12 Benchmarking with SOFI3D

A slightly modified version of the example problem mentioned above can be used for benchmarking. Generally, we can on one hand test the scaling behavior of the SOFI3D modeling code and on the other hand - if the SOFI3D scaling behavior is known - the cluster computer (hardware + software installation) itself. Benchmarking can be performed assuming two different strategies: *speed-up* or *scale-up*. For the speed-up scenario we keep the modeling domain constant (NX=NY=NZ=512 gridpoints) and increase the number of cores (PEs) for the simulation. This way we assume that we have a fixed modeling problem and just want to get the results faster. In theory, if we double the number of cores (PEs) we reduce the total runtime by half. By plotting the runtime

$$t = \frac{t_8}{t_N} \tag{25}$$

with the total runtimes using 8 cores t_8, the total runtime t_N using N cores (N = NPROCX $\times$ NPROCY $\times$ NPROCZ) versus the number of cores N, we can theoretically gain a linear speed-up. This means, every time we double the number of cores (e.g. 16, 32, 64, 128, ...) we speed-up the calculation by factor 2. If the speed-up is not linear this can be caused by two different reasons. If the number of grid points per PE is decreasing due to increasing number of PEs, caching effects (storing data in the core cache instead of the memory) can speed-up the update of the wave field (over-linear behavior). If the amount of data that has to be exchanged is increasing due to the increasing number of cores and the communication interface (network interconnect) can not handle that increased amount of exchange, the total runtime can be overpowered by the exchange times and thus the speed-up is below linear. The latter case will most likely occur at some point.

The speed-up benchmarks performed on the several super computers are displayed in Figure 20. For more information on the hardware architecture and further basic specifications of each cluster computer see Table 5. From Figure 20 we see that SOFI3D scales both at the Juropa and the Hermit cluster excellent. The speed-up is close to the linear curve up to 4096 cores (see Figure 20a). In contrast the IC2 speed-up decreases significantly with increasing number of cores. At some point above 1000 cores, the speed-up is even almost 1, i.e. further increasing the number of cores does not significantly speed-up the simulation. The pure speed-up benchmark does not tell the absolute performance of SOFI3D, this is displayed in Figure 20b where only the total computation times are plotted. It can be seen that these total times can be quite different, especially at large number of cores which is usually due to the different node interconnect (network connection). Depending on the hardware architecture the massive simultaneous exchange of data and, thus, the time needed for exchange of data, can significantly dominate the total time resulting in a good or bad speed-up behavior.

In contrast to the speed-up benchmark, the scale-up benchmarks assume that we have a simulation that reaches the hardware limitations by a certain number of PEs. By increasing the modeling domain we might exceed the hardware resources (e.g. memory consumption). If we increase the number of PEs and increase the modeling domain such that the number of gridpoints per PE is constant (NX=NY=NZ=128 gridpoints per PE!), we can expect to have a constant runtime. Usually the runtime will increase with increasing number of PEs due to an increase in communication overhead, i.e., the update times will be constant but

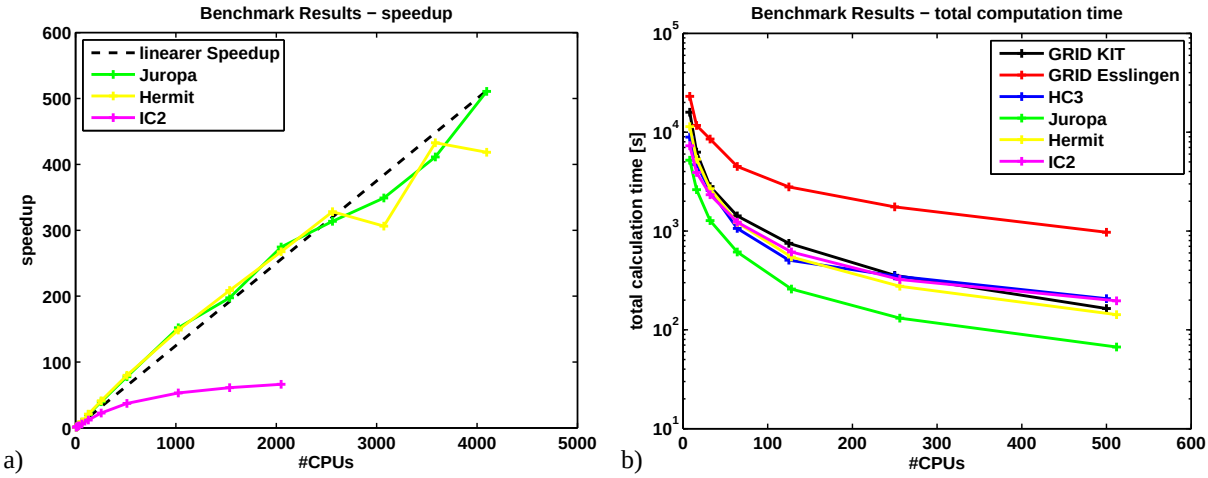

Figure 20: Speed-up benchmarking results performed on the several super computers: a) speed-up factors, b) total computation time.

the exchange times will increase. The SOFI3D scale-up behavior performed on several super computers (see Table 5) is displayed in Figure 21. As can be seen from the subfigures, the interpretation can be quite complicated. The general scale-up behavior of SOFI3D on both Juropa and Hermit is reasonably stable. Only at larger numbers of cores, the total computation at Hermit increases. However, each simulation at constant input parameters takes more than twice the time at the Hermit cluster which already have been shown in Figure 20b. In contrast, the IC2 behavior can be seen as a bad example. Instead of being constant, the total computation time steadily increases with the increasing number of cores. From Figure 21c we see that this is not due to the update times but due to a steady increase in the exchange times (see Figure 21c). Obviously the interconnect between each node is too slow or cannot handle massive simultaneous exchange of data very well. In the end these exchange times overpower the good scaling behavior of the update times and result in a bad scale-up behavior.

Alltogether, we believe that the scale-up test is more realistic and can be directly used to also compare the absolute performace of each cluster but it is generally more consuming. For both the scale-up and speed-up benchmark test, input files are prepared and located in the folder *par/in_and_out/benchmark*. The corresponding source-location files are located in the folder *par/sources/benchmark*. In order to ensure comparable benchmark scenarios especially for the scale-up test, each PE has been assigned a source. This way, every PE has to update the wavefield right from the start and not just update with zero values until the wavefield propagates in each PE. As a result, the source file for the scale-up considering 4096 cores, 4096 source locations are defined, too.

For convenience, there is a shell-script located in the folder *par/* that can be used to perform the benchmarks:

```
-bash-2.05b$~/sofi3D/par> ./startBENCH.sh 8
```

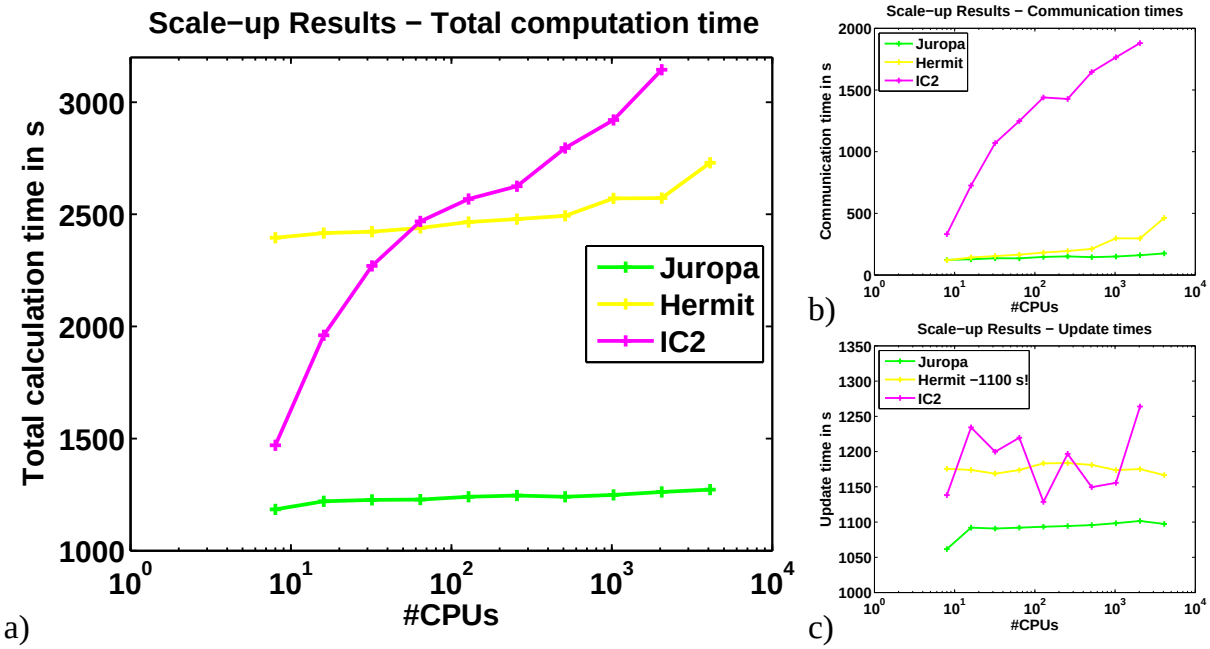

Figure 21: Scale-up benchmarking results performed on several super computers: a) total computation times, b) communication times, c) update times.

The script *startBENCH.sh* will have to be modified according to the job-submission syntax on each cluster computer. Table 6 can be used to estimate the range of total runtime that has to be specified for the job-submission. The total runtime of the scale-up benchmark can range from 1200 s (Juropa) to 3000 s (IC2).

Name	web address
Juropa	http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUROPA/JUROPA_node.html
Hermit	http://www.hlrs.de/systems/platforms/cray-xe6-hermit/
IC2	http://www.scc.kit.edu/dienste/ic2.php
GRID KIT	http://www.scc.kit.edu/dienste/7349.php
GRID Esslingen	http://www.hs-esslingen.de/de/hochschule/fakultaeten/informationstechnik/it-forschung/bwgrid.html
HC3	http://www.scc.kit.edu/dienste/hc3.php

Table 5: Web addresses for information on the architecture and hardware information of each super computer considered for the speed-up and scale-up benchmark.

Number of PEs Number of PEs	Allow simulation time in h (Juropa / Hermit)	Memory consumption per PE in MB
8	16:00:00 / 32:00:00	1500
16	09:00:00 / 15:00:00	750
32	05:00:00 / 08:00:00	400
64	03:00:00 / 03:30:00	210
128	01:00:00 / 01:45:00	120
256	00:45:00 / 01:00:00	70
512	00:30:00 / 00:45:00	35
1024	00:15:00 / 00:30:00	20
1536	00:10:00 / 00:20:00	15
2048	00:05:00 / 00:00:00	12
2560	00:05:00 / 00:10:00	10
3072	00:05:00 / 00:10:00	9
3584	00:05:00 / 00:05:00	8
4096	00:05:00 / 00:05:00	7

Table 6: Compilation of number of PEs, minimum expected total simulation times (gained from Juropa Cluster computer at Juelich and estimated for job-script file) and memory consumption (rounded up) for the speed-up benchmarking.