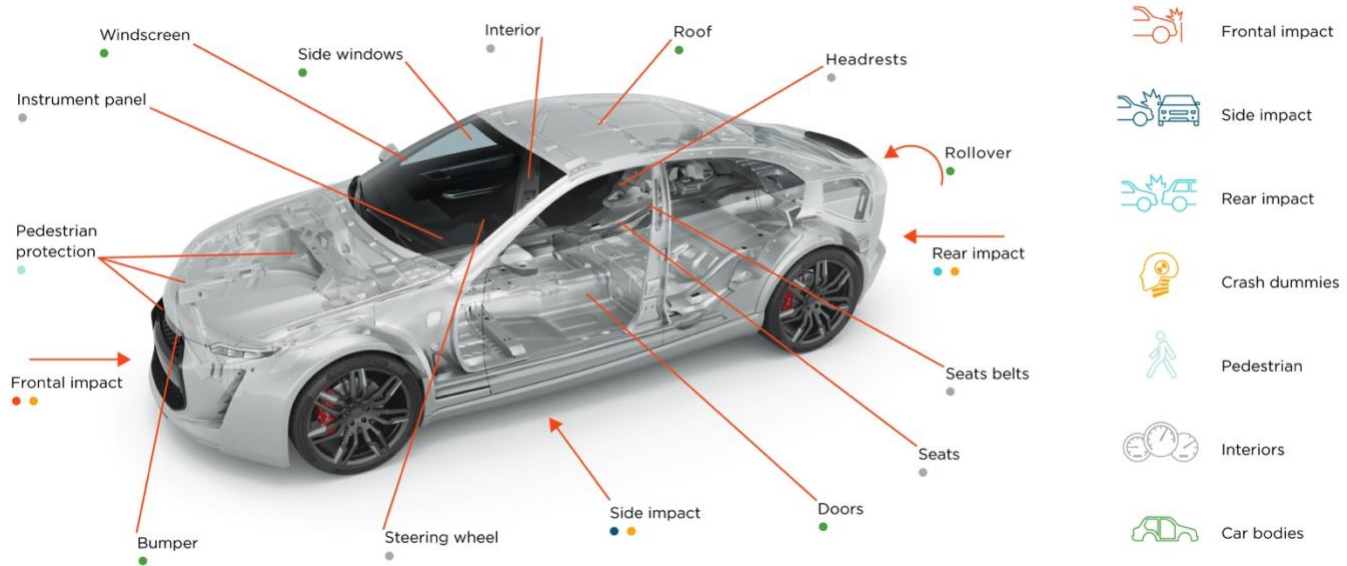△ **ALTAIR**

ASSURING SCALABILITY: ALTAIR RADIOSS™ DELIVERS ROBUST RESULTS QUICKLY FOR CRASH-SAFE VEHICLE DESIGNS

Eric Lequiniou - VP Radioss Development & Altair Solver **/** Dario Mendolicchio - HyperWorks Business Development **/** May 26, 2020

Introduction

In the race to deliver safe new vehicles, efficient and accurate simulation of structural performance under different crash scenarios is a key alternative costly physical testing. With growing model complexity, the demand for CPUs is increasing. To assure a fast job turnaround time, HPC and solver scalability is critical. This article shows that by working together with market-leading hardware developers to keep one step ahead of customer needs, Altair Radioss™ offers fast, efficient structural analysis irrespective of the hardware the solver is running on: scalability is proven and assured.

Proving the crashworthiness of a new vehicle is not only a legal requirement, it also significantly influences customer decisions to purchase, impacting the overall success of a vehicle. For the carmakers, this represents an expensive and time-consuming part of any vehicle project.

Since their widescale adoption in the car industry, simulation tools that enable fast and cost-efficient design for crashworthiness have reduced the number of physical crash tests and provided an advantage in problem-solving and productivity over competitors.

With extensive, proven usage worldwide within the car industry, Radioss is a market-leading structural analysis solver for highly non-linear problems under dynamic loadings that delivers improved crashworthiness, safety, and manufacturability of structural vehicle designs. One reason for Radioss' success is that, by working together with market-leading hardware developers, it has kept one step ahead of customer demands for fast, efficient structural analysis irrespective of the hardware the solver is running on: assuring scalability. Only with Altair's understanding of how to deliver state-of-the-art software for evolving HPC technology, can designers attain robust results coupled with a fast job turnaround time.

Challenges in CAE

The first crash test simulation model in 1986 was a VW Polo front structure comprising 5661 finite elements, crash duration of 60 milliseconds, 4 hours runtime. For comparison, today's vehicle models are typically about 6 to 8 million elements, and if the trend of doubling every 5 years continues, model sizes of 26 million elements will be commonplace in 2030, as shown in Figure 1.

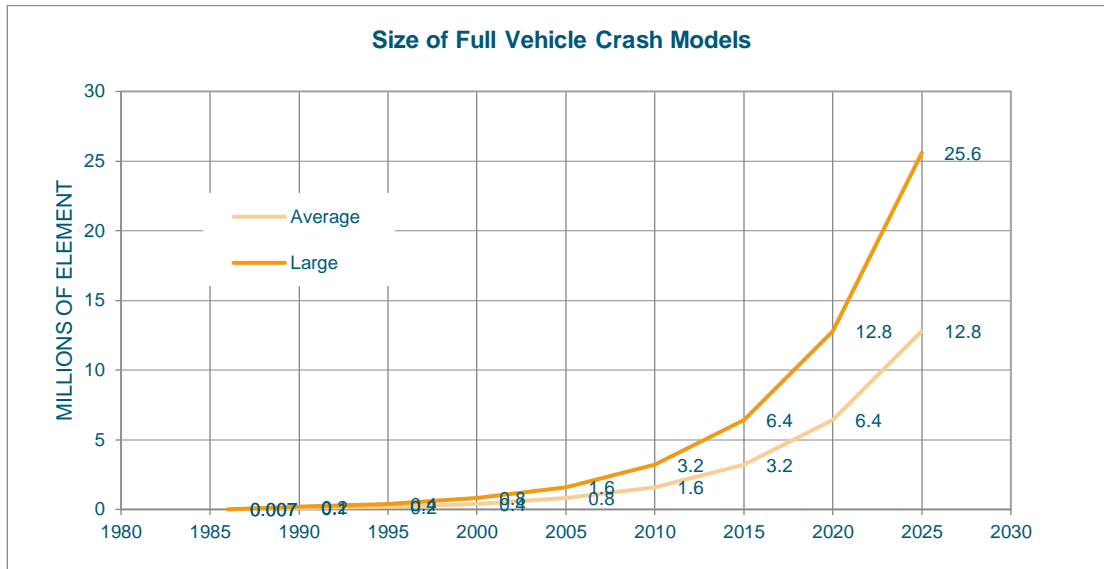**Size of Full Vehicle Crash Models**

Figure 1 – Evolution of Full Vehicle Crash Models

Regulatory authorities are demanding more test conditions and load cases, which makes physical testing alone almost prohibitive on cost. It is contributing to the need for larger models to understand in more detail modern material deformation, stress, rupture and propagation of failure, along with a multiphysics approach for airbags, fluid in the tank, etc. Overall, to attain robust, optimized designs many more runs using different load cases are necessary, but the accuracy of the results obtained becomes critical when predictive models replace physical testing. All this implies a need for many more, large model simulation runs that lead to a significant demand for increased CPU performance.

HPC Evolution

Contributory factors to CPU performance - transistors, power, frequency - have steadily evolved such that until about 2005, CPU performance doubled every 18 months (Moore's Law). This is no longer true partly due to reducing power, frequency and reaching the CPU inherent architecture electrical/thermal performance limits. From 2005, new generation CPUs having increased number of cores has provided a more efficient multicore and many-core architectures (violet line), along with technologies such as clusters with high speed interconnect. At this point, scalability then becomes strategic to meet computing needs; shown in Figure 2.
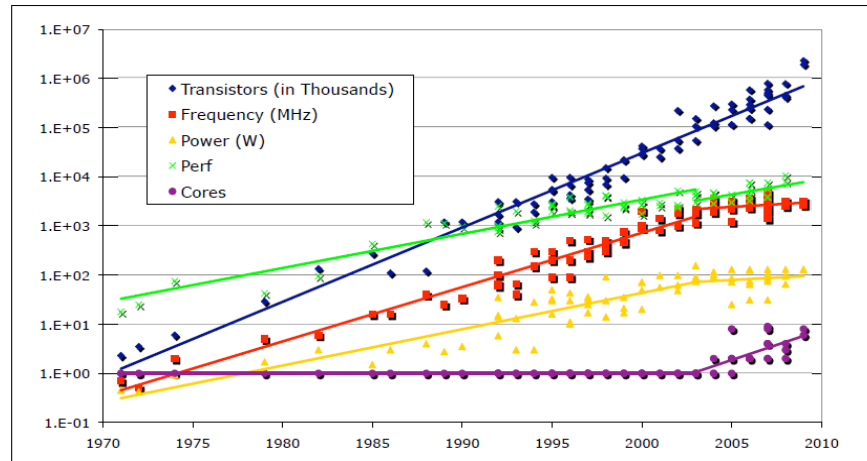
Figure 2 – HPC Revolution: CPU Performance Factors
Source: Intel Corp

Looking closer at the evolution of a single node (dual socket), the CPU frequency has not changed much but the number of cores per node has increased across Xeon generations by a factor 12x in 10 years. The new generation CPUs from Intel® (Skylake™ Gold and Platinum Cascade Lake™) offer a big performance boost owing to AVX512 vectorization technology and better memory bandwidth due to having 6 memory channels compared with 4 for the previous Broadwell generation.
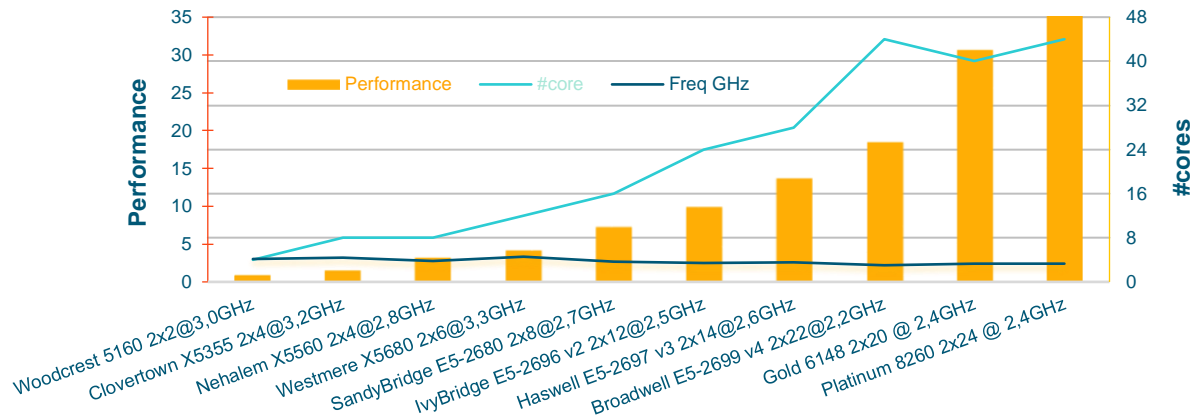

Figure 3 – Benchmarking CPU Performance

From a Radioss perspective, the performance gain from new generation CPUs is that the software can exploit all the cores by being able to scale both on single node on new CPUs, along with more cores on clusters running on multinodes. Within 10 Xeon® generations, Radioss performance has increased by a factor 35x, a figure unobtainable without parallelism. For code, on the other hand, the performance improvement between today's CPU and those of 10 years ago is less drastic.

Radioss in Today's HPC Hardware Landscape

Regarding CPUs, Altair have assessed Radioss on the latest generation CPUs on the market, with the findings available in open publications. [1,2,3]

Intel's, Xeon® Gold and Xeon® Platinum with up to 28 cores per socket and 56 cores per node for their Cascade Lake SP, has been the dominant player for years and a reference point in terms of CPU performance.

Recently, AMD® released its second generation EPYC™ processor with up to 64 cores per socket and, 128 cores per node, providing a credible alternative in terms of per- node performance. ARM®'s, ARM64™ architecture -, implemented by Marvell® in the ThunderX2® and by Fujitsu®, with their recently announced A64FX™ chip -, also has some increased ambition in the HPC server market. [5]

Interconnect is a fundamental ingredient to get the best out of a cluster, with Infinband and Omni Path key for HPC and scalability.

Radioss Scalability

Considering today's HPC scenario, hybrid parallelization schemes (MPI and OpenMP) are considered the best methods to provide Radioss with the necessary flexibility to attain the highest scalability over thousands of cores. It does, however, need to be tuned according to the CPU architecture; accounting for the number of sockets per node and, cores per CPU socket. Figure 4 illustrates an example of scalability vs number of nodes: pure MPP parallelization reaches its maximum around 128 cores, whereas Hybrid-MPP parallelization continues to be efficient to much greater core numbers. The inflection point depends on the size of the benchmark, along with a combination of other factors, but it demonstrates that hybrid yields higher efficiencies when pure MPI scalability.
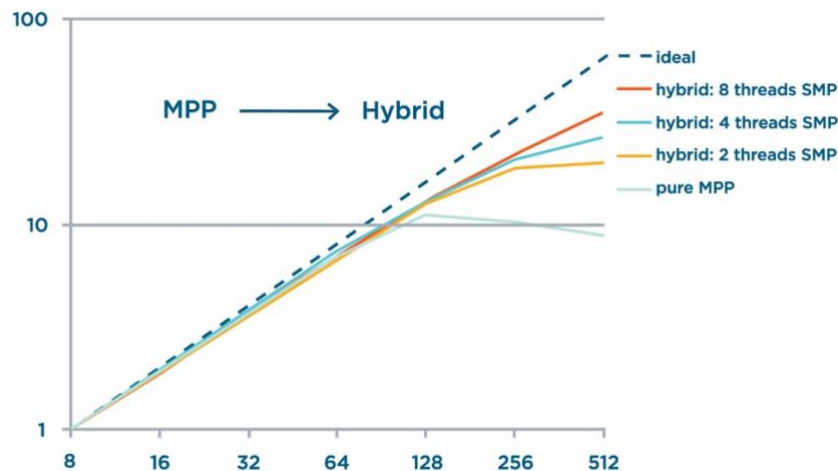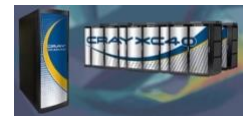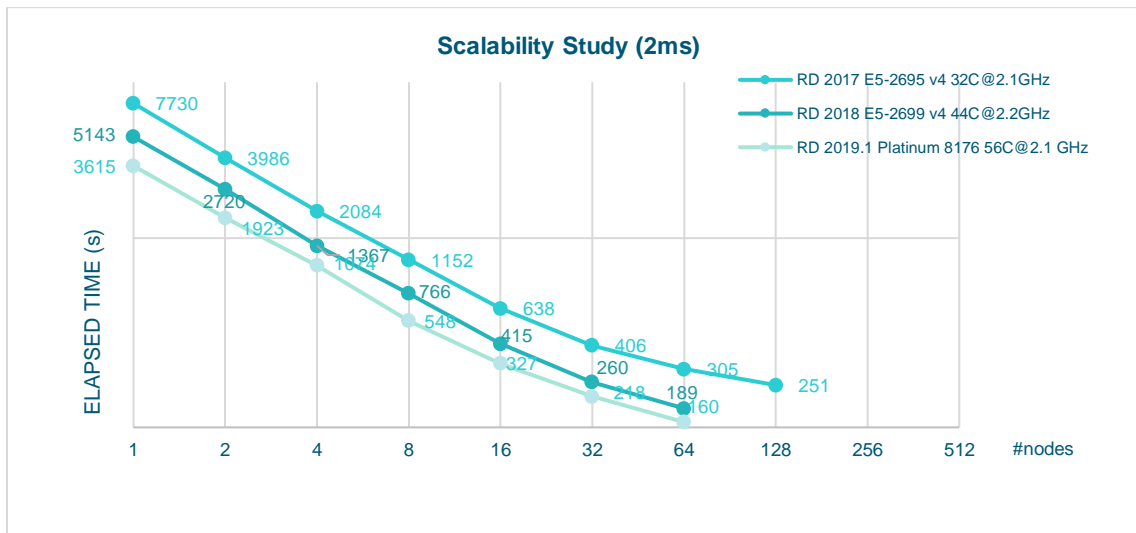

Figure 4 – Scalability: Better Efficiency provided by Hybrid Parallelization compared with pure MPI

In summary, Radioss is a highly parallel code with a hybrid model, where MPI parallelization is based on a domain decomposition approach and OpenMP exploits shared memory parallelism, along with higher computational efficiency by vectorization. Highly efficient on large HPC clusters, exploiting OpenMP/ MPI gives good flexibility, the ability to tune the performance on any type of cluster. Radioss delivers robust, fully repetitive results due to its parallel arithmetic option. When selected, single precision results are delivered 1.4x faster than for double precision.

Scalability in Radioss Full Vehicle Model

To measure Radioss full vehicle model performance, the public Taurus Model with 10 million elements is applied.[4] It can run 6H18 on 64 nodes/2816 cores.

Scalability Study (2ms)

Legend:
- RD 2017 E5-2695 v4 32C@2.1GHz
- RD 2018 E5-2699 v4 44C@2.2GHz
- RD 2019.1 Platinum 8176 56C@2.1 GHz

ELAPSED TIME (s) vs #nodes

RD 2017: 7730, 3986, 2084, 1152, 638, 406, 305, 251
RD 2018: 5143, 2720, 1367, 766, 415, 260, 189
RD 2019.1: 3615, 1923, 1027, 548, 327, 248, 160

32	64	128		256	512	1024	2048	4096	8192	16384	#cores
44	88	176		352	704	1408	2816				#cores (E5-2699 v4)
56	112	224		448	896	1792	3584				#cores (Platinum 8176)

Taurus 10 million elements

Figure 5 – Comparison between Performance of Multiple Nodes, Cores/Platform for 10 million element Taurus Model

This large model is used for the scalability of clusters, whereas the previous NEON smaller model was used for benchmarking CPU on a single node. Although an old car design, it was chosen because it is larger than what is currently used in the industry and retained because it enables comparison between previous hardware configurations/set-ups.

Looking at the multi-nodes performance using the Taurus model, Figure 5 shows results that combine testing hardware with changes to code, too. Owing to changes to the architecture for each version, a 1:1 comparison is effectively invalid, but performance values of a multiple nodes, and cores/platforms can be assessed. Working in partnership with Cray, the model is always run on the same computer (CrayX40) that is upgraded from time to time, with the same interconnect but different generation of processors.

When comparing 1 node elapsed time between Radioss version 2017 with Radioss 2018, while the processor is effectively the same, the significant difference (7730s v 5143s) can be attributed to not only the higher number of cores, but also to Altair Radioss code optimizations for v2018. Radioss 2018 is the first to support the AVX512 vector instruction set, along with maintained scalability across increased nodes. The improvement in performance between Radioss 2019 and 2018 (5143s v 3615s) is mainly due to a new, more performant processor (Skylake and Cascade Lake), which supports AVX512 vectorization and demonstrates scalability. Again, the combination of new generations of Radioss software and hardware now run about 2x faster, which is a significant performance improvement of great interest to customers.

Optimizing Hardware and Setup to Achieve Robust Results Efficiently

Some general hardware recommendations to obtain good performance with Altair Radioss involve using a 2-socket cluster node under Linux. All nodes should have the same type of CPU across the cluster and likewise be dedicated to a single Radioss job. Using high-speed interconnect (such as Omni-Path or Infiniband) is extremely important to achieve good scalability.

When setting up the job, the same number of MPI per socket is recommended. To attain good scalability, it is important to balance CPU and network load. If the number of elements per core is too few, if network communication increases significantly compared with CPU time, then scalability is compromised. The aim is to keep cores busy by, for example, splitting the model in order to have at least 4000 elements/core for current generation CPU (up to Broadwell). New generation CPUs are faster (from Skylake) so the number of elements/cores should be at least 10000.

Where core numbers are low, say less than 256, pure MPI gives good scalability, whereas for higher core numbers a hybrid parallelization scheme is necessary.

Question 1: How many cores/nodes should I use to minimize elapsed time for a 10 million element model?

Looking at 2 different hardware configurations while keeping in mind the recommended minimum value of elements/core, i.e. 4000 for Broadwell, and 10000 for new generation Platinum, the computation approach is the same but the configuration to minimize elapsed time is different.

Hardware Configuration	Broadwell E5-2699 v4 2 sockets, 22 cores @2.2GHz		Platinum 8260 2 socket 24 cores @2.4GHz	
Compute max N° of cores:	10 000 000 / 4000	= 2500 cores	10 000 000 / 10 000	= 1000 cores
Compute number of MPI (domains):	2500 cores / 22 OMP*	= 114 MPI (about)	1000 cores / 24 OMP*	= 42 MPI (about)
Compute number of nodes:	114 / 2	= 57 nodes (= 2508 cores)	42 / 2	= 21 nodes (= 1008 cores)
Minimized Elapsed Time configuration:		57 nodes 2508 cores		21 nodes 1008 cores

*OMP: OpenMP

Table 1 – Minimizing Elapsed Time for Different Hardware Configurations

Question 2: How many nodes should I use to get 80% efficiency (10 million element model)?

From an IT perspective, the overall aim is to maximize cluster efficiency, where efficiency is defined as actual/ideal speedup. The answer is obtained by running a scalability study with a representative sized model and compute efficiency at 2, 4, 8, …. 16 nodes.

In Figure 6 the results for running a 10M model on Radioss 2019, with an Intel Skylake (Platinum 8176, 56 cores/node @ 2.1GHz) show that 82% efficiency can be attained using 8 nodes (448 cores).
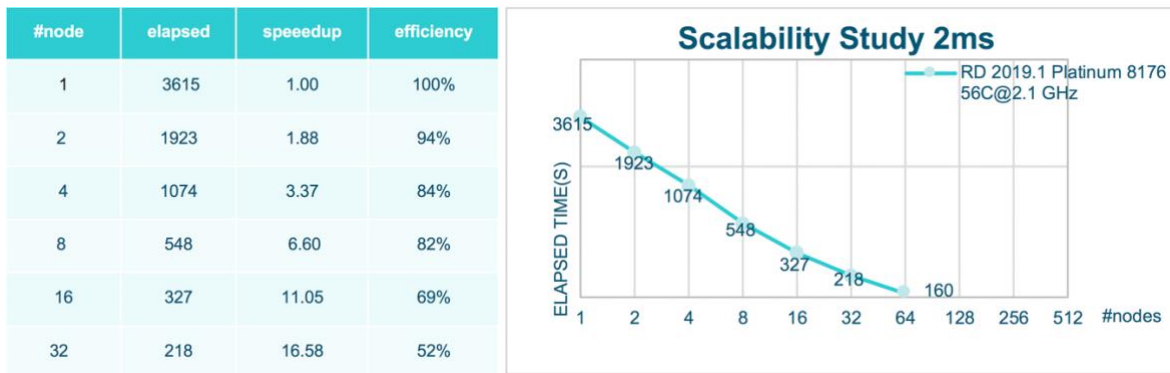
#node	elapsed	speeedup	efficiency
1	3615	1.00	100%
2	1923	1.88	94%
4	1074	3.37	84%
8	548	6.60	82%
16	327	11.05	69%
32	218	16.58	52%

Figure 6 – Determine Number of Nodes to Attain an 80% Cluster Efficiency

Radioss Advanced Numerical Methods

Radioss employs advanced numerical methods to ensure it delivers the solution fast, such as single precision, Tetra10 with dynamic condensation, Advanced Mass Scaling (AMS).

Beside HPC exploitation, Radioss code offers several options and strategies to accelerate the solution. Released in Radioss 2018, a tetra10 element which has a higher time step can give a gain of 2x while retaining the accuracy compared with a regular tetra10 element. It is particularly well suited for electronics models which generally contain many tiny Tetra10 elements, which are bounding the time step.

Another method available for increasing or maintaining higher time step is Advanced Mass Scaling (AMS) which enables the time step to be increased without affecting the kinetic energy: a recognised problem with mass scaling is kinetic energy. As shown in Figure 7, it can be applied to full models or user defined parts. In Radioss 2018, elements with a time step less than that defined by the user are then automatically selected for treatment by AMS. The idea behind this automatization is to apply AMS selectively, only for very fine mesh parts maintaining a higher time step for the entire model without adding too much CPU overcost due to AMS specific treatments which remain localized to few elements only.
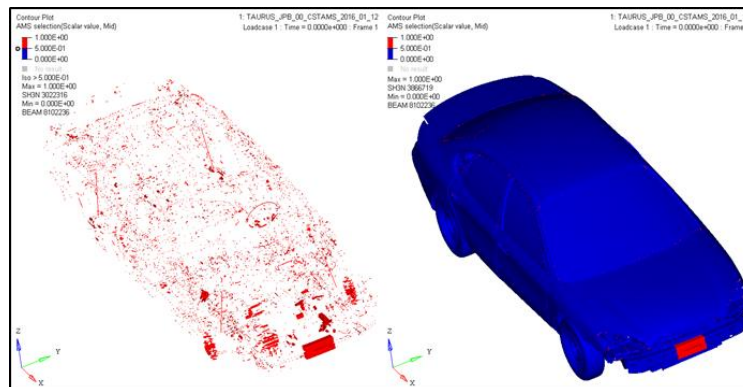

Figure 7 – Advanced Mass Scaling (AMS): Increase Time Step Without Affecting Kinetic Energy

Another approach is by employing multi-domains. Each domain is computed separately using its own time step which is very efficient for Fluid Structure Interactions (FSI), where the time step of the fluid domain is typically different from structure time step, such as in a ditching scenario. Typically, results can be obtained by a factor of 2x to 10x faster using multi-domains, depending on the ratio between the time steps and the ratio between the number of elements.

Conclusions and Outlook

In the race to deliver new safe vehicles, efficient and accurate simulation of performance under different crash scenarios is key. With growing model complexity, the demand for CPUs is increasing. To assure a fast job turnaround time, HPC and solver scalability is critical. By working together with market-leading hardware developers to keep one step ahead of customer needs, Radioss offers fast, efficient structural analysis. Irrespective of the hardware the solver is running on: scalability is proven and assured.

In addition to running on clusters of cores across the latest CPUs, Altair's state-of-the-art solver employs HPC with evolving advanced hybrid parallelization technologies, along with numerical methods to enhance intrinsic performance all at an affordable price for engineers to attain robust results coupled with a fast job turnaround time.

While large HPC systems are still mostly reserved to large companies, the growing adoption of the cloud is making HPC affordable for many more small business customers. A recent study [7] has proven that HPC on the cloud is as performant as best-in-class supercomputers. This opens additional perspectives of HPC to solve a broader number of complex challenges. This is especially true when combining the parallel performance of Altair's Radioss solver with the efficiency of PBS Professional™ to maximize the CPU resource utilization seamlessly on-premises and on the cloud: a unique feature that only Altair can provide to its customers.

References & Further Information

1. Altair Radioss performance on Intel Download
2. Altair Radioss performance on AMD Download
3. Altair Radioss performance on ARM Download
4. "Taurus Crash Model": Altair University
5. "Crash Simulation with Arm and the Catalyst UK Project", Altair Blog, January 2020
6. Presentations made at SC19 International Conference for High Performance Computing, Networking, Storage and Analysis, Colorado, 17-22 November 2019
7. "Powering Altair Radioss™ Crash Simulations in Microsoft Azure", Altair Blog, April 2020