2.8 SPECFEM3D

2.8.1 Description of the code

SPECFEM3D Globe is an HPC scientific code developed by the Computational Infrastructure for Geodynamics (CIG), Princeton University (USA), CNRS and University of Marseille (France) and ETH Zurich (Switzerland). It simulates seismic wave propagation at the local or regional scale based upon spectral-element method (SEM) with very good accuracy and convergence properties. This approach, combining finite elements and pseudo-spectral methods, allows the formulation of the seismic wave equations with a greater accuracy and flexibility if compared to more traditional methodologies. SPECFEM3D is a Fortran application, but a subset of the globe version has been ported to C, to experiment with CUDA, StarSs and OpenCL. This subset contains the main computation loop of the main application. The full application is composed of 50k lines of Fortran, while the subset contains 3k lines of C.

SPECFEM3D is a reference application for supercomputer benchmarking thanks to its good scaling capabilities. It supports asynchronous MPI communications as well as OpenCL and CUDA GPU acceleration. It shows strong scaling up to 896 GPUs and more than 21,675 CRAY XE nodes with 693,600 MPI ranks and sustained over 1 PF/s on the NCSA BlueWaters petascale system. In 2010, its multi-GPU port won the French Bull-Fourier supercomputing prize.

2.8.2 Code version

We used the official development version of SPECFEM3D Globe 7.0 of March 10th, 2015 (Git commit #a717e94).

2.8.3 Problem sizes

SPECFEM3D was configured to simulate a regional-scale Greek earthquake (regional_Greece_small). The accuracy of the simulation (SPECFEM3D `NEX`, controlling the number of elements at the surface along the two sides of the first chunk) varied between 64 and 1008 (the higher the more complex). The NEX number must be multiple of 16.

The number of processor cores (SPECFEM3D `NPROC`) that run the application depends of the `NEX` parameter. It must be 8 * multiple of `NEX`. We successfully ran the application with the following configurations:

NEX	Number of tasks
64	1, 4, 16, 64
128	1, 4, 16, 64, 256
256	1, 4, 16, 64, 256, 1024

For each of these configurations, we activated one or two CPU cores per node, and enabled or not the usage of the GPGPU accelerator.

Overall, we captured 45 valid execution results. Execution with bigger problem sizes or higher number of tasks could not run to completion, or their exhibited clearly invalid performance measurements (e.g., nodes over-consuming, which implies that the CPU was used by other processes, or under-consuming, which means they were running too slow and stained the entire performance and power profile).

2.8.4 Weak Scaling

The following figures present the weak scaling graphs of SPECFEM3D results. Figures 2.8.1 – 2.8.4 present the weak scaling graphs of SPECFEM3D results, on the four different hardware configurations: with one or two CPU cores and with or without GPU usage. The reference time for a given NEX problem size is the largest time-step (i.e., poorest performances) obtained on one CPU core, without using the GPU.

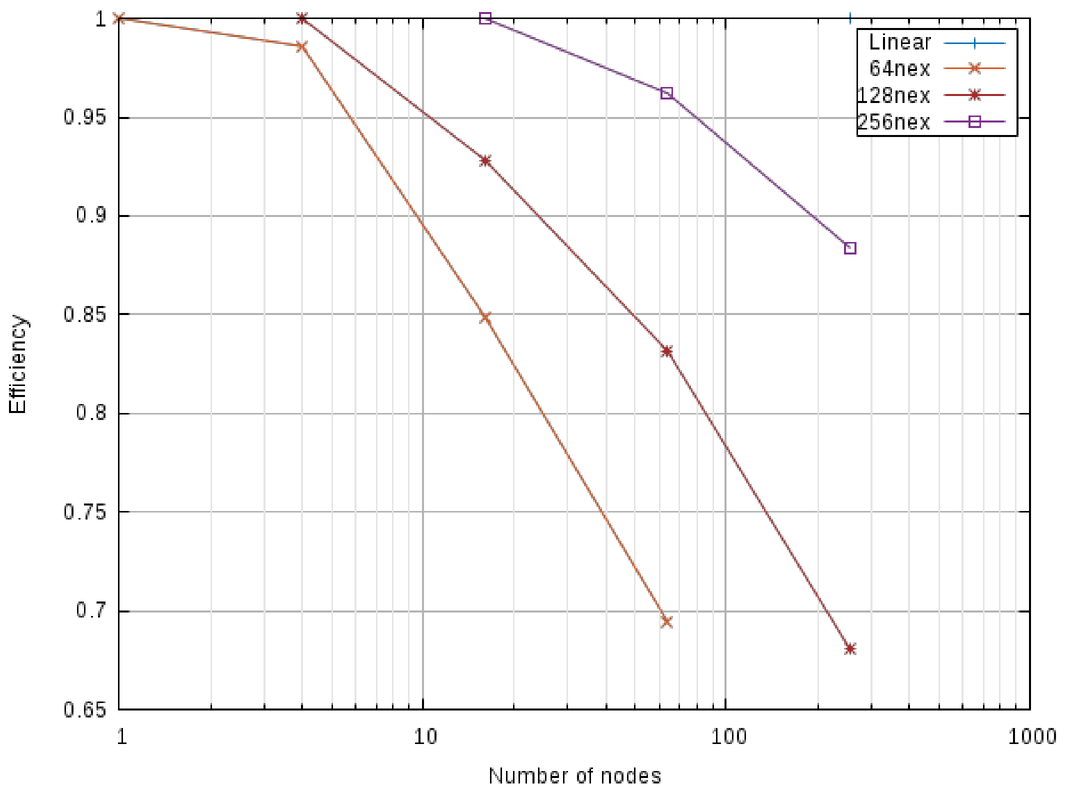

Figure 2.8.1 Specfem3D weak-scaling chart with 1 core and GPU

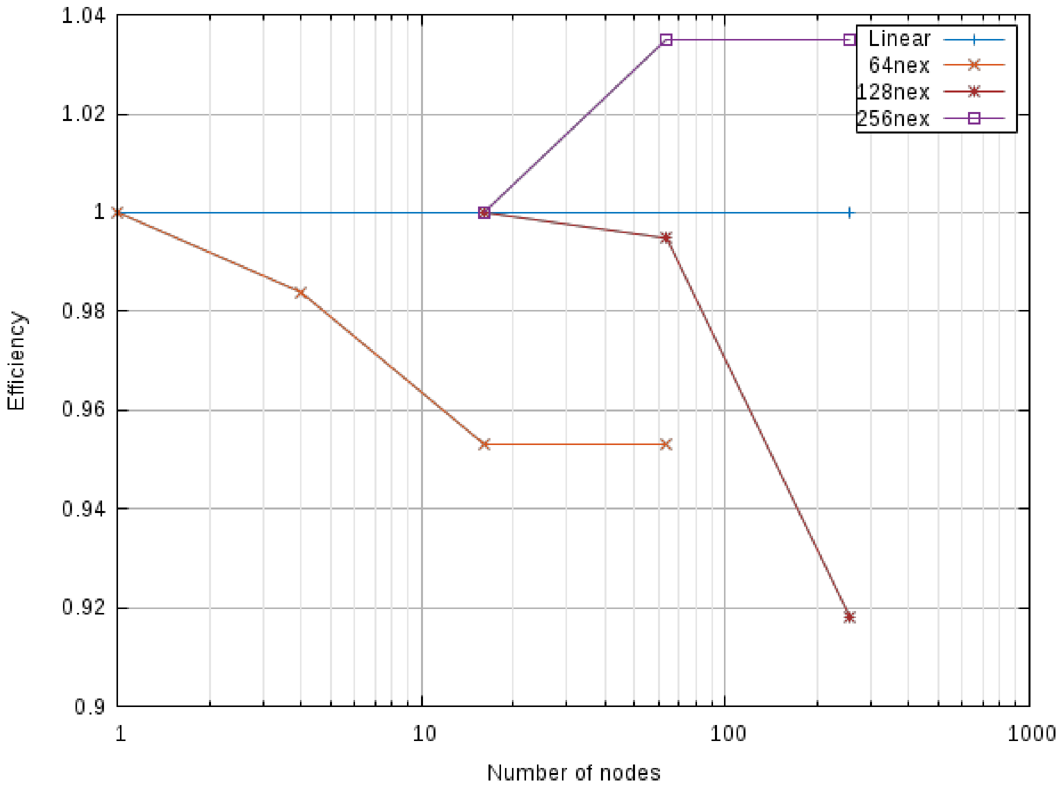

Figure 2.8.2 Specfem3D weak-scaling chart with 1 core and no GPU

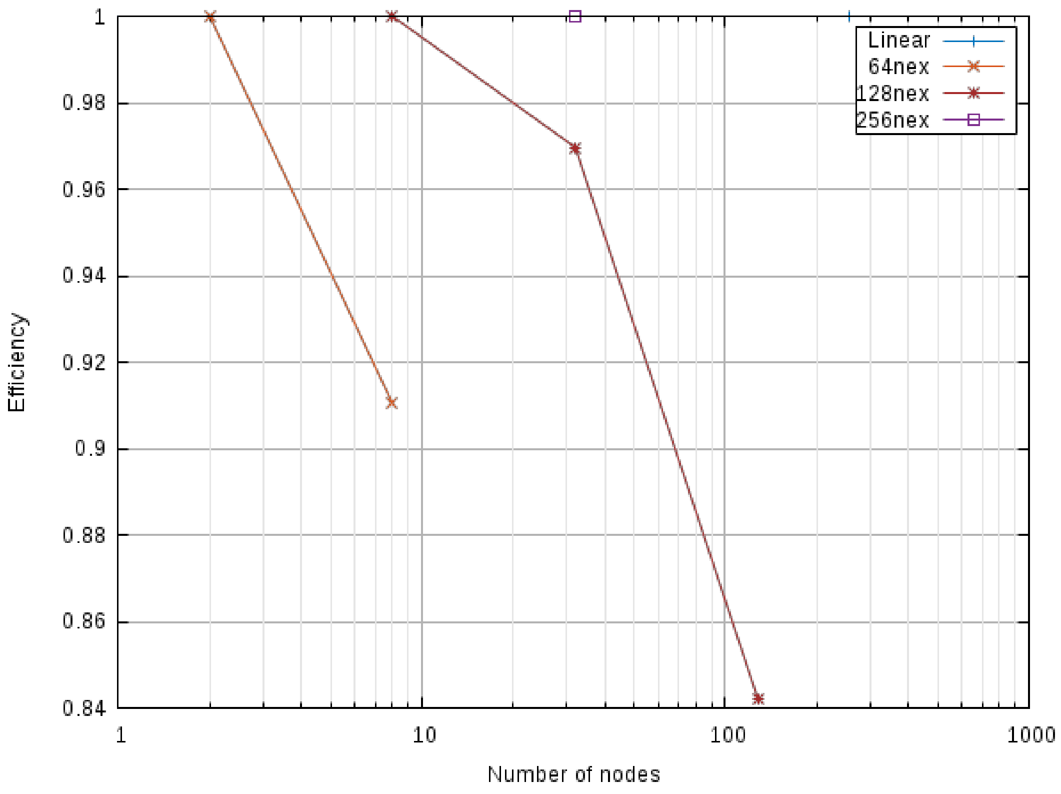

Figure 2.8.3 Specfem3D weak-scaling chart with 2 cores and GPU

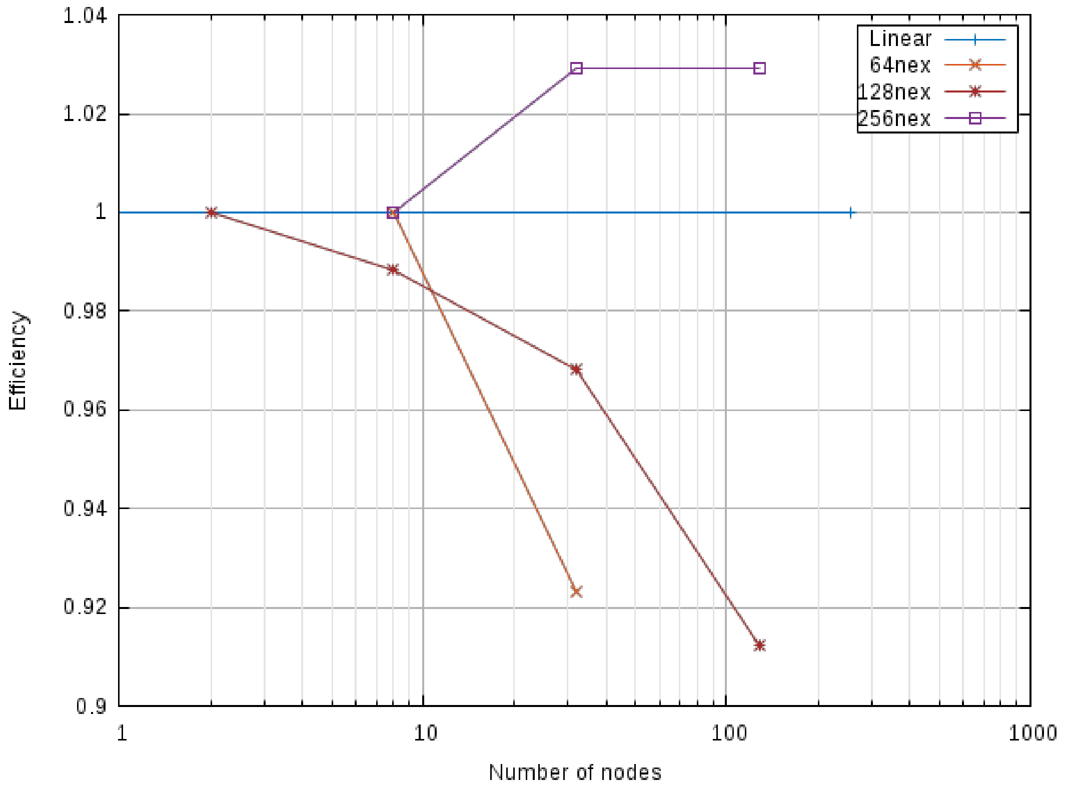

Figure 2.8.4 Specfem3D weak-scaling chart with 2 cores and no GPU

2.8.5 Strong Scaling

Figures 2.8.5 – 2.8.7 present the strong scaling graphs of SPECFEM3D results. Each figure corresponds to a SPECFEM3D problem size: 64, 128, 256 NEX (we could not collect enough data to plot 512 NEX because of the instability of the cluster and its filesystem). The reference time for a given NEX problem size is the largest time-step (i.e., poorest performances) obtained on one CPU core, without using the GPU.

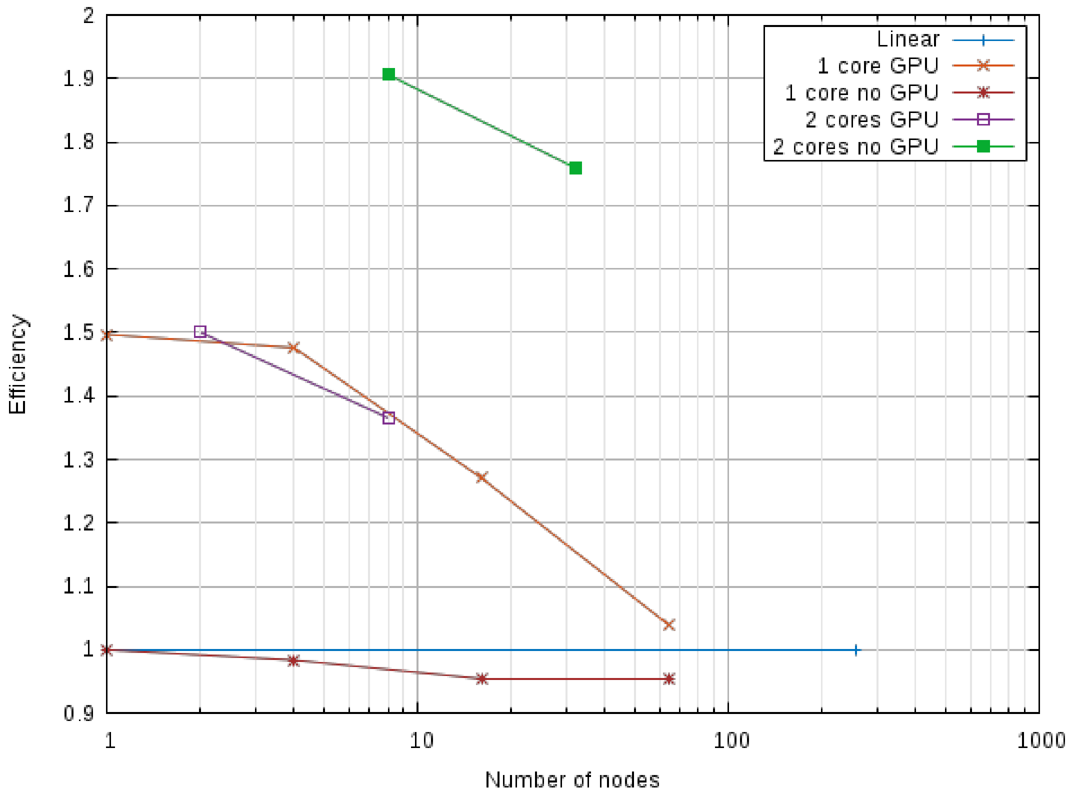

Figure 2.8.5 Specfem3D strong-scaling chart with 64-NEX problem size

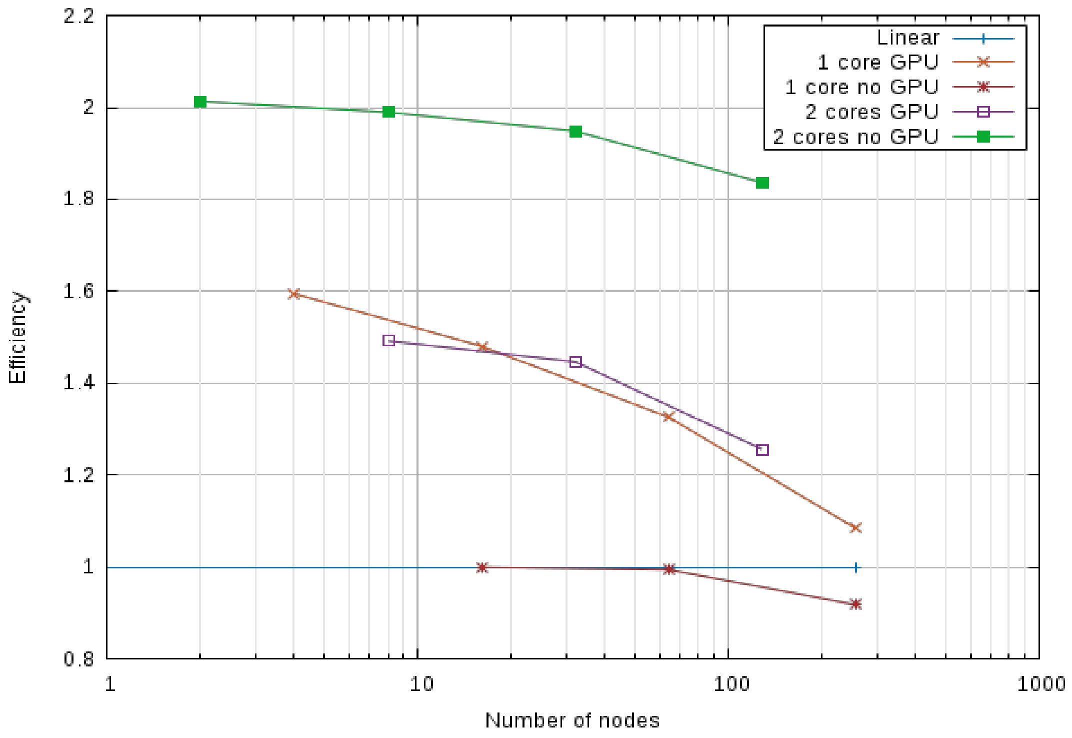

Figure 2.8.6 Specfem3D strong-scaling chart with 128-NEX problem size

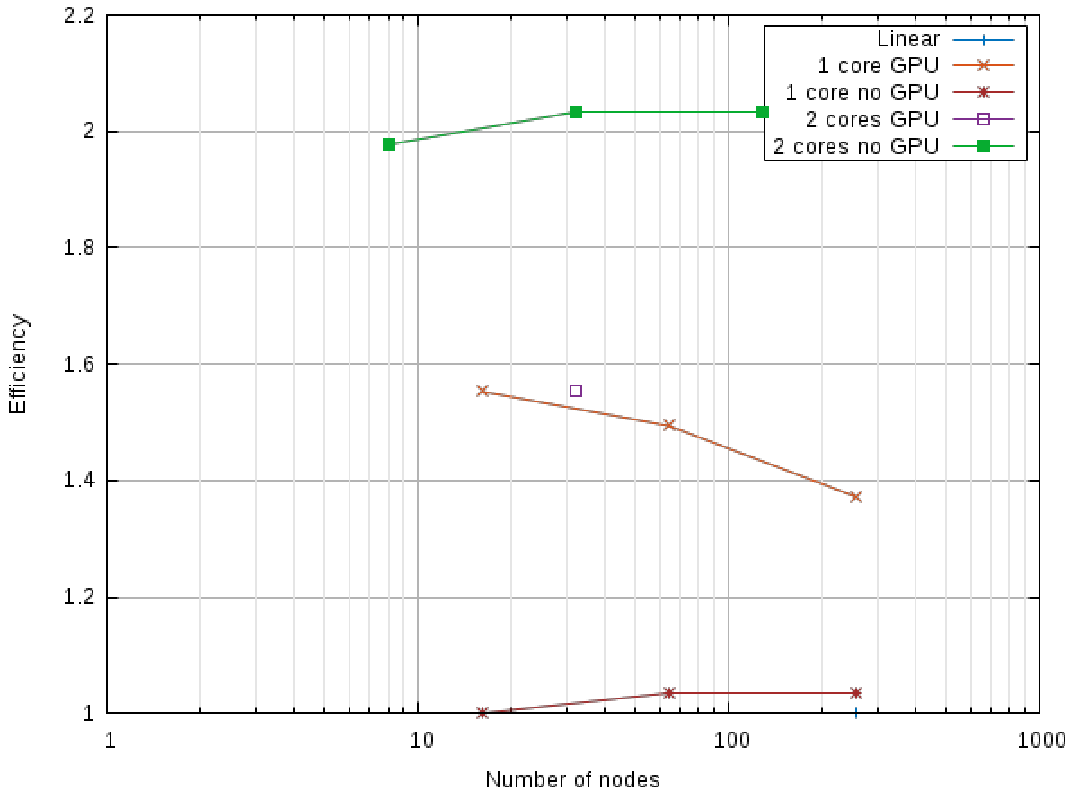

Figure 2.8.7 Specfem3D strong-scaling chart with 256-NEX problem-size

With relative order of the curves, which is consistent over the different problem sizes, we can understand that the one-CPU-core configuration offers the lowest performance, and the two-CPU-core configuration offers the best. When activating the GPU, using one or two CPU cores lead to the same performances. This level of performance is better than one-CPU core, and lower than two-CPU cores.

The CPU-only curves appear to scale well while increasing the number of nodes, however the GPU configurations tend to collapse when using more than 16 nodes, for the smallest problem sizes (64 and 128 NEX).

2.8.6 Energy profiling

Figures 2.8.8 – 2.8.12 present an example of the energy consumption reported for an idle run, and then, on the four hardware configurations for a given SPECFEM3D problem size (128 NEX). We can notice two spikes during the first seconds of the profiling. They correspond to our calibration preamble code. They indicate the time-scale (a spike lasts 10 seconds) and the power scale (first one core is used intensively, then two cores). After that, and similarly at the end of the simulation, the power consumption becomes irregular during a few seconds. That corresponds to the application set-up and tear-down. We did not keep these measurements in the following analyses.

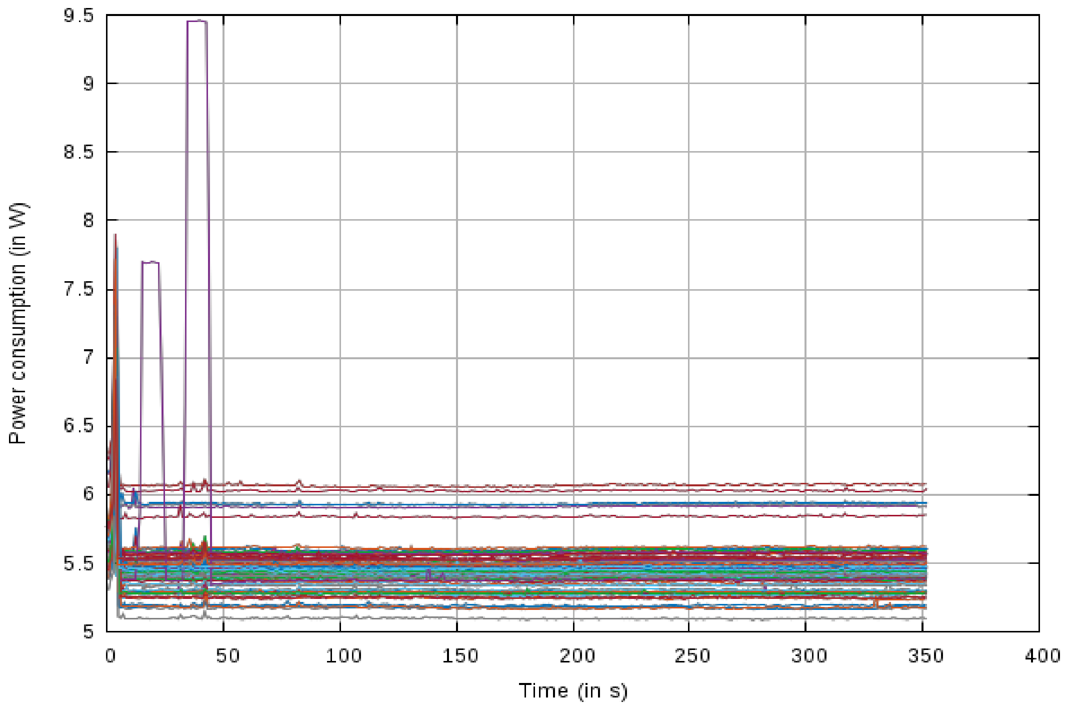

Figure 2.8.8 Specfem3D Energy consumptionon Mont-Blanc cluster for an idle run.

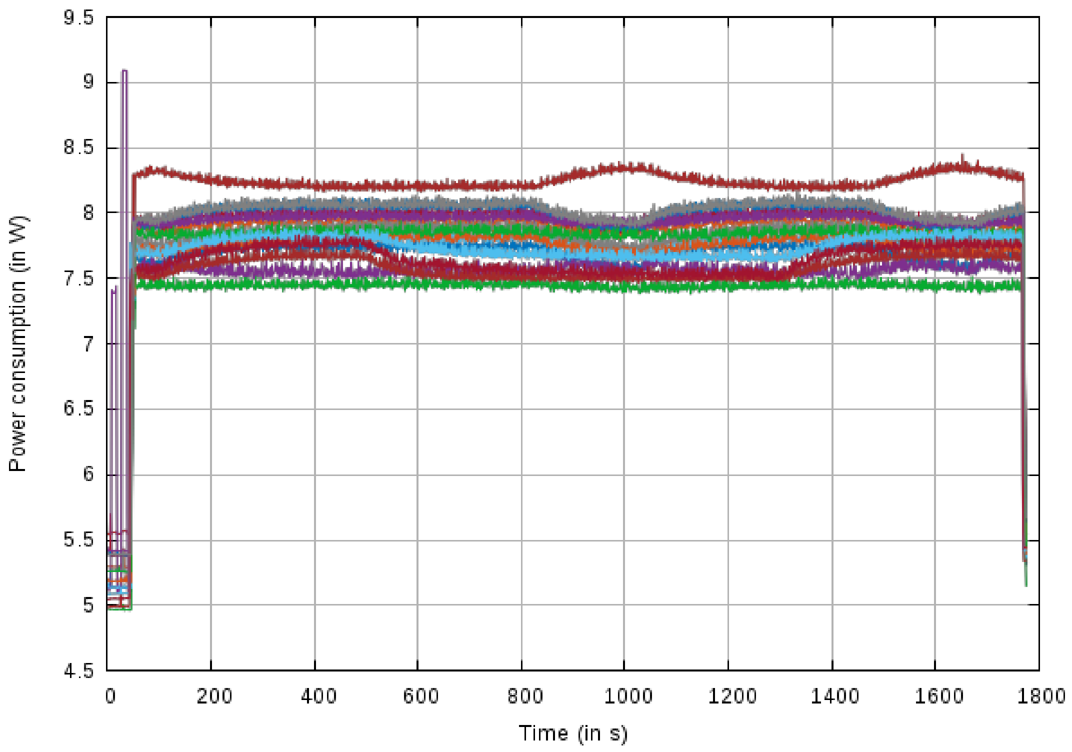

Figure 2.8.9 Specfem3D Energy consumption on Mont-Blanc cluster using 1 core per node

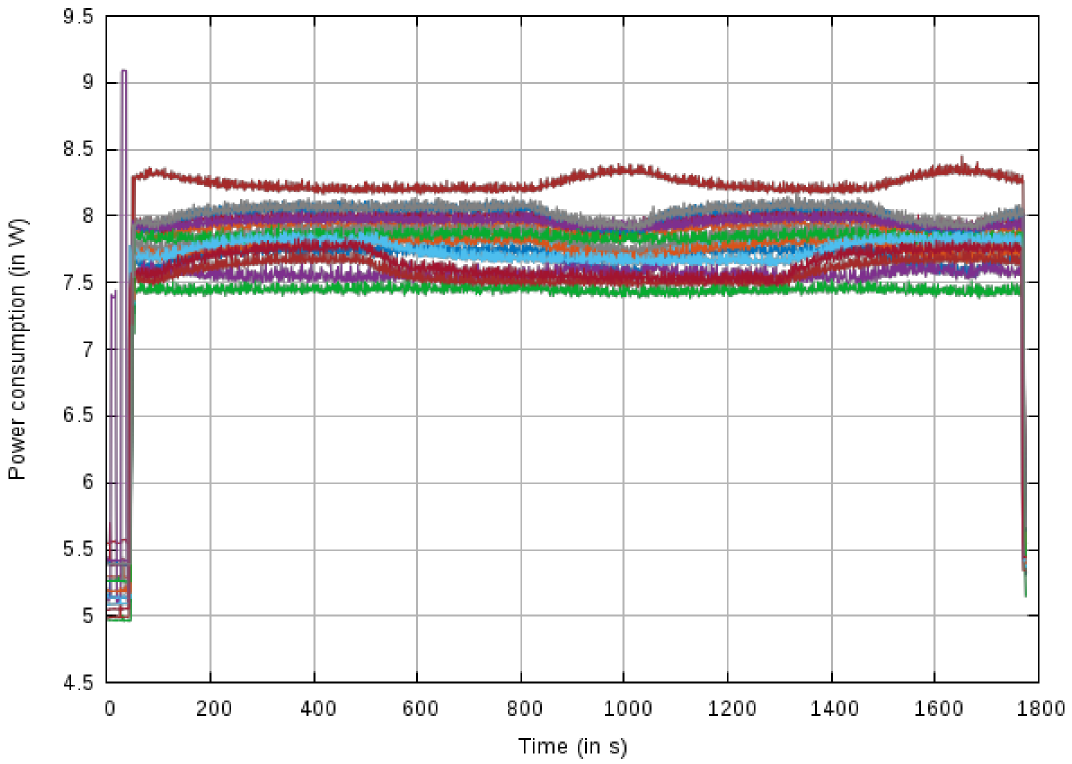

Figure 2.8.10 Specfem3D Energy consumption on Mont-Blanc cluster using 2 cores per node

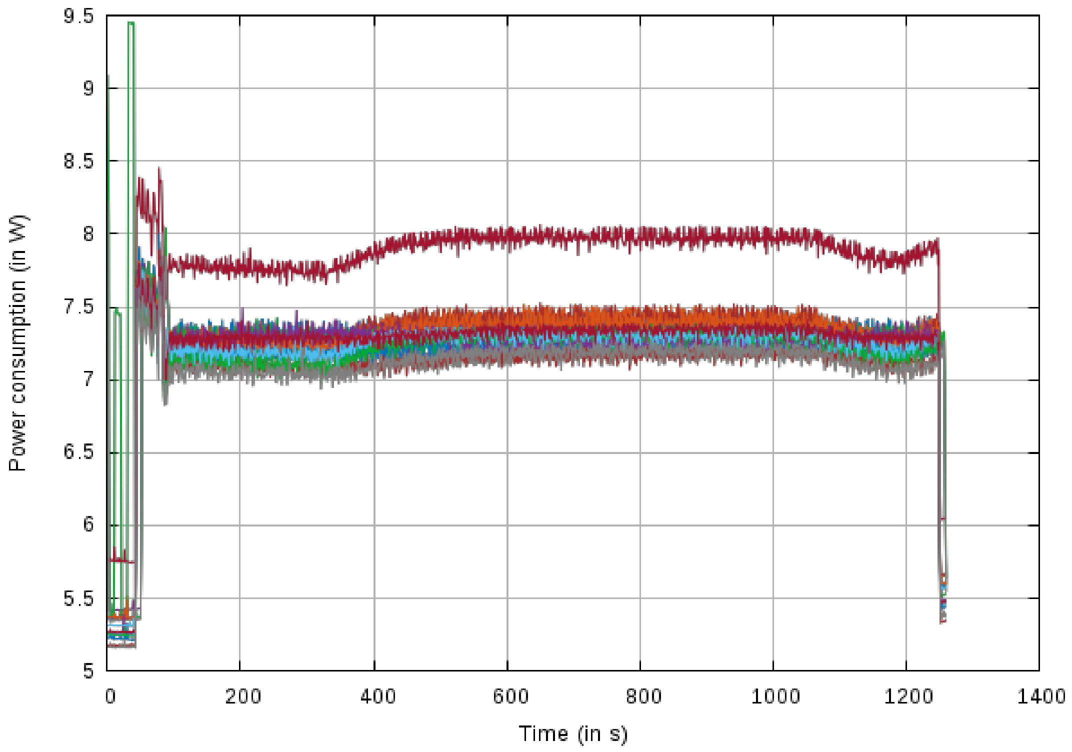

Figure 2.8.11 Specfem3D Energy consumption on Mont-Blanc Cluster using 1 core and a GPU per node

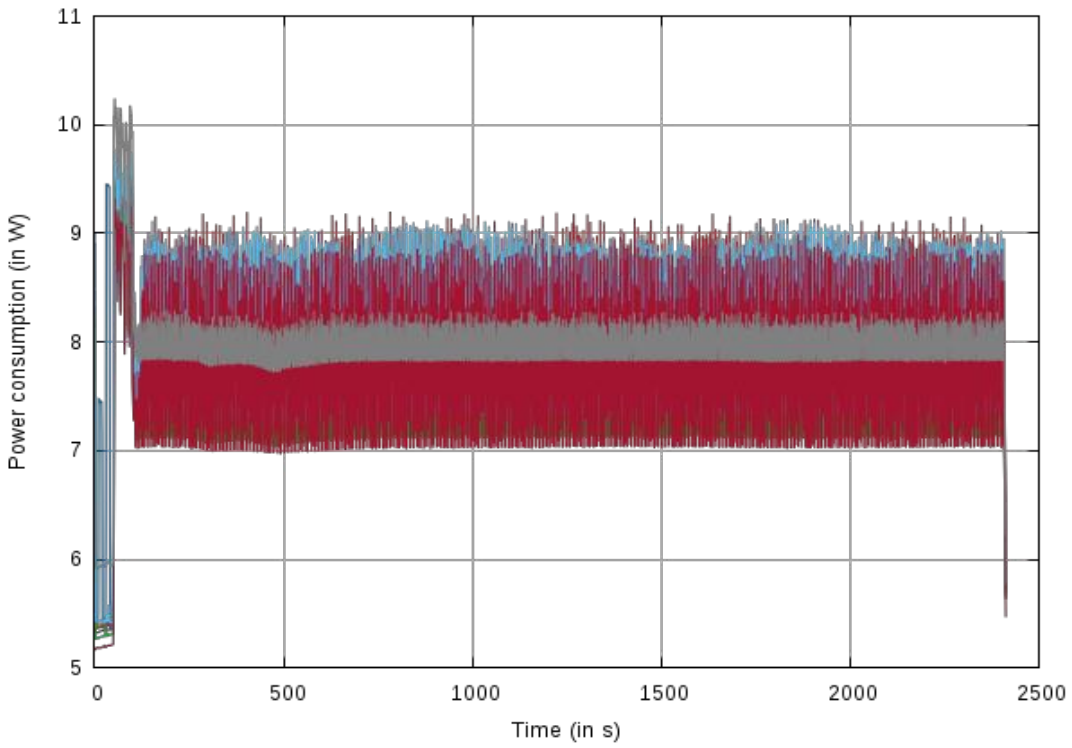

Figure 2.8.12 Specfem3D Energy consumption on Mont-Blanc cluster using 2 cores and a GPU per node

We can notice in Figure 2.8.12 that when the GPU is used in combination with two CPU cores, the power consumption varies heavily between 7W and 9W. The other hardware configurations are more regular. We believe that this is due to the sharing of the GPU between the two CPU-cores. In this situation, the GPU driver will have to wait more often for a GPU answer, and a busy-wait loop may explain the consumption spikes.

The Figures below illustrate the consumption variability for the different hardware configurations, with the instant consumption of one of the nodes plotted over the time.

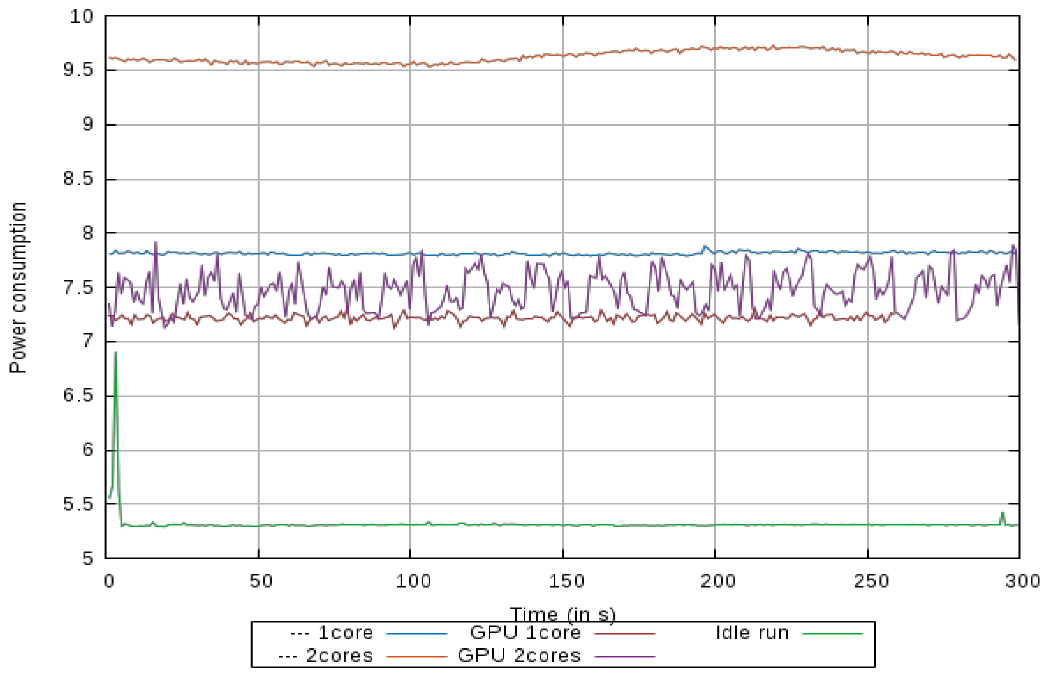

Figure 2.8.13 Specfem3D Energy consumption for each hardware configuration

Finally, the following figures present the power efficiency (power consumption per SPECFEM3D time-step) over the different problem sizes (except 512 NEX for which we could not collect enough data), and altogether.

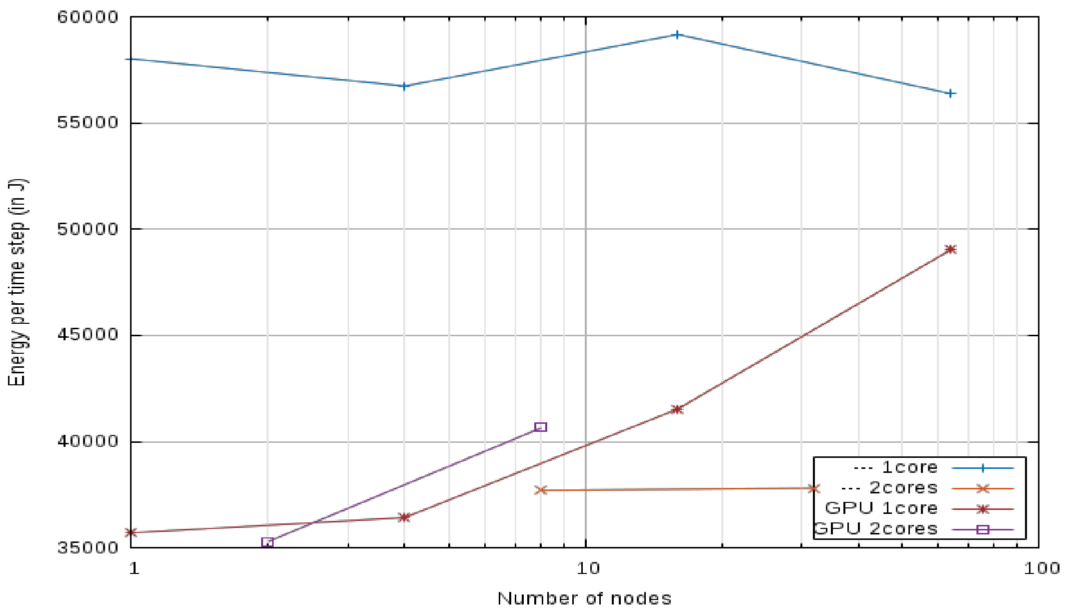

Figure 2.8.14 Specfem3D Power efficiency on Mont-Blanc cluster for the 64 NEX problem size

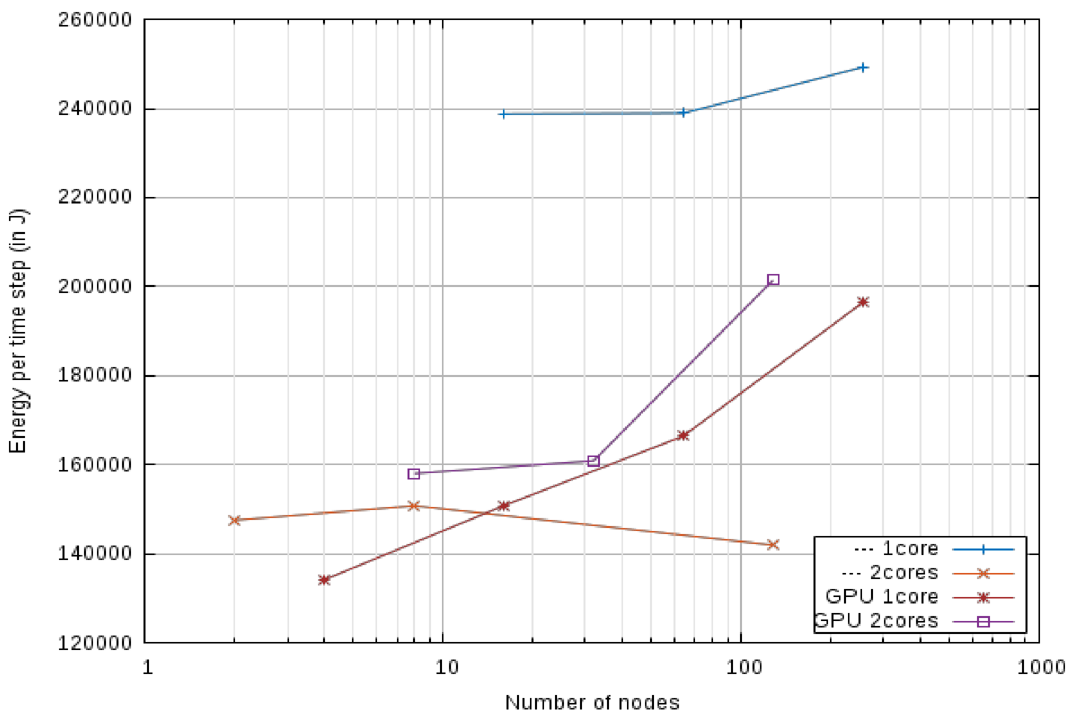

Figure 2.8.15 Specfem3D Power efficiency on Mont-Blanc cluster for the 128NEX problem size

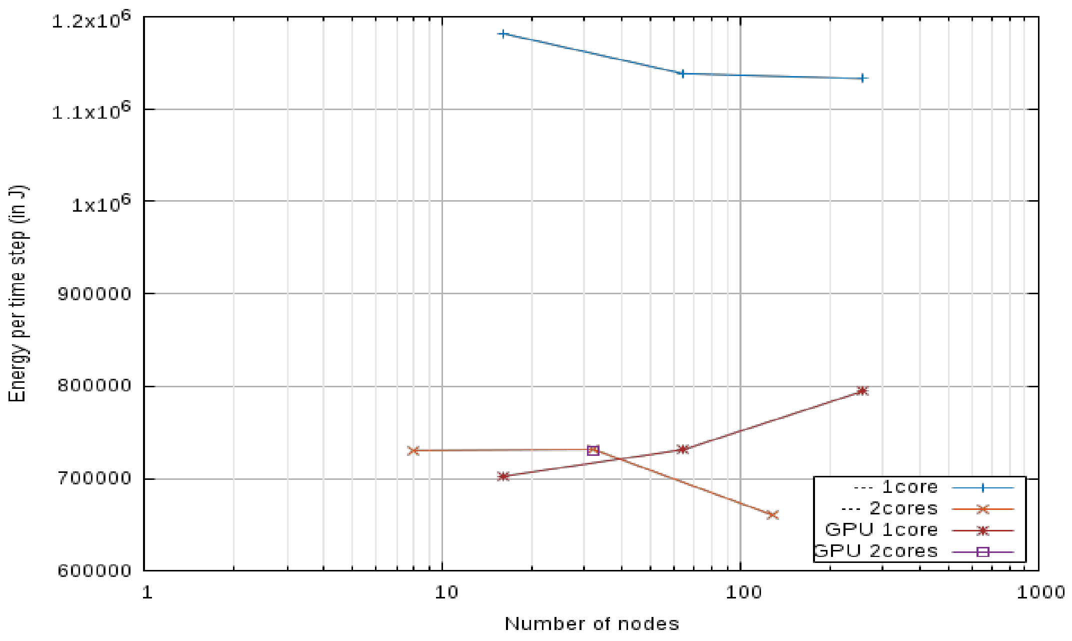

Figure 2.8.16 Specfem3D Power efficiency on Mont-Blanc Cluster for the 256NEX problem size

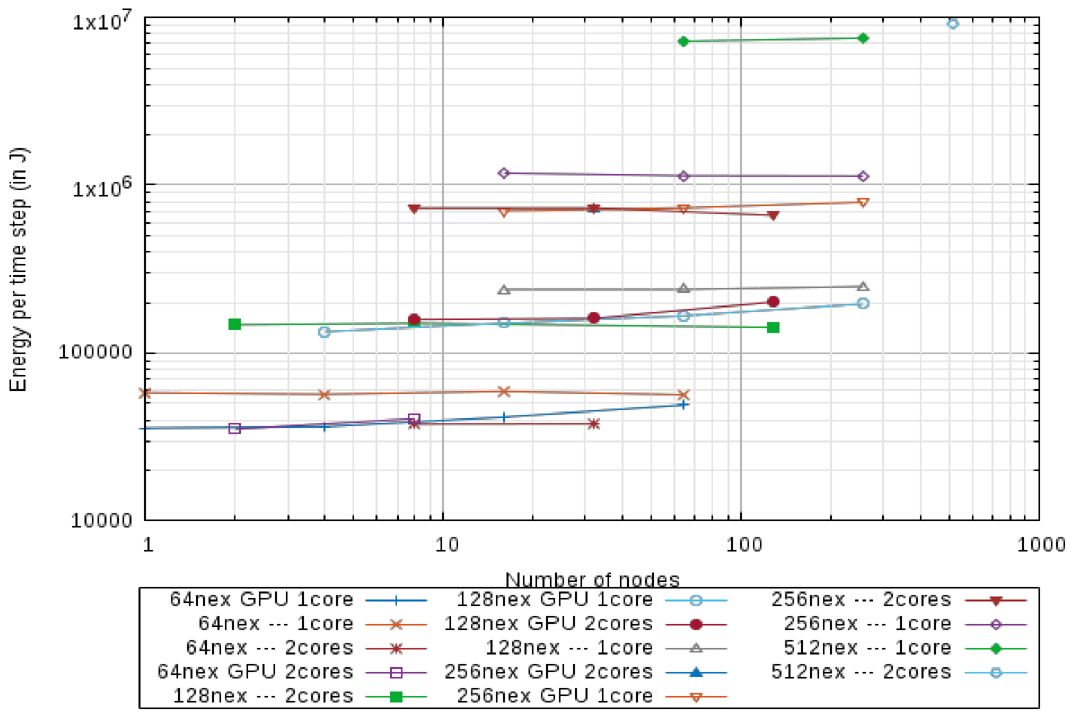

Figure 2.8.17 Specfem3D Power efficiency

2.8.7 Report on scaling and energy profiling

We can see in Specfem3D strong-scaling charts that the CPU benchmarks have a good strong scaling. The scaling of the GPU version is not as good. We suspect that the computation-over-communication overlapping may not be correctly handled by Mont-Blanc cluster's GPU drivers, which would explain the poorer performance obtained with the largest executions.

Regarding the energy consumption, a good strong scaling should keep the energy cost per iteration constant. On the power-efficiency charts Figures 2.8.14 – 2.8.17, we can see that this is only the case with CPU benchmarks, as well as with GPU runs with a low node count. However, the GPU consumption then rises because of its bad scaling and overcomes the CPU-only consumption.

Regarding the weak scaling Figures 2.8.1 – 2.8.4, we can see that the Mont-Blanc cluster prototype has been able to execute a wide range of SPECFEM3D simulation problem sizes, with a 200-ratio between the smallest and the largest problems. The charts show that the execution scales well with the increasing problem sizes.

On the energy consumption side, we can notice that the energy consumption of each node is different. Figures 2.8.18 – 2.8.22 below show the distribution of these consumptions, first for the idle run, then for SPECFEM3D runs with 256 NEX / 256 tasks for 1 and 2 cores, and 1 core + GPU, and 128NEX / 256 tasks for 2 cores + GPU.

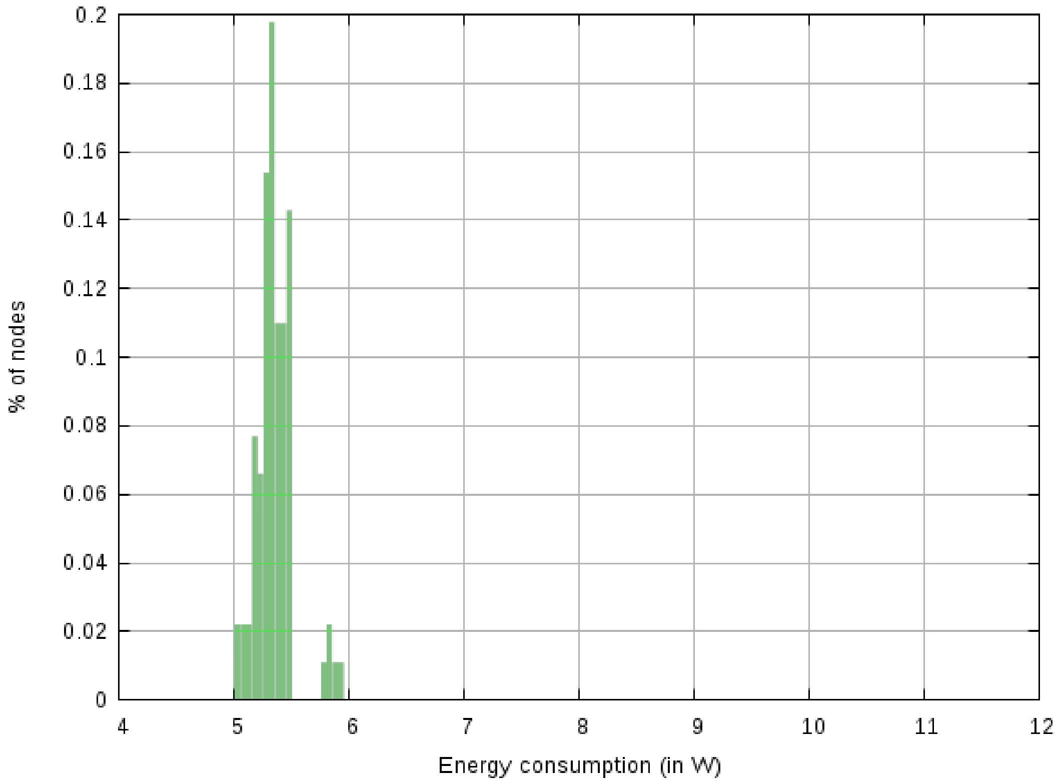

Figure 2.8.18 Specfem3D Node consumption for idle run

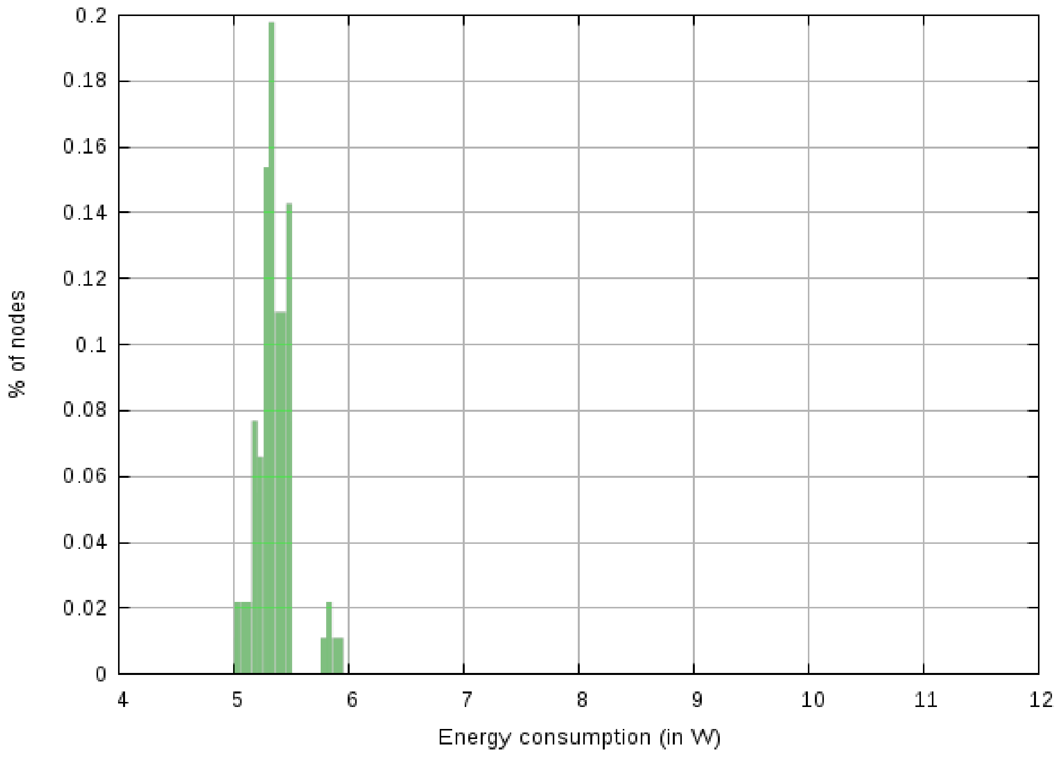

Figure 2.8.19 Specfem3D node consumption 1 core/node

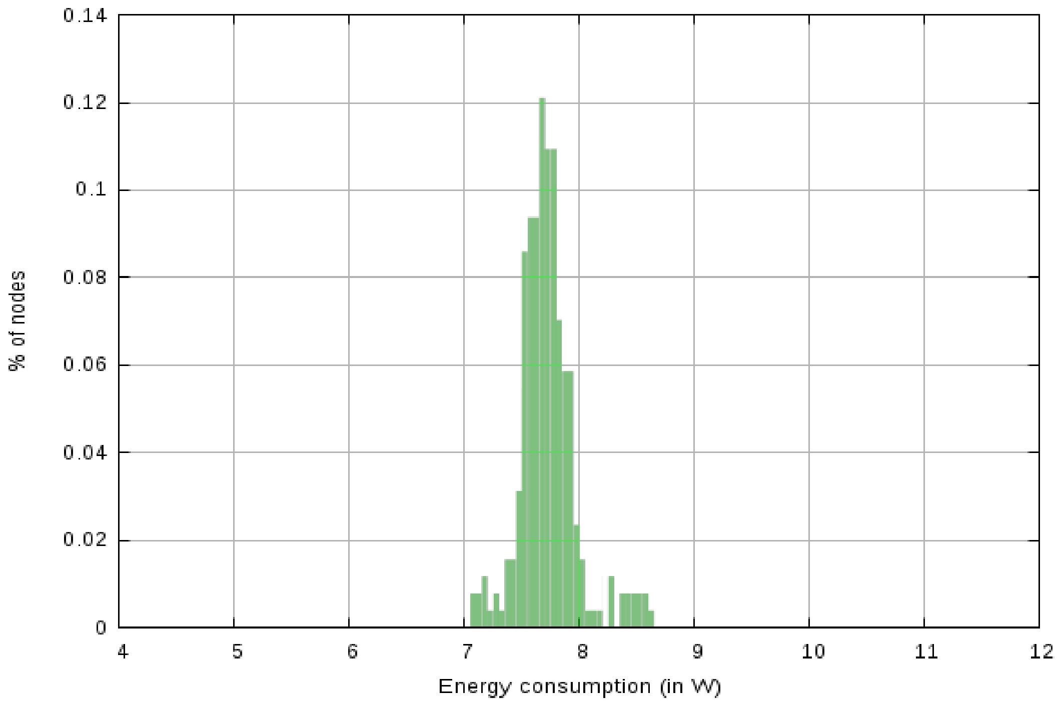

Figure 2.8.20 Specfem3D Node consumption 2 cores/node

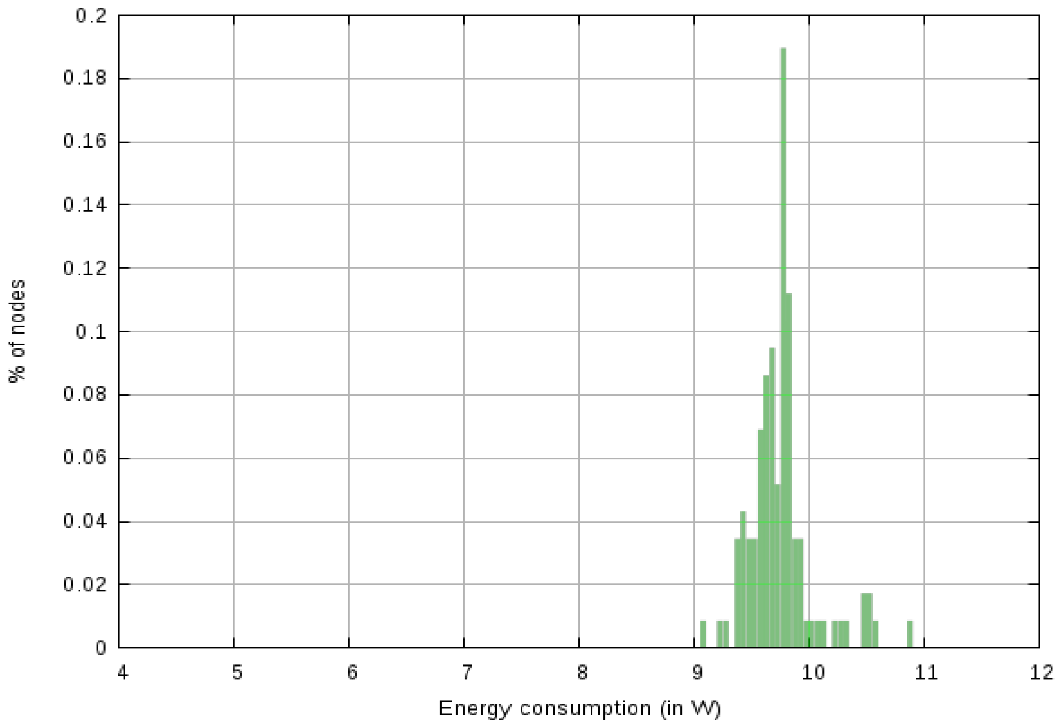

Figure 2.8.21 Specfem3D Node consumption 1core- 1 GPU / node

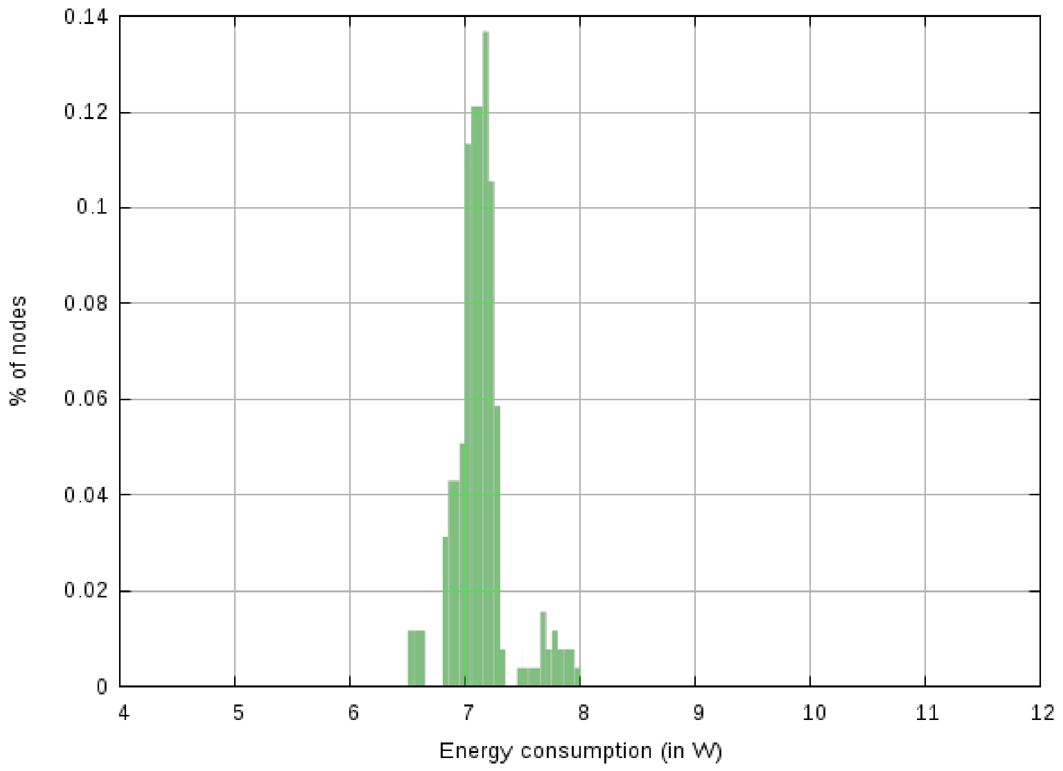

Figure 2.8.22 Specfem3D Node consumption 2 cores – 1 GPU / node

We can also see that all the nodes are not equal, but rather have a Gaussian distribution. The energy consumptions during SPECFEM3D executions have a Gaussian distribution. Over the time, they are quite regular, except when two CPU cores are used with the GPU.

The average consumptions are as follows:

Configuration	Average consumption
Idle	Baseline = 5.44W
1 core	Baseline + 2.46W = 7.80W
2 cores	Baseline + 4.38W = 9.82W
1 core + GPU	Baseline + 1.78W = 7.21W
2 cores + GPU	Baseline + 2.85W = 8.29W

2.8.8 Synthesis & Best practices

The Mont-Blanc cluster prototype can execute SPECFEM3D without major problem and scales well, with the real full-size application. As part of the Mont-Blanc project, the OpenCL GPU port of SPECFEM3D has been developed, contributed and incorporated into the development branch of SPECFEM3D, as well as the different optimization patches required to run SPECFEM3D on the Mont-Blanc cluster.

With GPU acceleration, executions with one CPU-core are more efficient that with two CPU-cores. Both configurations achieve similar performances, but the 1-CPU-core version has a lower consumption rate.

Without GPU acceleration, it is more efficient to use two CPU cores: it outperforms the single-core version with a lower consumption rate.

Overall, the dual-core configuration is better than the single CPU-core plus GPU version because of the poor GPU scaling. However, this is not true with small node-count size (under 4 for 64 and 128 NEX or under 16 for 256 NEX) where the GPU remains more efficient.