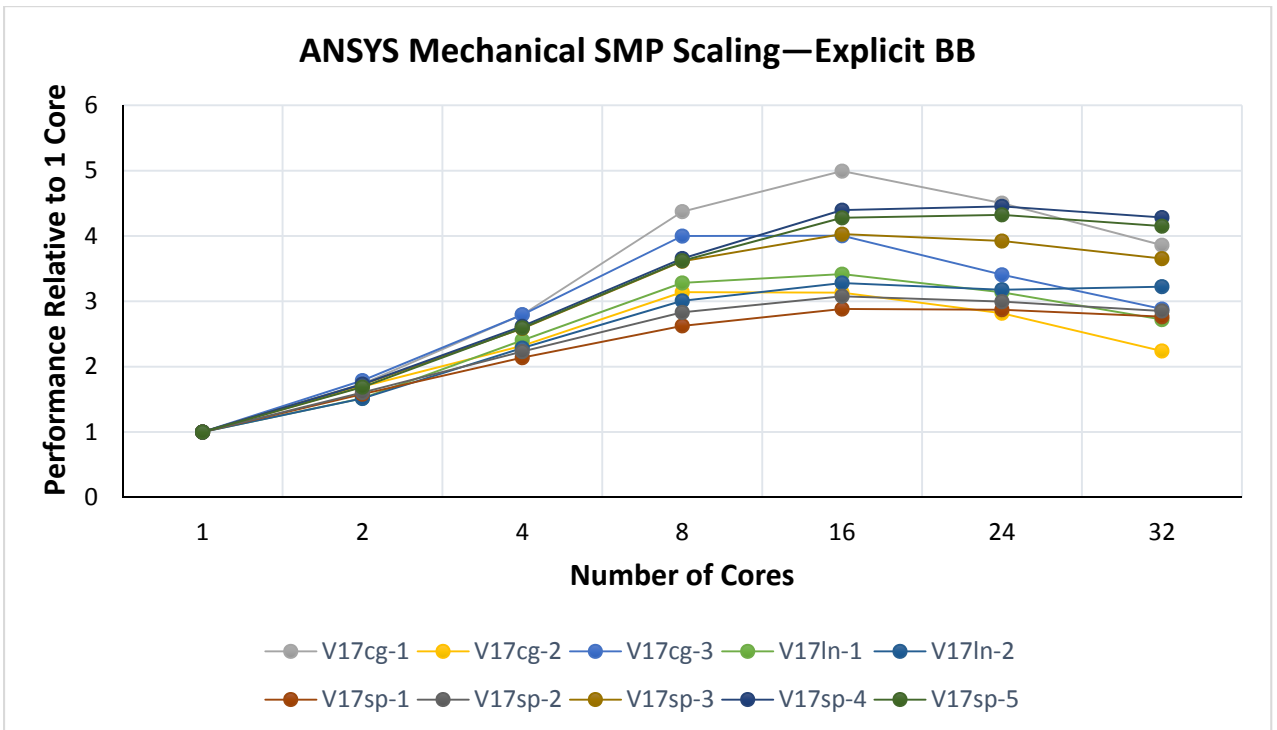
Figure 24 ANSYS Mechanical SMP Scaling—Explicit BB

4.5 CD-adapco STAR-CCM+

CD-adapco, a Siemens Business, produces STAR-CCM+ software. STAR-CCM+ is used in many engineering disciplines to simulate a wide range of physics. STAR-CCM+ is often used for Computational Fluid Dynamics (CFD), and to simulate heat transfer, chemical reaction, combustion, solid transport, acoustics, and Fluid Structure Interaction (FSI).

CFD applications typically scale well across multiple processor cores and servers, use moderate amounts of memory per core, and perform minimal disk I/O while solving. For applications with these characteristics, the Explicit building block servers are appropriate.

Fifteen benchmark problems from the STAR-CCM+ benchmark suite were evaluated on the EBB servers in the reference system.

The results for STAR-CCM+ are presented by using the Average Elapsed Time metric which is the number of elapsed seconds per solver iteration. A lower value represents better performance.

Figure 25 shows the relative performance of the three compute building block types for eight of the STAR-CCM+ benchmarks. For this comparison, all processor cores in the individual building blocks are utilized while running STAR-CCM+; GPUs are not used. This comparison demonstrates that for STAR-CCM+, application performance is primarily determined by processor performance. The Intel Xeon E5-2697A v4 processor used in the Explicit BB is a good choice for STAR-CCM+. However, the Intel Xeon E5-2680 v4 processor used in the Implicit GPGPU BB also provides good performance.
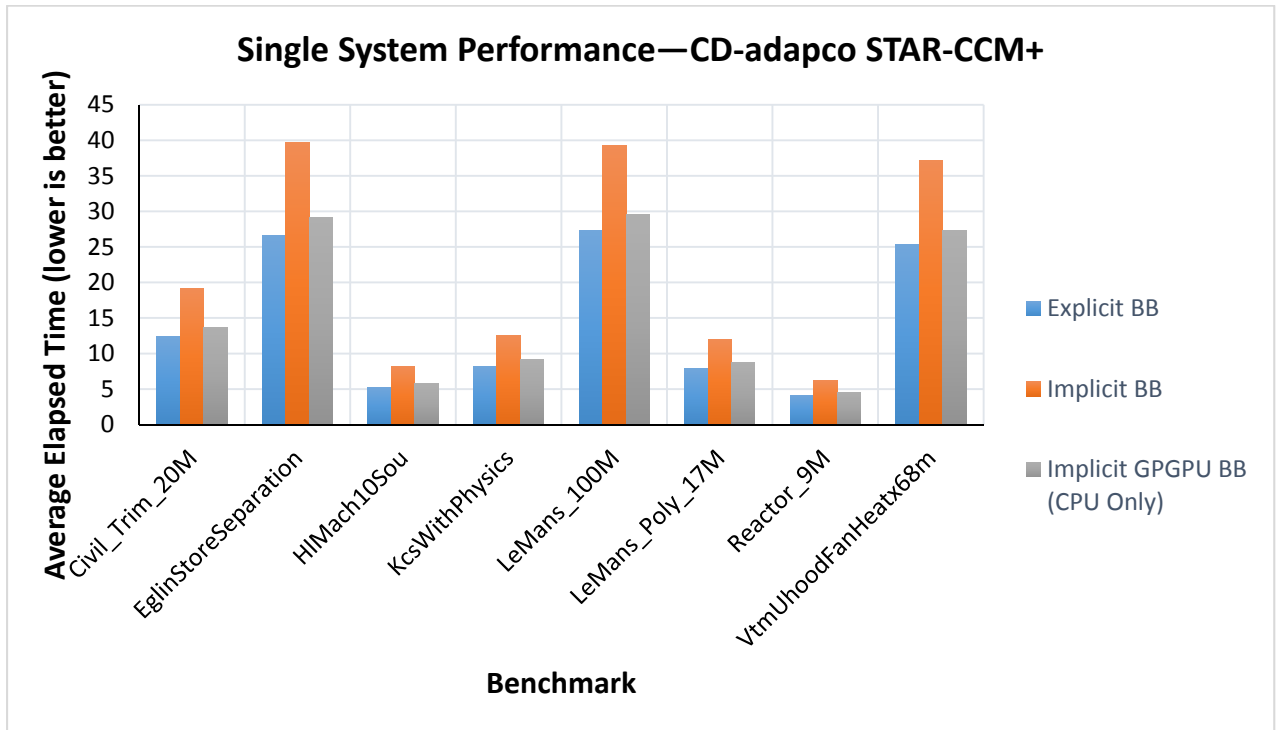

Figure 25 Performance of Individual Building Blocks—CD-adapco STAR-CCM+

The graphs in Figure 26 through Figure 29 show the measured performance of the reference system, on one to eight EBBs, by using 32–256 cores. Each data point on the graphs records the performance of the specific benchmark data set by using the number of cores marked on the horizontal axis in a parallel simulation. The results are divided into four charts for easy readability because some models run much faster than others depending on the number of cells in the model, type of solver used, and physics of the problem.

LeMans_100M_Coupled, SlidingMorphingNopostHelicopter, vtmBenchmark_178M and LeMans_514M_Coupled require two or more servers for sufficient memory capacity. The results for these cases start with the first valid result obtained for each benchmark problem.
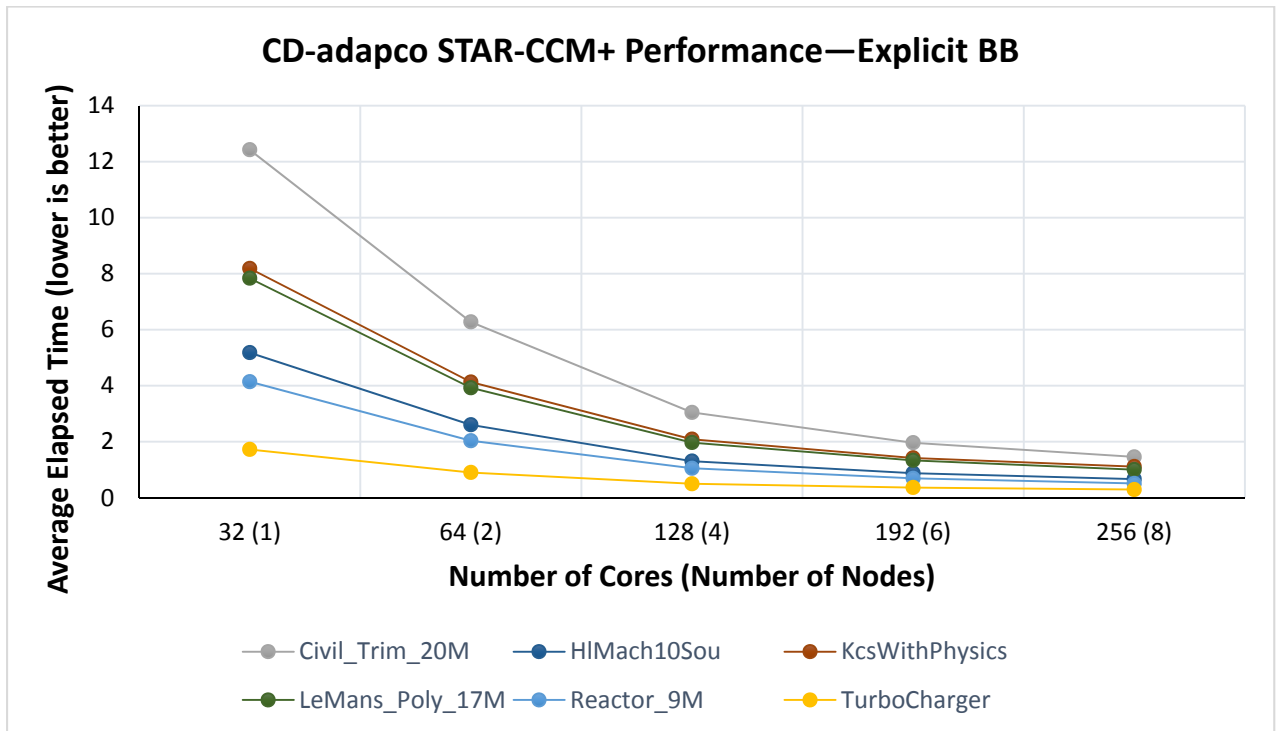
Figure 26 CD-adapco STAR-CCM+ Performance—Explicit BB (1)
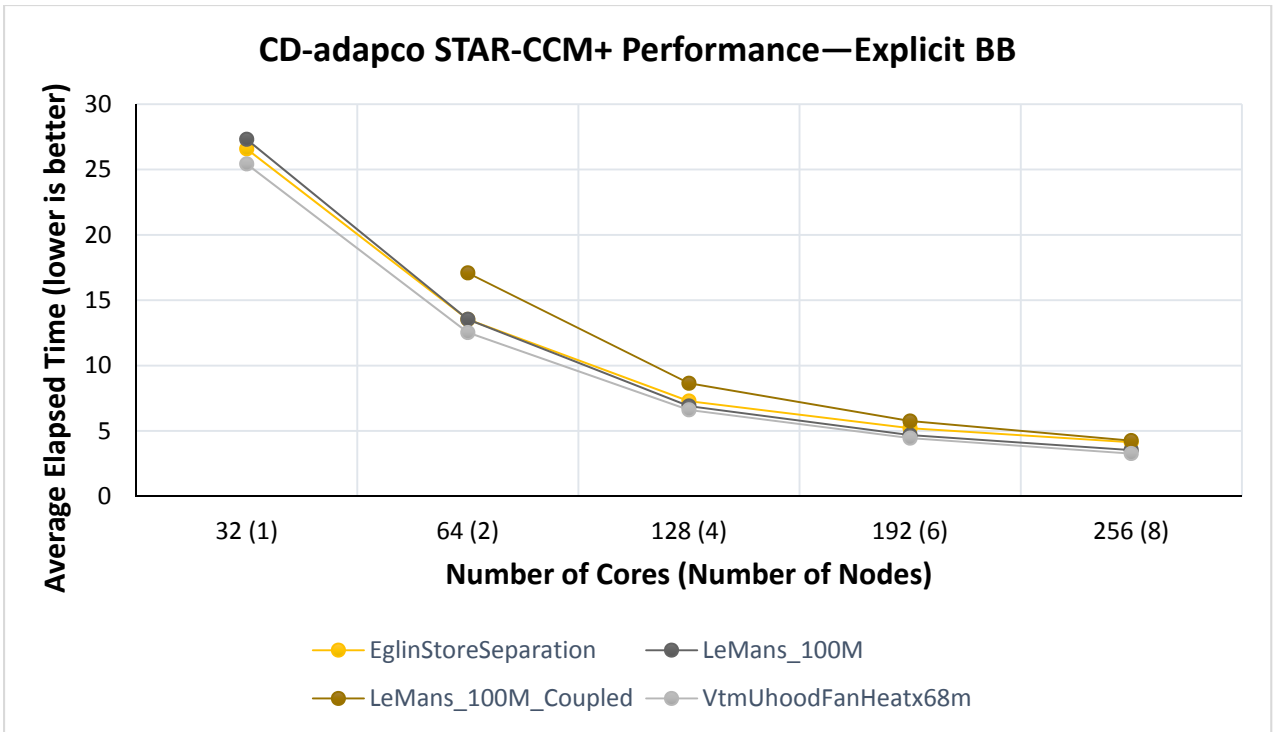
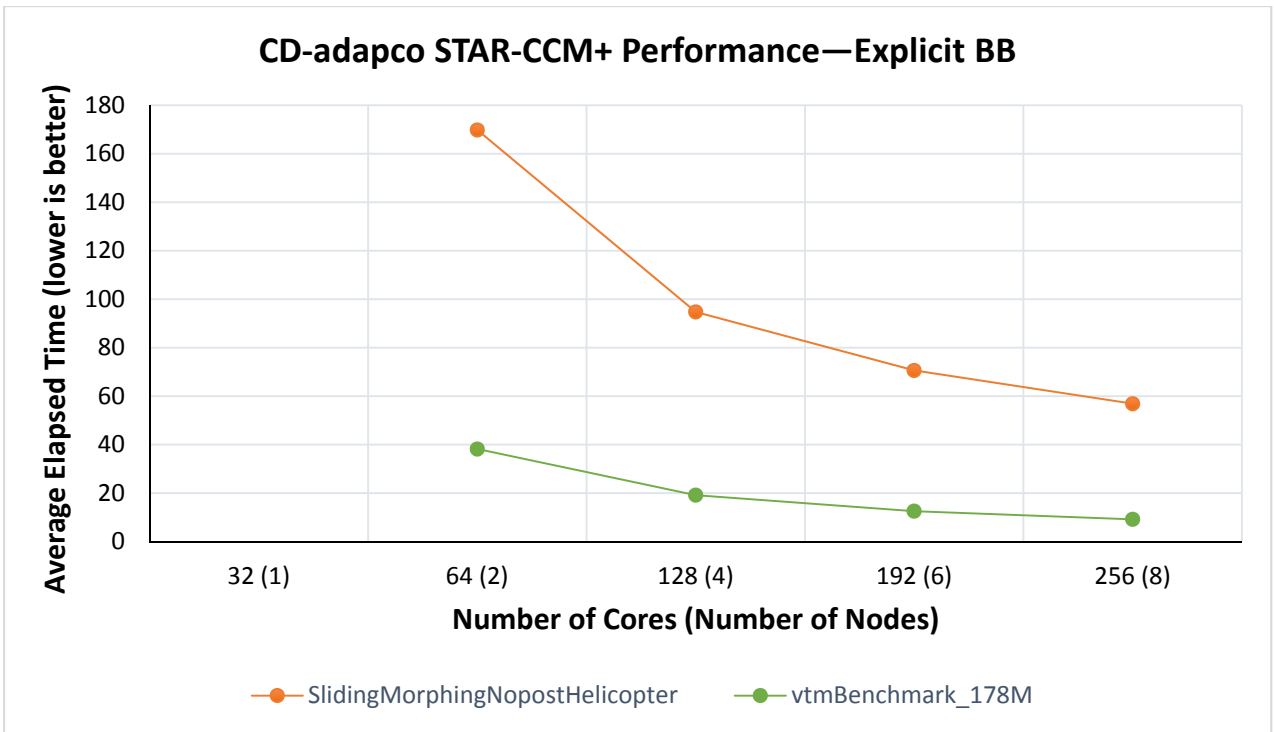Figure 27 CD-adapco STAR-CCM+ Performance—Explicit BB (2)

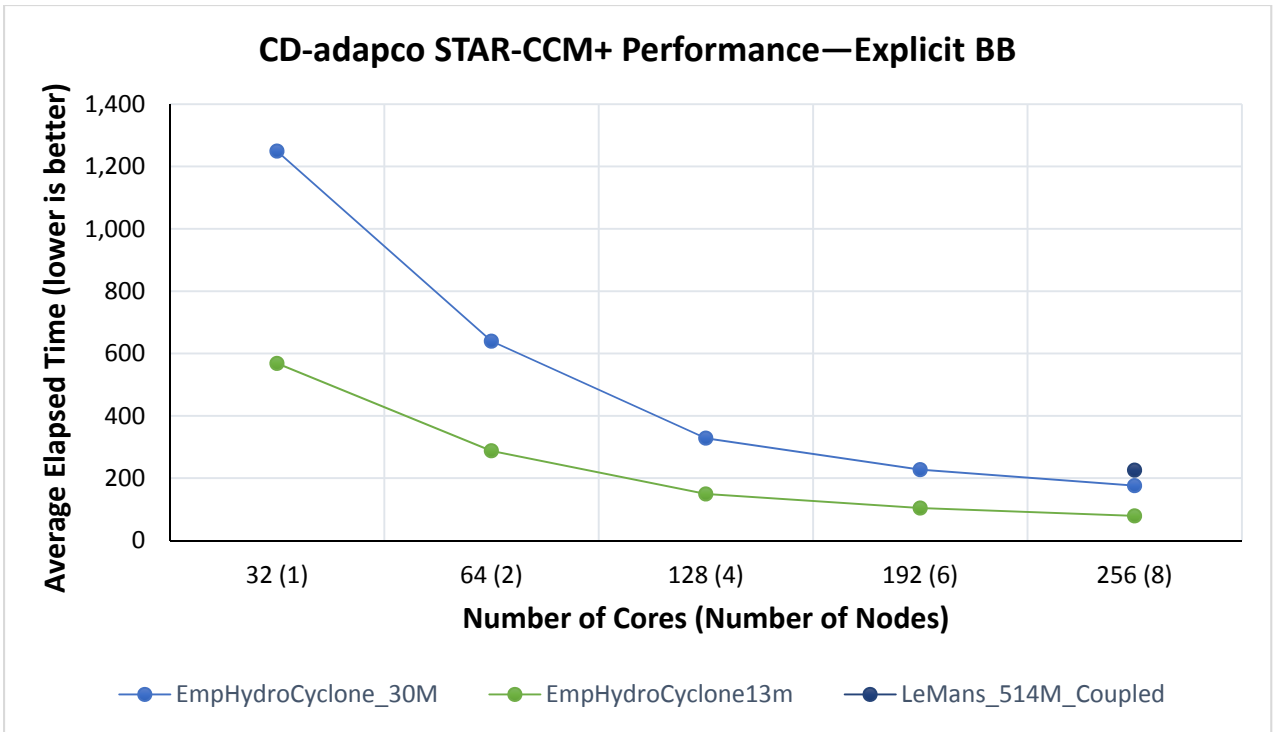Figure 28 CD-adapco STAR-CCM+ Performance—Explicit BB (3)

CD-adapco STAR-CCM+ Performance—Explicit BB

Figure 29 CD-adapco STAR-CCM+ Performance—Explicit BB (4)

Figure 30 through Figure 33 present the same performance data but plotted relative to the "32-cores (1 Node)" result or "64-cores (2 Nodes)" result for problems that require two servers to run. This makes it easy to see the scaling of the solution—the performance improvement when more cores are used for the analysis. Problem scalability depends on the cell count and physics of each problem. For many of the problems, scaling is nearly linear, with relative performance values near eight with 256 cores or eight nodes.
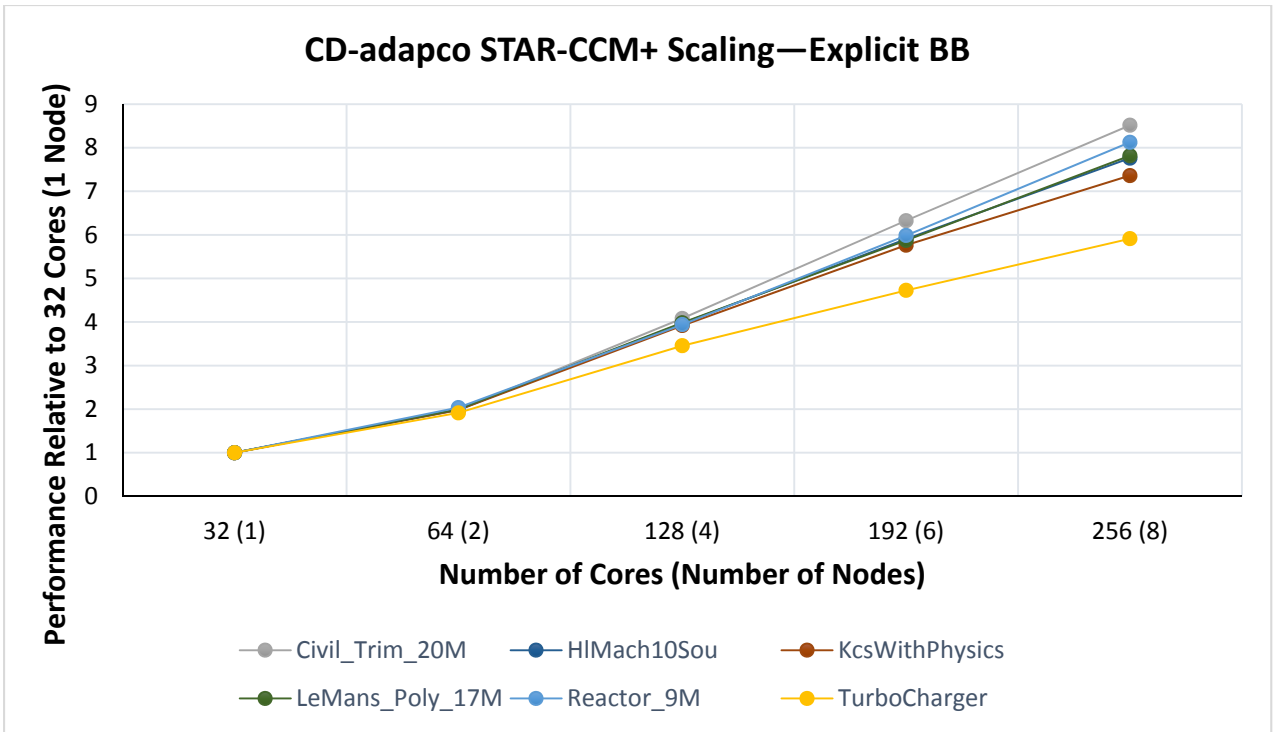
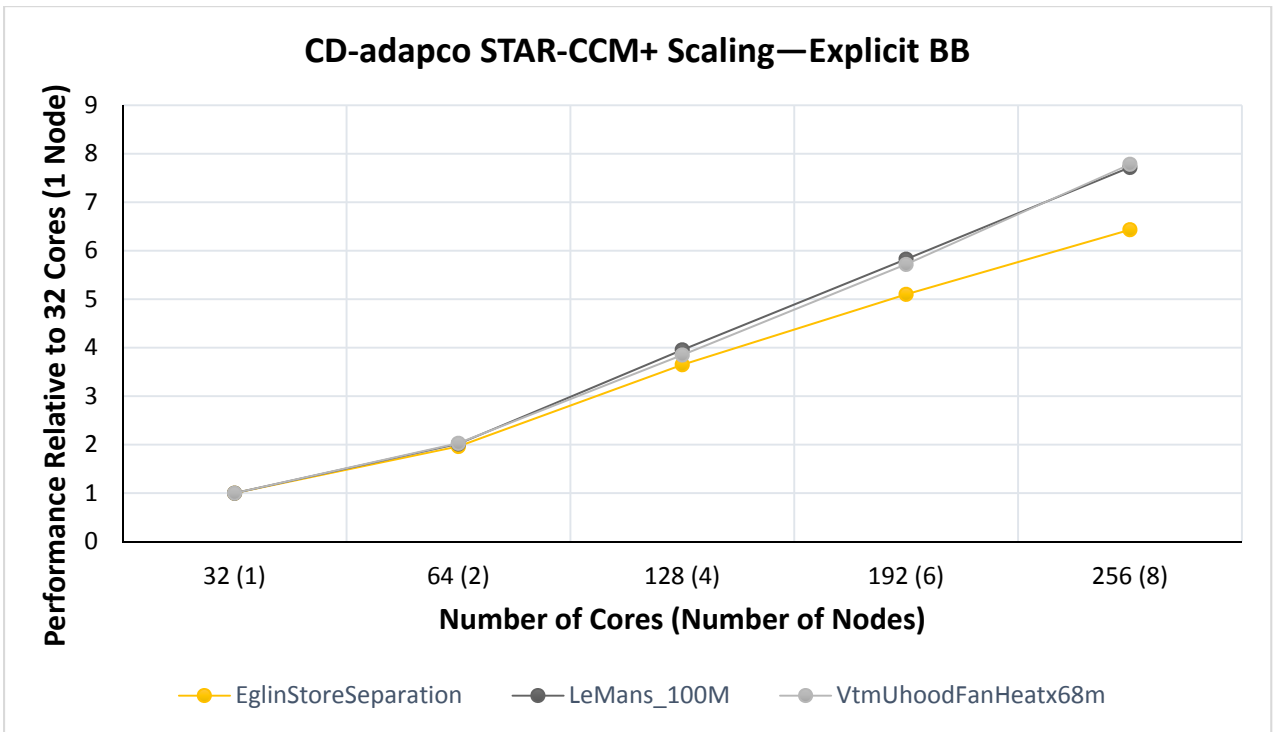Figure 30 CD-adapco STAR-CCM+ Scaling—Explicit BB (1)

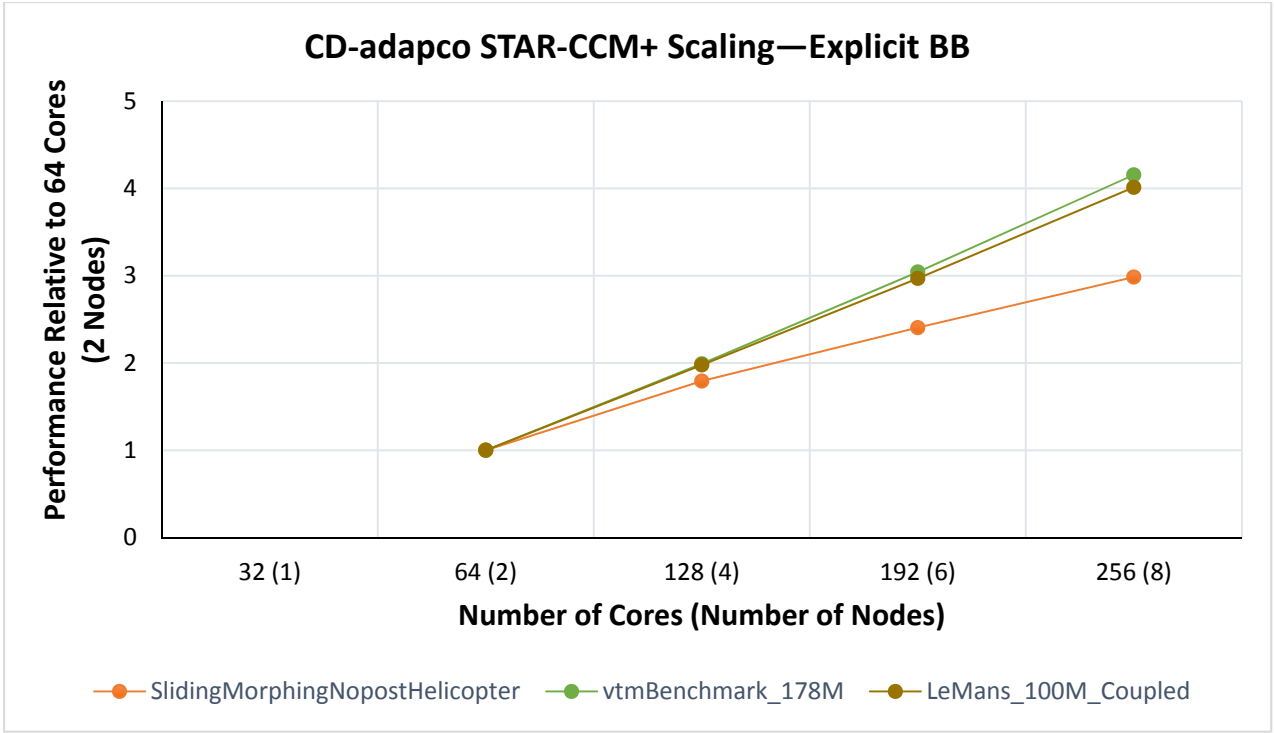Figure 31 CD-adapco STAR-CCM+ Scaling—Explicit BB (2)

Figure 32 CD-adapco STAR-CCM+ Scaling—Explicit BB (3)
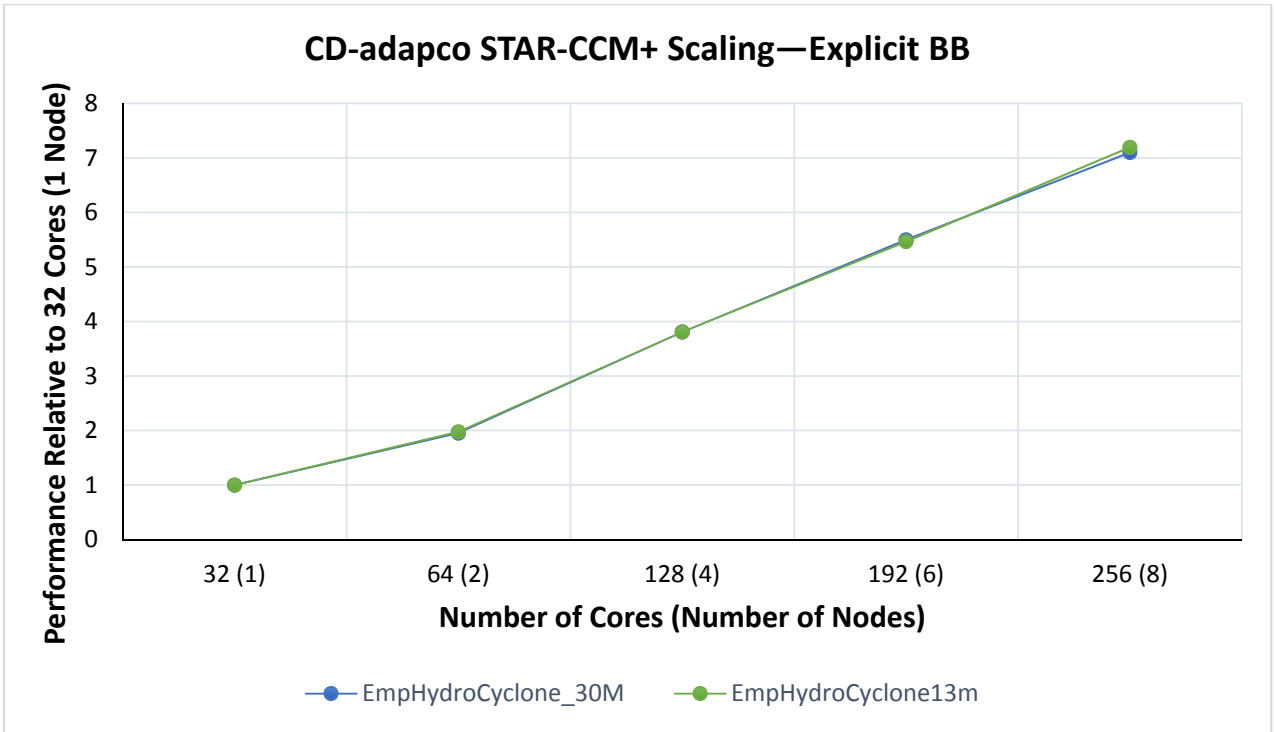

Figure 33 CD-adapco STAR-CCM+ Scaling—Explicit BB (4)

4.6 LSTC LS-DYNA

LSTC LS-DYNA is a multi-physics Finite Element Analysis (FEA) software commonly used in multiple engineering disciplines. Depending on the specific problem types, FEA applications may or may not scale well across multiple processor cores and servers. The two benchmark problems presented here use the LS-DYNA explicit solver, which typically scales much more efficiently than the implicit solver. In the subsequent sections, data is presented for the Explicit Building Block systems because the characteristics of the LS-DYNA explicit solver are most appropriate for this building block type.

The performance results for LS-DYNA are presented by using the Elapsed Time metric. This metric is the total elapsed time for the simulation in seconds as reported by LS-DYNA. A lower value represents better performance.

Figure 34 shows the relative performance of the three compute building block types for the LS-DYNA Car2Car benchmark. For this comparison, all processor cores in the individual building blocks are utilized while running LS-DYNA; GPUs are not used. This comparison demonstrates that for the LS-DYNA explicit solver, application performance is primarily determined by processor performance. The Intel Xeon E5-2697A v4 processor used in the Explicit BB is a good choice for LS-DYNA.

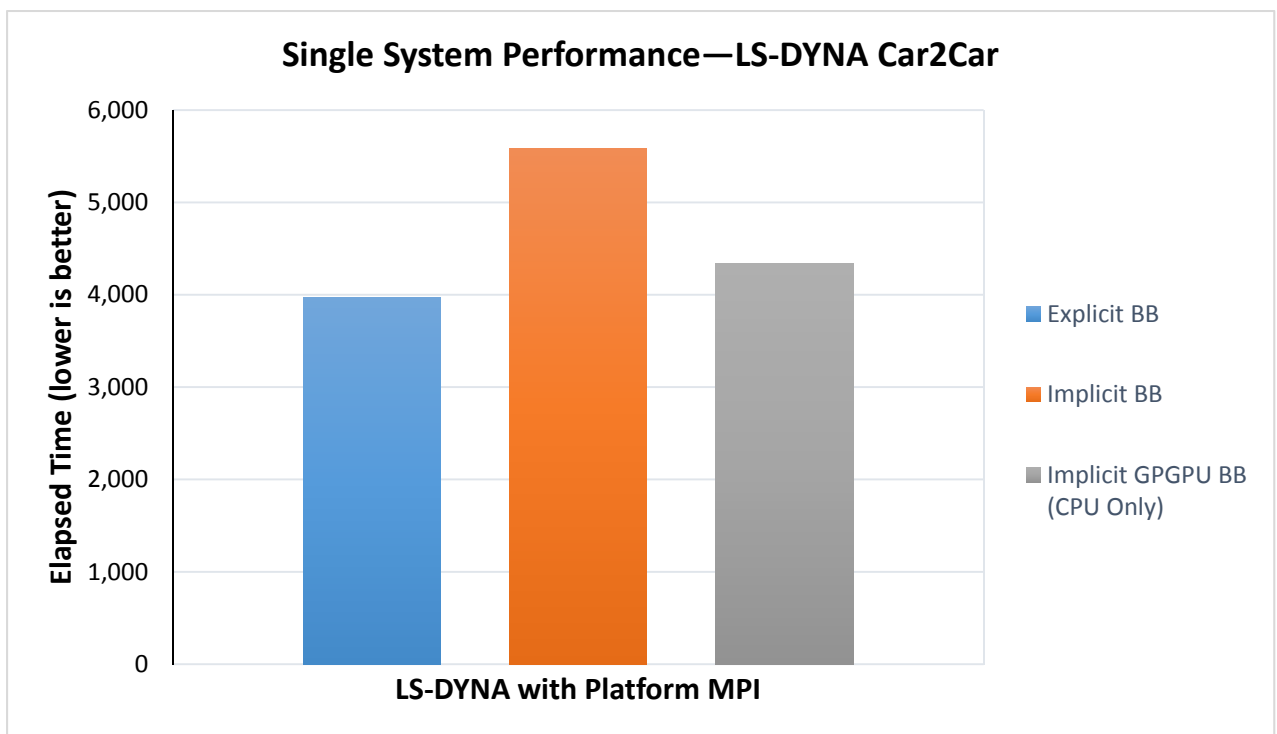**Single System Performance—LS-DYNA Car2Car**

Figure 34 Performance of Individual Building Blocks—LSTC LS-DYNA

4.6.1   Car2Car

The car2car benchmark is a simulation of a two vehicle collision. This benchmark model contains 2.4 million elements, which is relatively small compared to current automotive industry usage. Figure 35 shows the measured performance of the reference system for the car2car benchmark, on one to eight EBBs, using 32 to 256 cores. Each data point on the graph records the performance using the number of cores marked on the horizontal axis in a parallel simulation.

Figure 36 presents the same performance data plotted relative to the "32-cores (1 Node)" result. This makes it easy to see the scaling of the solution—the performance improvement when more cores are used for the analysis. Problem scalability for LS-DYNA depends on the number of elements and other features being simulated in the problem such as contact or airbags. The Car2Car benchmark is a relatively small problem, but scaling is reasonable up to 256 cores or eight nodes.

This problem was run using both Intel MPI and Platform MPI. The performance obtained from the two MPI implementations was nearly identical.
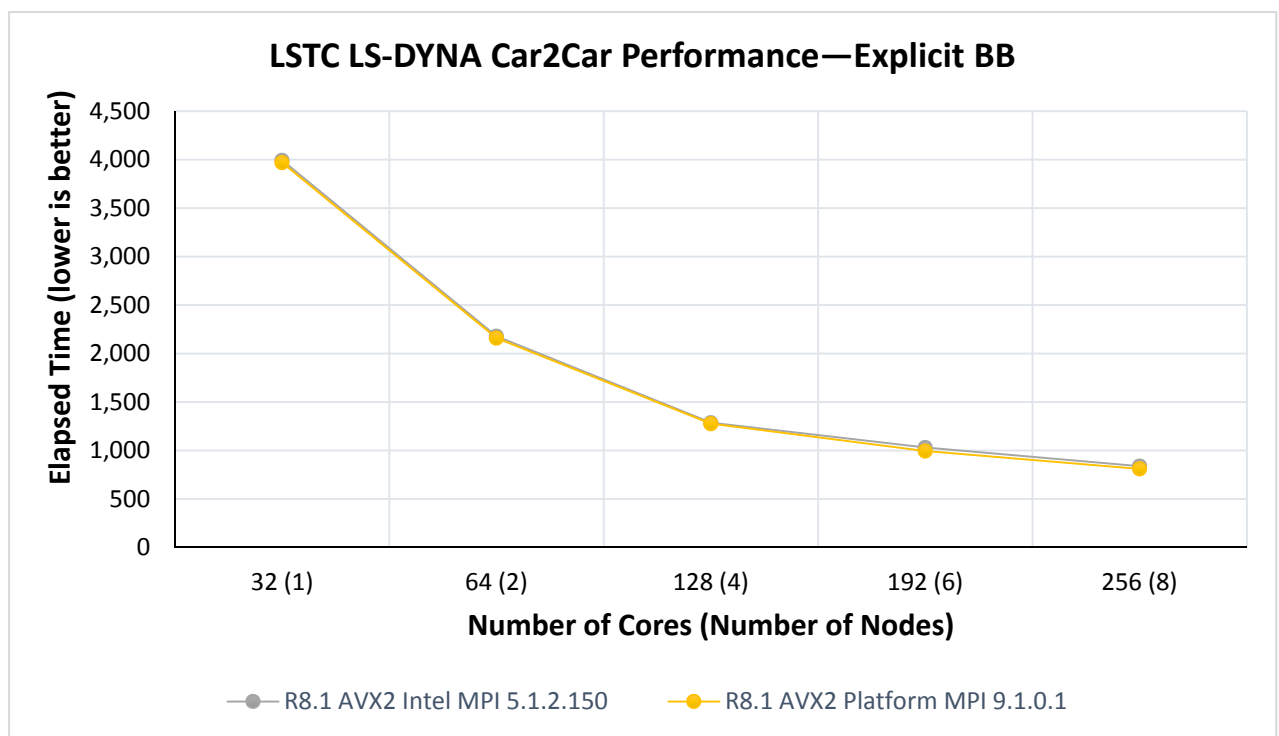

Figure 35 LSTC LS-DYNA Car2Car Performance—Explicit BB
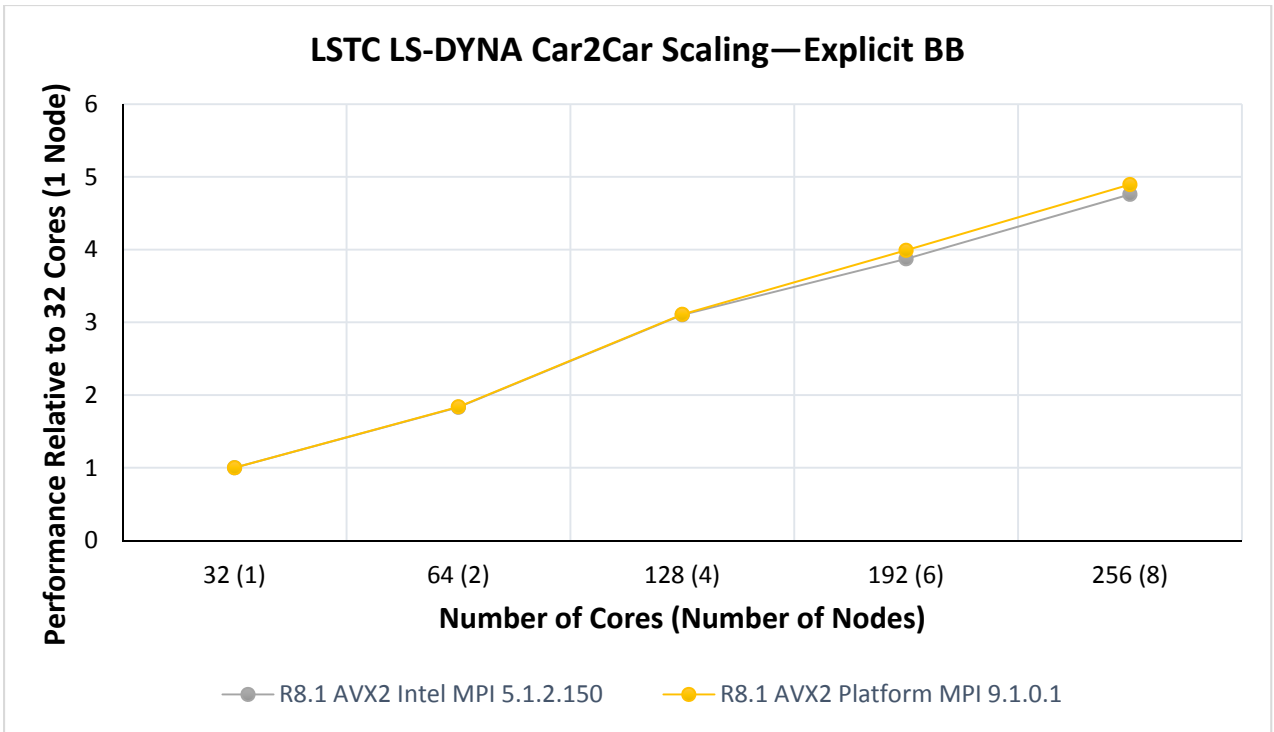
LSTC LS-DYNA Car2Car Scaling—Explicit BB

Figure 36 LSTC LS-DYNA Car2Car Scaling—Explicit BB

4.6.2   ODB-10M

The ODB-10M benchmark is a simulation of a vehicle colliding into an offset deformable barrier. This benchmark model contains 10.6 million elements. Figure 37 shows the measured performance of the reference system for the ODB-10M benchmark, on one to eight EBBs, by using 32 to 256 cores. Each data point on the graph records the performance using the number of cores marked on the horizontal axis in a parallel simulation.

Figure 38 presents the same performance data plotted relative to the "32-cores (1 Node)" result. This makes it easy to see the scaling of the solution—the performance improvement as more cores are used for the analysis. Problem scalability for LS-DYNA depends on the number of elements and other features being simulated in the problem such as contact or airbags. The ODB-10M benchmark scales somewhat better than the Car2Car benchmark, because the ODB-10M benchmark is a larger problem.

This benchmark was run using both Intel MPI and Platform MPI. The performance obtained from the two MPI implementations was nearly identical.
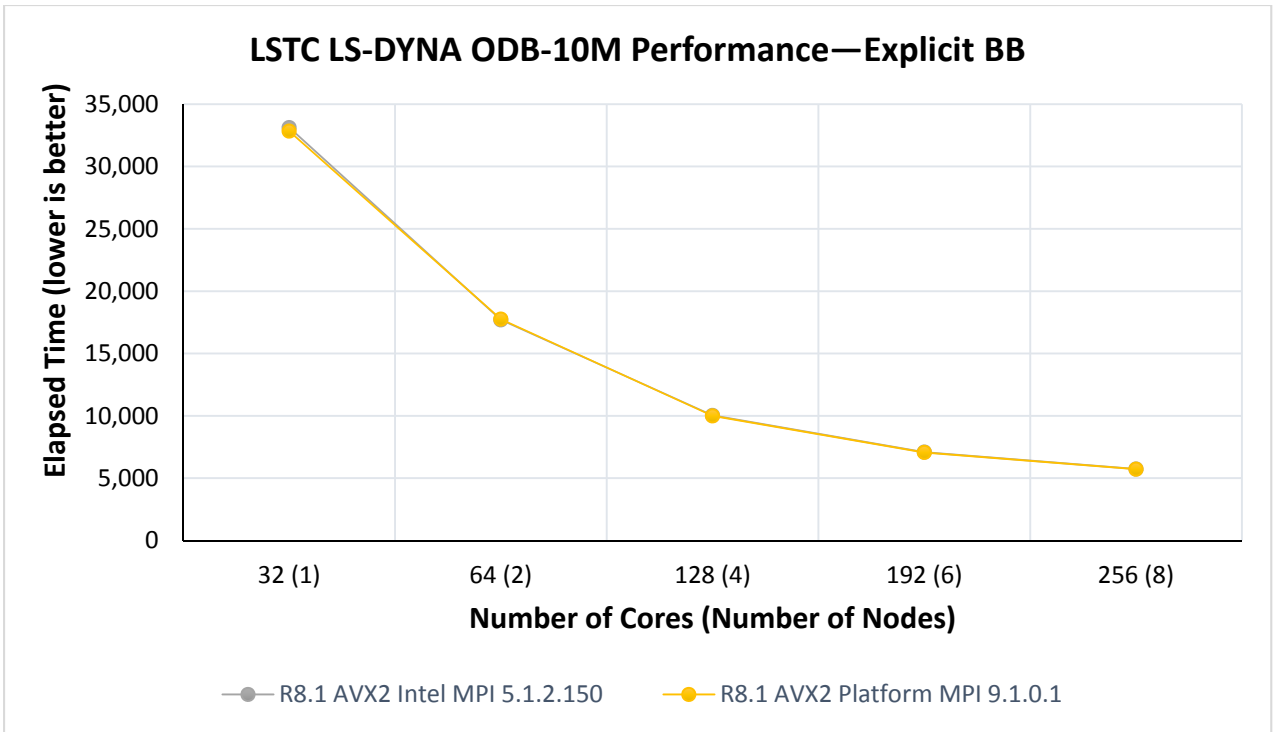
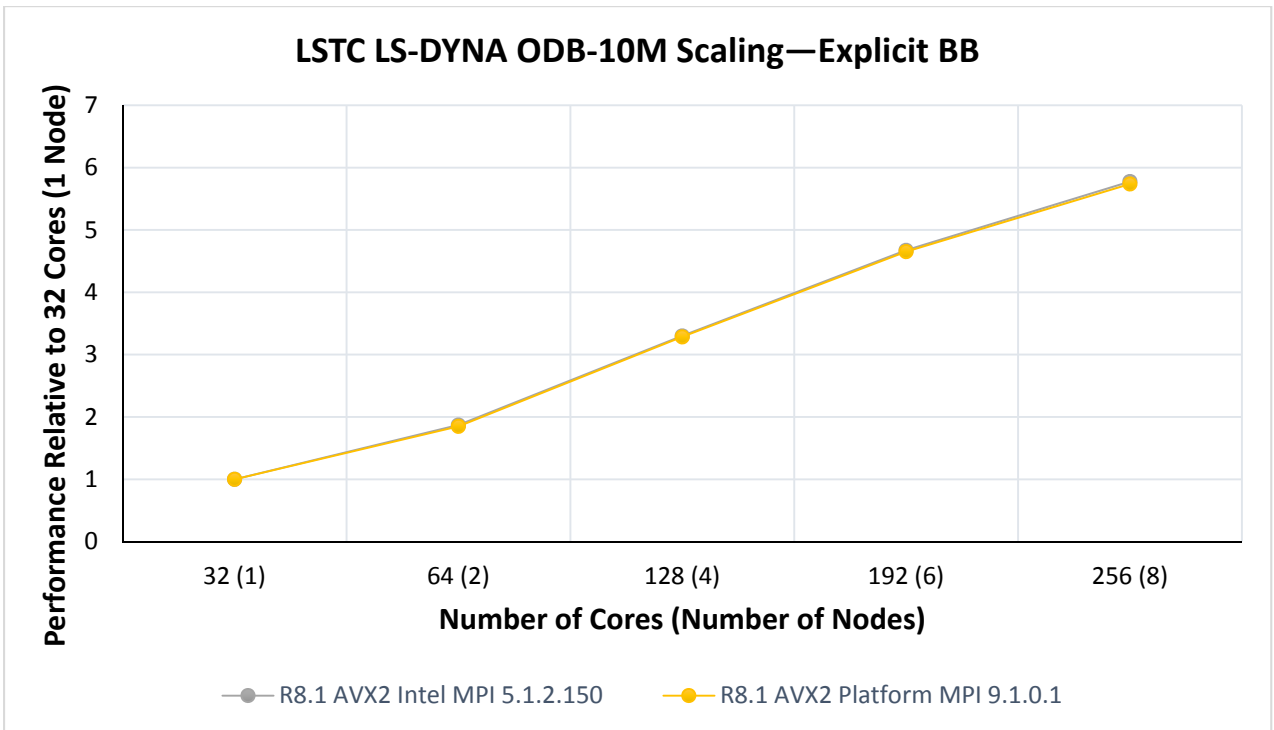Figure 37 LSTC LS-DYNA ODB-10M Performance—Explicit BB

Figure 38 LSTC LS-DYNA ODB-10M Scaling—Explicit BB

5   System Power Requirements

Power requirements and power budgeting is an important consideration when installing any new equipment. This section reports the power consumed by the three compute building block types for the different applications described in Section 3. This data was obtained by using metered rack power distribution units (PDU) and recording the actual power consumption of the building blocks during benchmarking.

The recommended system profile for the Dell HPC System for Manufacturing is the Performance Profile. This profile favors system performance over energy efficiency and typically yields an increase in performance of a few percent as compared with the DAPC profile. All application power measurements were performed using the Performance Profile. The application power graphs plot both the average and peak power recorded while running each benchmark dataset. Some of the benchmark datasets complete too quickly to provide accurate power measurements, so data for those benchmarks are not reported.

Figure 39 plots the average idle power consumption for the three compute building block types. Because there is a significant difference in idle power consumption for the Performance and DAPC profiles, idle power was recorded for both of these system profiles. The reported idle power is the average idle power consumed by one server of each building block type. The value of the average idle power measurement is shown at the top of each bar.
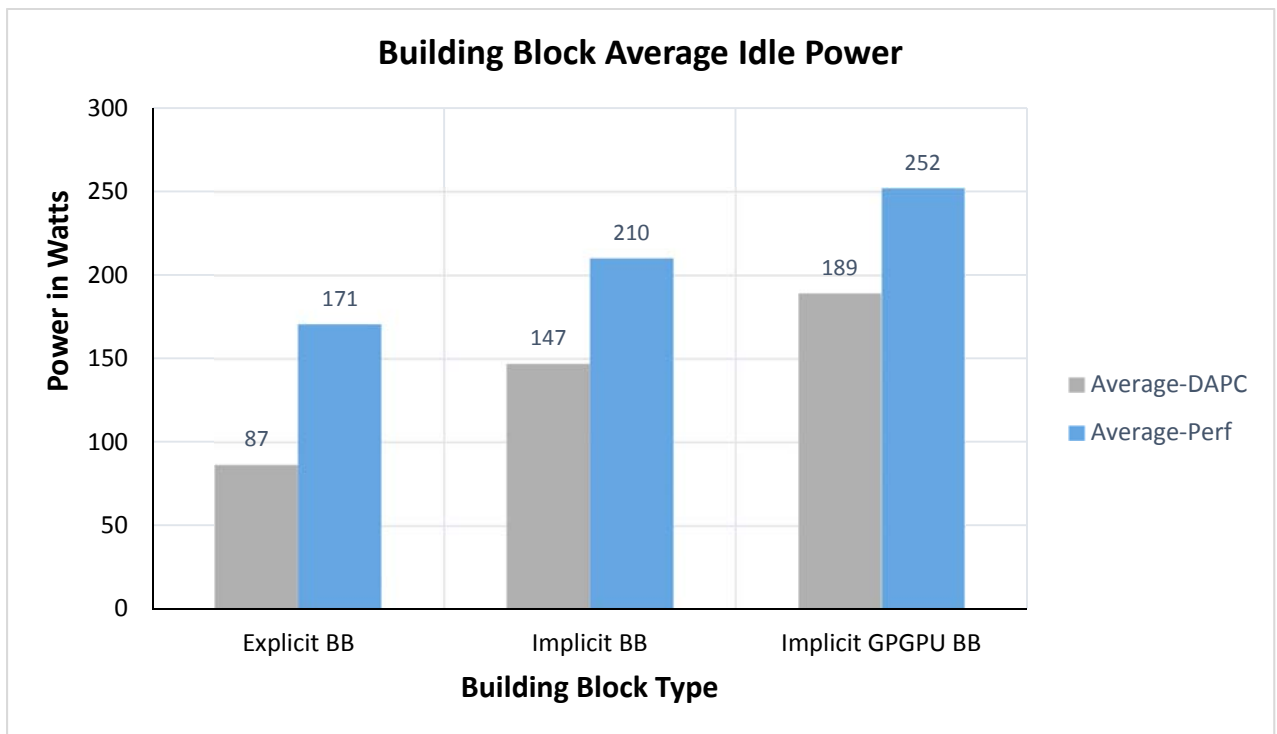

Figure 39 Building Block Average Idle Power

Figure 40 plots the power consumption of the three compute building block types when running HPL. The peak power draw for HPL is likely to be the maximum power draw of the system under load. Most

applications will not stress the system as much as HPL and will not consume as much power as HPL. This is also evident from the subsequent graphs in this section. For the Implicit GPGPU building block, power consumption was measured while running HPL on CPUs only and also while using the GPUs with a CUDA enabled version of HPL. The values of the average and peak power measurements are shown on the graph, with the average power displayed in the center of the average power bars and the peak power displayed on top of the bars.
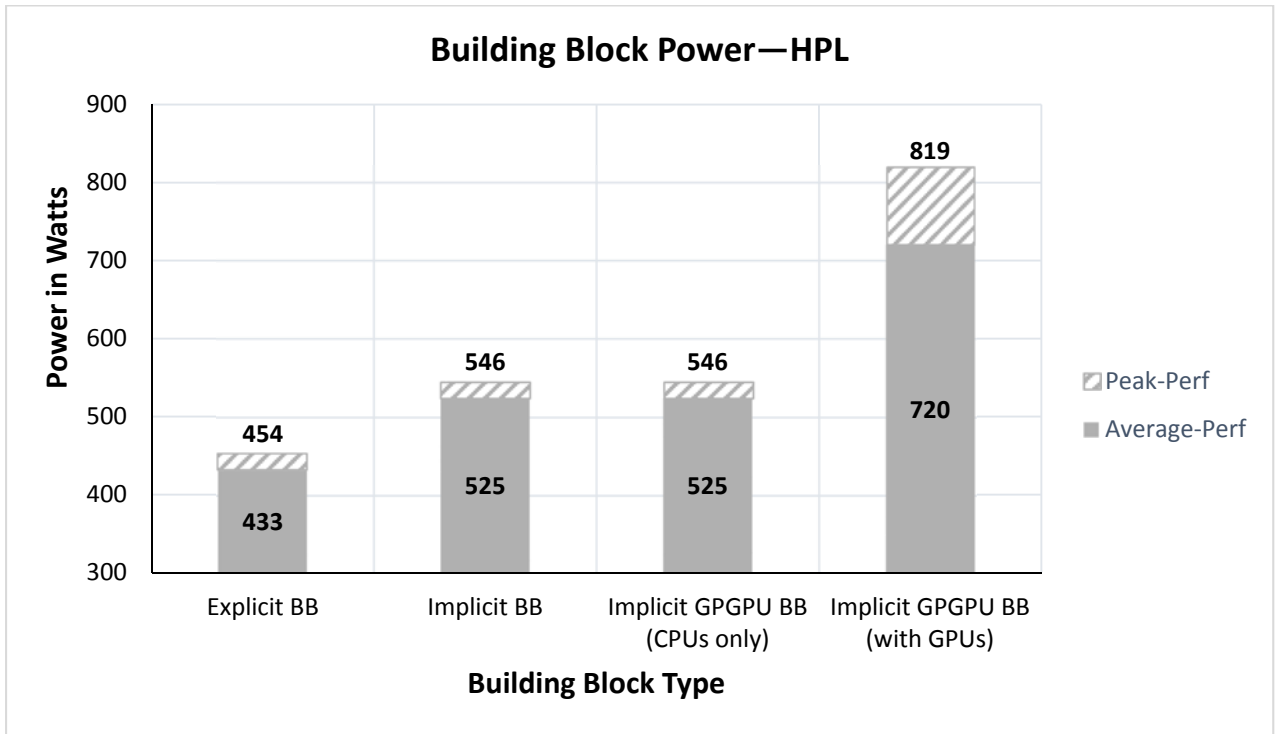

Figure 40 Building Block Power—HPL

Figure 41 plots the power consumption for one explicit building block when running a selection of ANSYS Fluent benchmark datasets. The values reported are the power required for one building block while running Fluent using 256 cores or eight nodes. The values of the average and peak power are shown on the graph, with the average power displayed in the center of the average power bars and the peak power displayed on top of the bars.

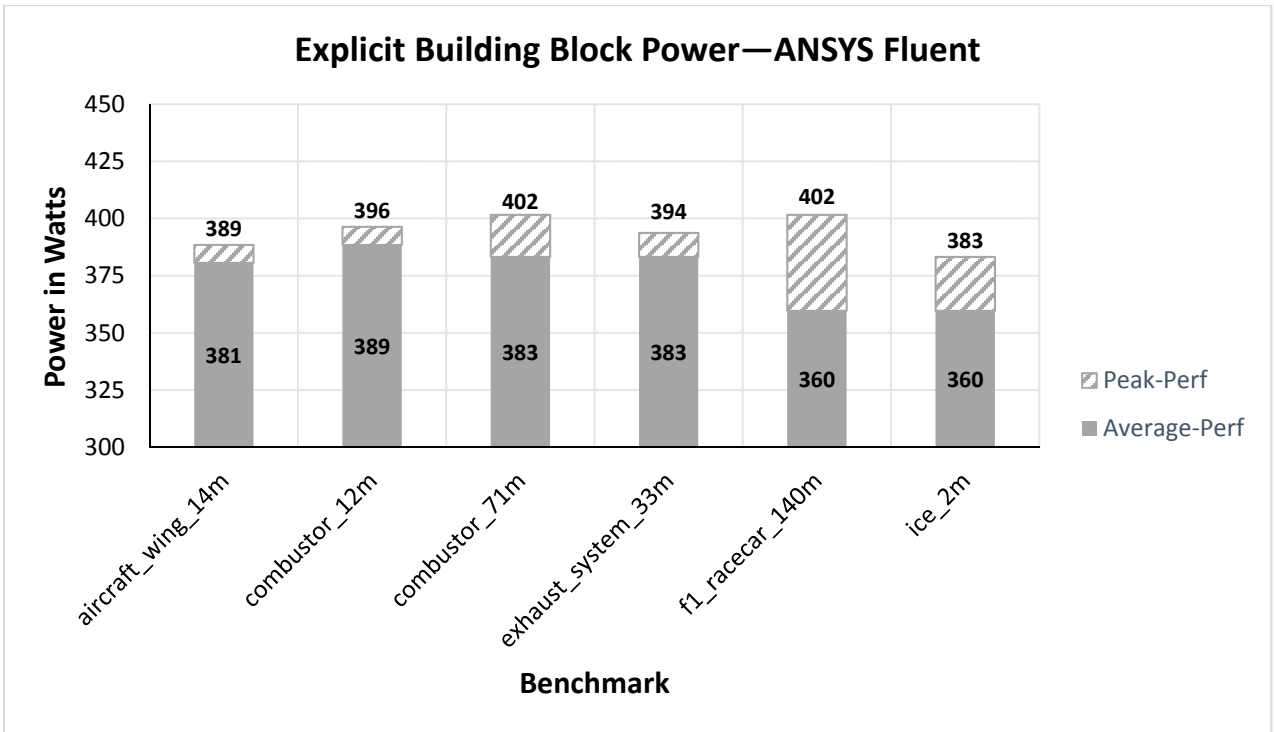**Explicit Building Block Power—ANSYS Fluent**

Figure 41 Explicit Building Block Power—ANSYS Fluent

Figure 42, Figure 43 and Figure 44 plot the power consumption for the three compute building block types when running a selection of ANSYS Mechanical benchmark datasets. The ANSYS Mechanical DMP solver was used for these power measurements. For the Explicit building block, the values reported are the power required for one building block while running ANSYS Mechanical by using 256 cores or eight nodes. For the Implicit building block, the power values were measured while running ANSYS Mechanical by using all CPU cores in the server. For the Implicit GPGPU building block, the power values were measured while running ANSYS Mechanical with all CPU cores plus both GPUs. The values of the measured average and peak power are shown on the graphs, with the average power displayed in the center of the average power bars and the peak power displayed above the bars.
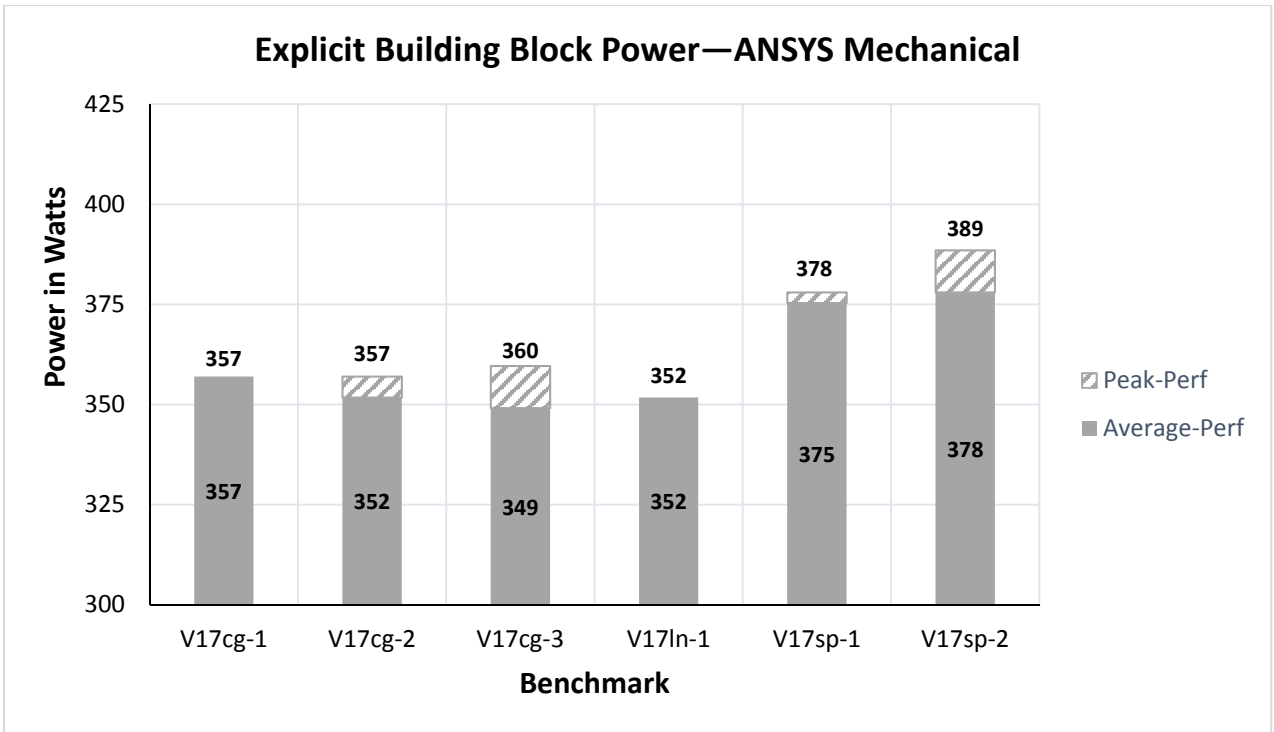
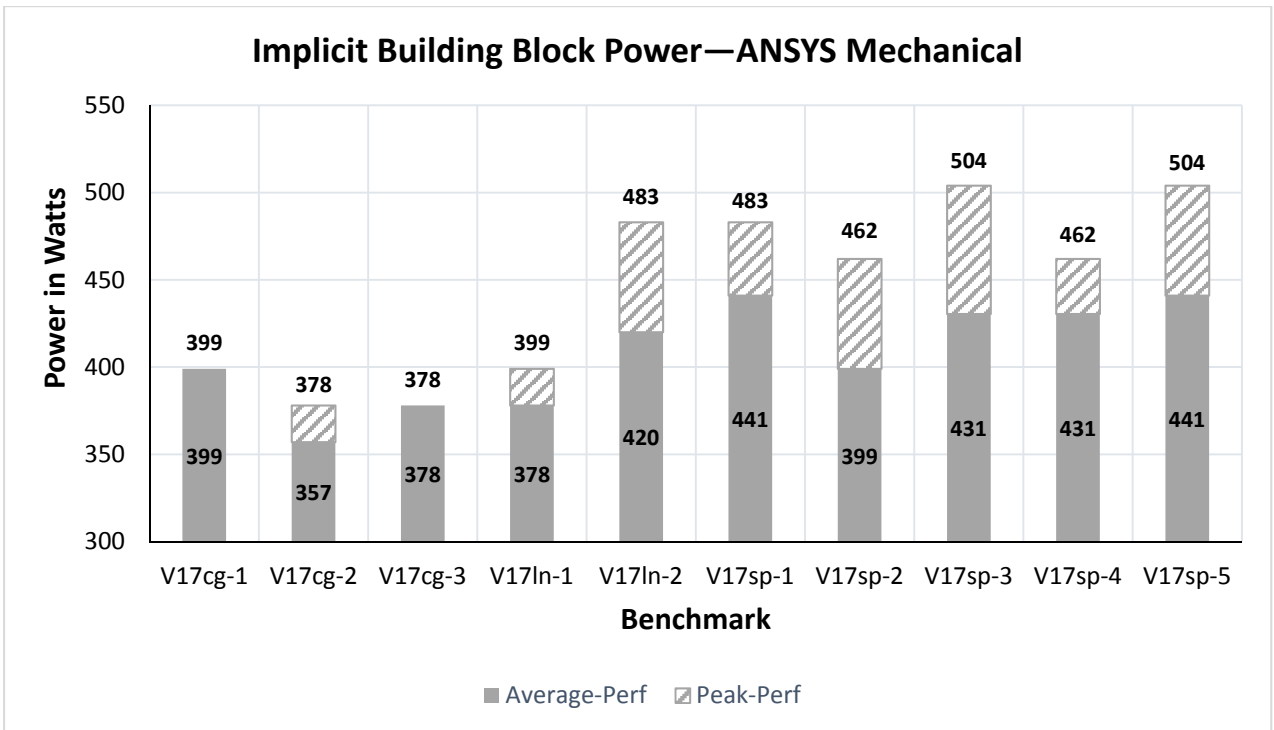Figure 42 Explicit Building Block Power—ANSYS Mechanical

Figure 43 Implicit Building Block Power—ANSYS Mechanical

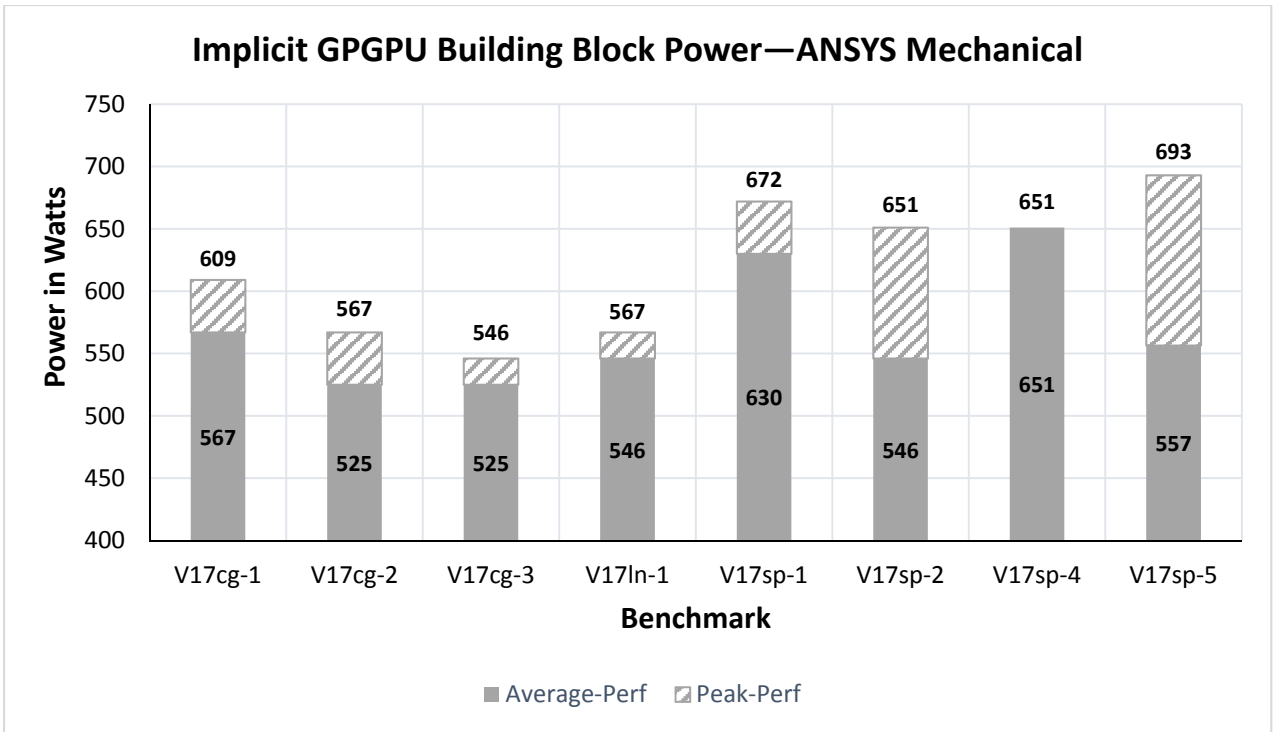**Implicit GPGPU Building Block Power—ANSYS Mechanical**

Figure 44 Implicit GPGPU Building Block Power—ANSYS Mechanical

Figure 45 plots the power consumption for one explicit building block when running a selection of CD-adapco STAR-CCM+ benchmark datasets. The values reported are the power required for one building block while running STAR-CCM+ using 256 cores or eight nodes. The values of the average and peak power are shown on the graph, with the average power displayed in the center of the average power bars and the peak power displayed on top of the bars.

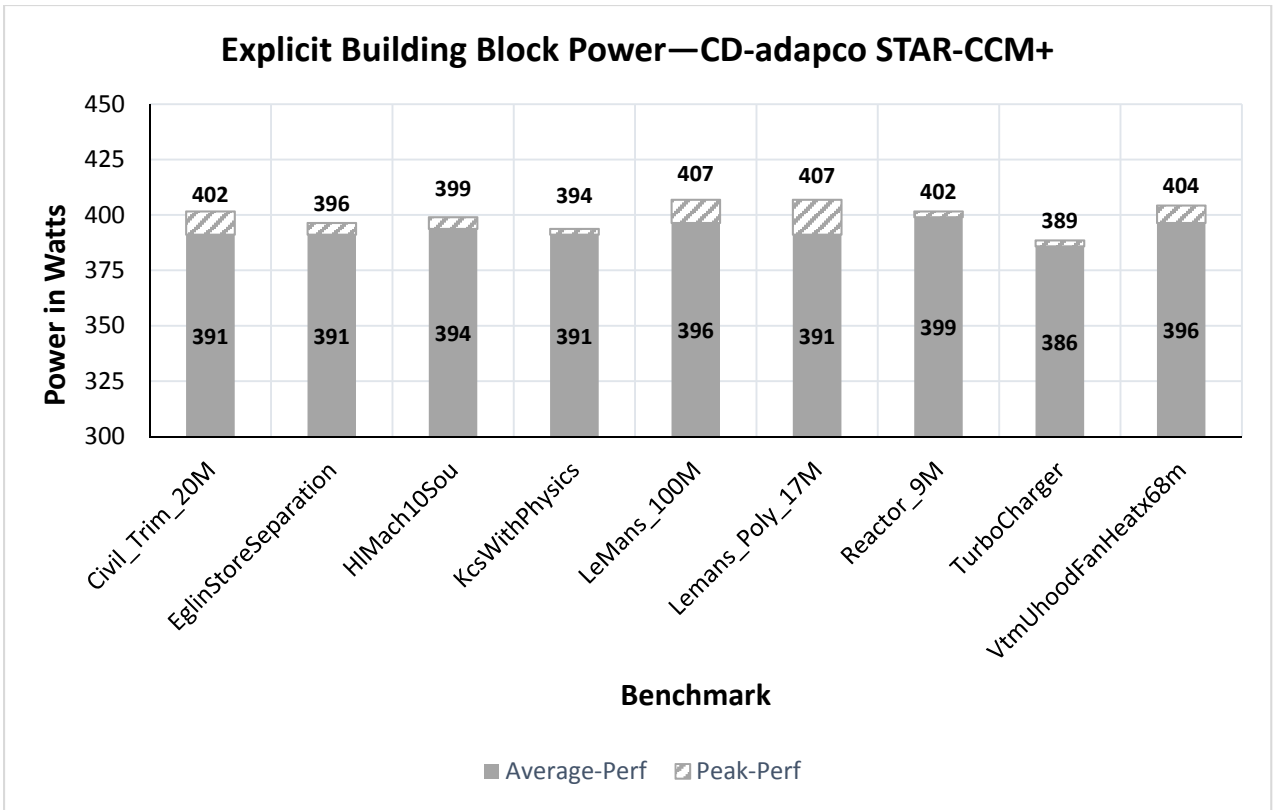**Explicit Building Block Power—CD-adapco STAR-CCM+**

Figure 45 Explicit Building Block Power—CD-adapco STAR-CCM+

Figure 46 plots the power consumption for one explicit building block when running the LSTC LS-DYNA benchmark datasets. The values reported are the power required for one building block while running LS-DYNA using 256 cores or eight nodes. The values of the average and peak power are shown on the graph, with the average power displayed in the center of the average power bars and the peak power displayed on top of the bars.
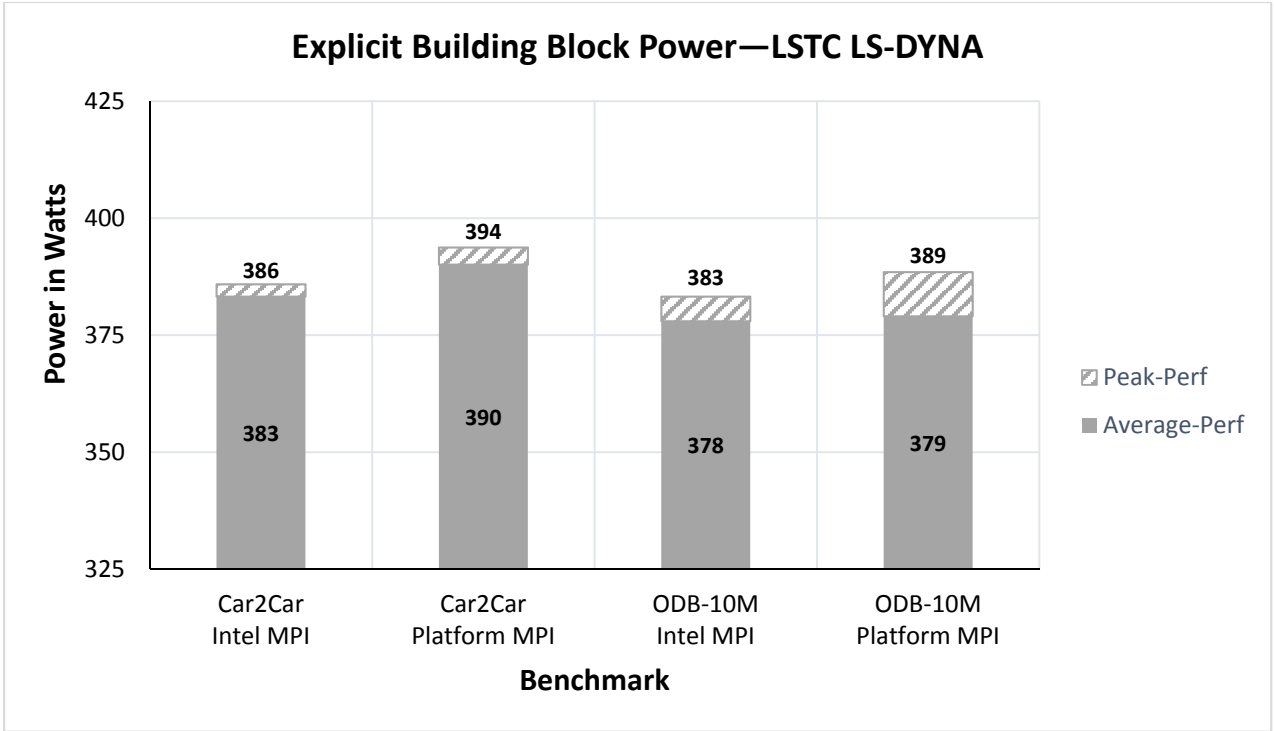
Figure 46 Explicit Building Block Power—LSTC LS-DYNA

6 Virtual Desktop Infrastructure (VDI)

A PowerEdge R730 VDI server was included in the reference system configured as previously described in Section 2.5. In order to evaluate the VDI server, NICE EnginFrame and Desktop Cloud Visualization (DCV) were installed on the reference system.

The NICE EnginFrame and DCV solution provides remote visualization software and a grid portal for managing remote visualization sessions and HPC job submission, control, and monitoring. NICE EnginFrame is the grid portal component. For this evaluation, EnginFrame 2015.1 r38894 was installed on the cluster master node. NICE DCV enables remote access to 2D and 3D applications over a standard network, providing GPU acceleration for 3D applications. DCV 2016.0 r16811 was installed on the VDI server.

With the NICE remote visualization solution, EnginFrame primarily provides management of remote visualization sessions and has no impact on the performance of the DCV component. For this evaluation, EnginFrame was tested to verify correct operation and successful integration with the overall system. It was also used to manage the remote desktop sessions on the VDI server. A screen capture of the EnginFrame VIEWS portal, showing an active Linux Desktop session, is shown in Figure 47.

Various applications and datasets were used to verify the operation of DCV, as listed in Table 6. This evaluation primarily focused on stability and correct operation of the NICE solution and a qualitative evaluation of the interactive application performance in both LAN and WAN environments. Screen captures showing several of the applications and datasets used for the evaluation are shown in Figure 48 through Figure 50.

Table 6.   DCV Evaluation Software

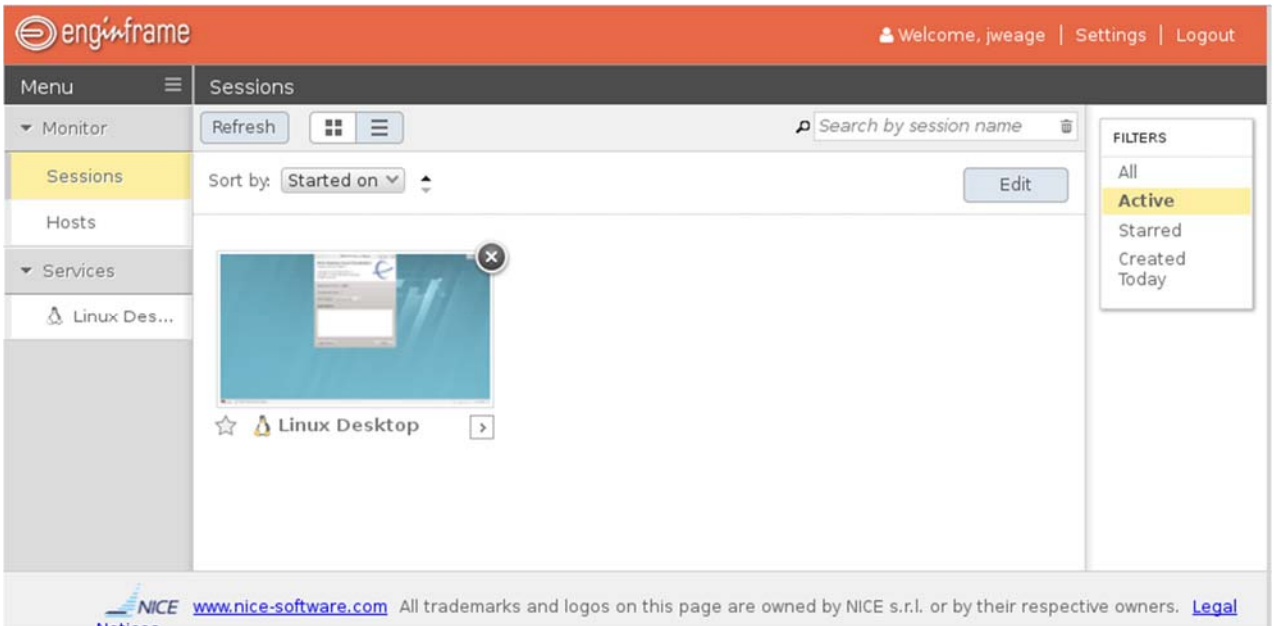| Software	Version	Datasets
LSTC LS-PrePost	4.3_centos7	car2car-ver10 ODB10M-ver14
ANSYS Fluent	v17.1	JetDemo small-indy
BETA CAE mETA Post	v16.1.0	OpenFOAM 2.4.0 motorBike
glmark2	2014.03	OpenGL ES 2.0

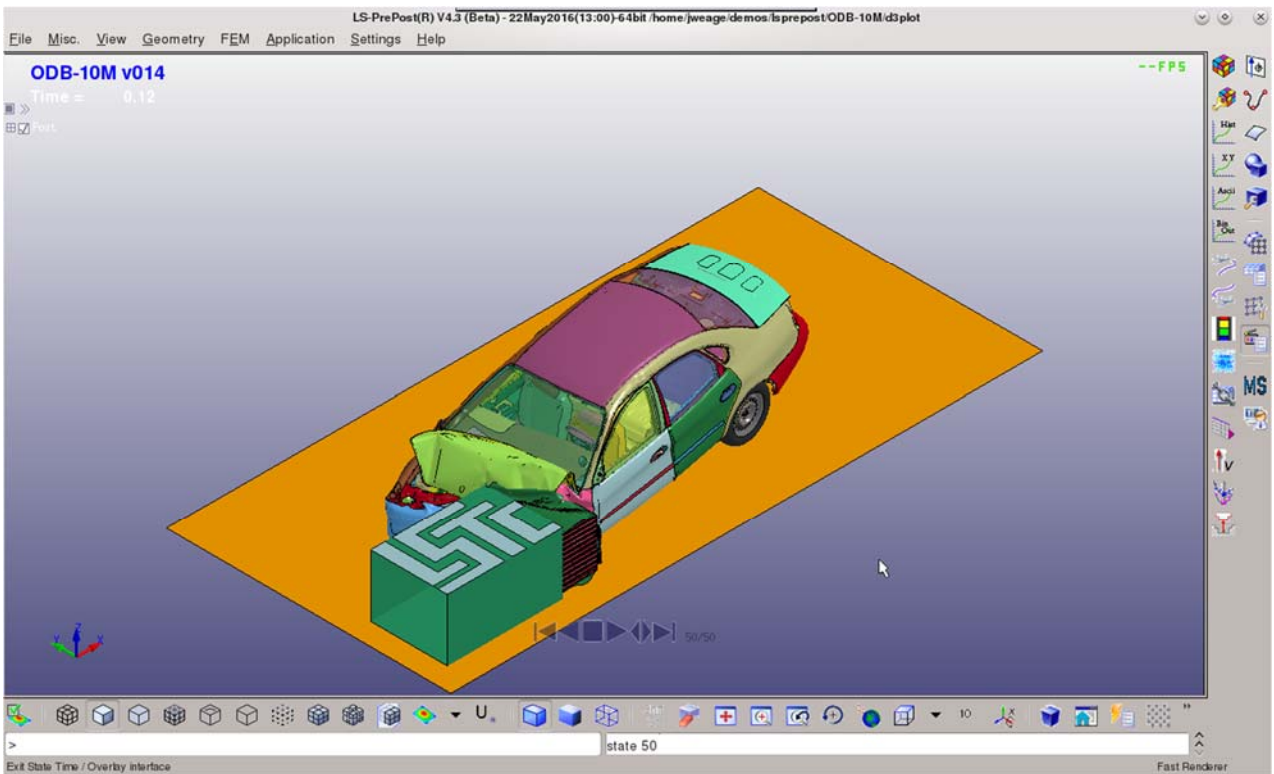Figure 47 NICE EnginFrame VIEWS Portal
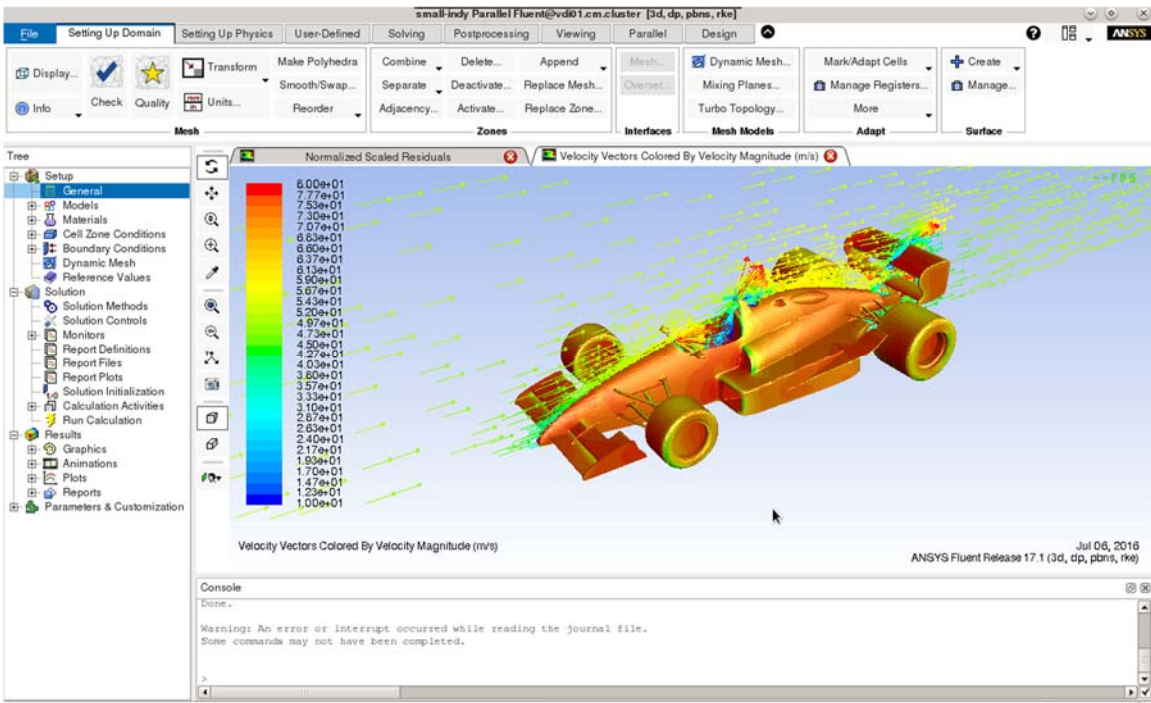

Figure 48 LS-PrePost with ODB-10M

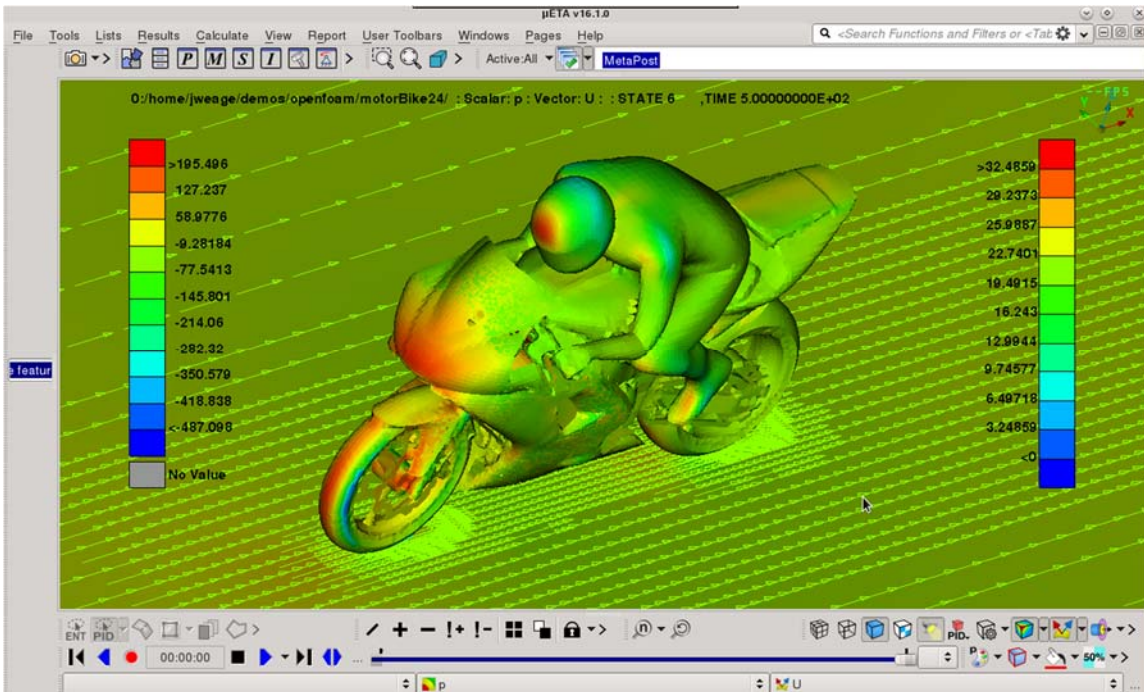Figure 49 Fluent with small-indy

Figure 50 mETA Post with motorbike

One of the features of the NICE DCV Endstation client is the DCV Console. The console allows the user to dynamically adjust quality vs network bandwidth utilization by using a slider bar and to monitor the bandwidth being used by the client. For most uses, the 60% setting provides a good balance between bandwidth usage and image quality. One of the useful features of the NICE DCV solution is that the final image delivered to the client after display updates have stopped is always lossless, regardless of the quality level setting. This ensures that static images are always shown with full quality on the client.
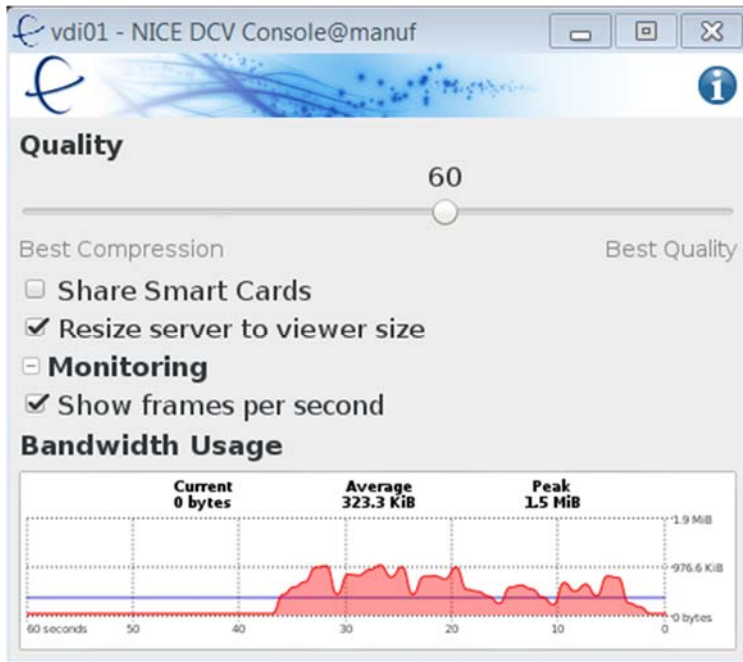

Figure 51 NICE DCV Endstation Console

For testing, the 60% quality setting was used for the client. In a LAN setting, with significant bandwidth and low latency, the remote application responsiveness and rendering performance was very good. In a WAN environment, application responsiveness and rendering performance was also very good as long as network latency remained less than about 150 ms and sufficient network bandwidth was available. When network latency exceeded about 150 ms, delays in the application response became noticeable, which is expected. NICE recommends that for these situations, the user can adjust compression on the fly to accommodate for low bandwidth using the DCV Console.

For typical applications at the 60% quality level, average network bandwidth utilization ranged from 150 to 800 KiBps during display updates and dropped to 0 kb/s for static images. Peak network bandwidth was approximately 1.0 MiBps for all of the tested applications at the 60% quality setting. At higher quality settings, average network bandwidth gradually increased, with a significant increase in bandwidth utilization from the 90% to the Lossless setting.

Overall, the NICE DCV solution performed well and offers a good solution for remote visualization users.

7 Conclusion

This technical white paper presents a validated architecture for the Dell HPC System for Manufacturing. The detailed analysis of the building block configurations demonstrate that the system is architected for a specific purpose—to provide a comprehensive HPC solution for the manufacturing domain. The design takes into account computation, storage, networking, visualization and software requirements and provides a solution that is easy to install, configure and manage, with installation services, and support readily available.

The performance benchmarking bears out the system design, providing actual measured results on the system for applications relevant to the manufacturing domain. Additionally, system power data is presented to allow for upfront power budgeting for this solution.