Hardware-Efficient Parallelized Optimization with COMSOL Multiphysics® and MATLAB®

Frommelt Thomas and Gutser Raphael
SGL Carbon GmbH, Technology & Innovation, Werner-von-Siemens Straße 18, 86405 Meitingen, Germany

COMSOL CONFERENCE ROTTERDAM2013

Introduction: Today, new processors provide increasing number of cores at rather constant clocking frequency. In sequential optimization algorithms, the forward model simulation is typically accelerated by multiple cores (shared-memory parallelization, SMP), which provides only limited speed-up and hardware efficiency. In this work, an alternative approach is presented with encouraging performance benefits.

Parallelization Benchmark: Platform specifications: Two Intel E5-2667 6-core CPUs, 64GB RAM allowing 12 parallel models, Windows 7, Comsol 4.3b.
As a key performance indicator the model runs per iteration are evaluated. The speed-up of single models with multiple cores by SMP is limited by the effort to split the problem and the slow remote access of cores onto the memory bank of the other socket [1]. Thus, the best SMP performance is achieved by occupying only one socket. Yet, simulating multiple models with single core provides almost ideal parallelization with a 4 times higher peak performance than SMP.

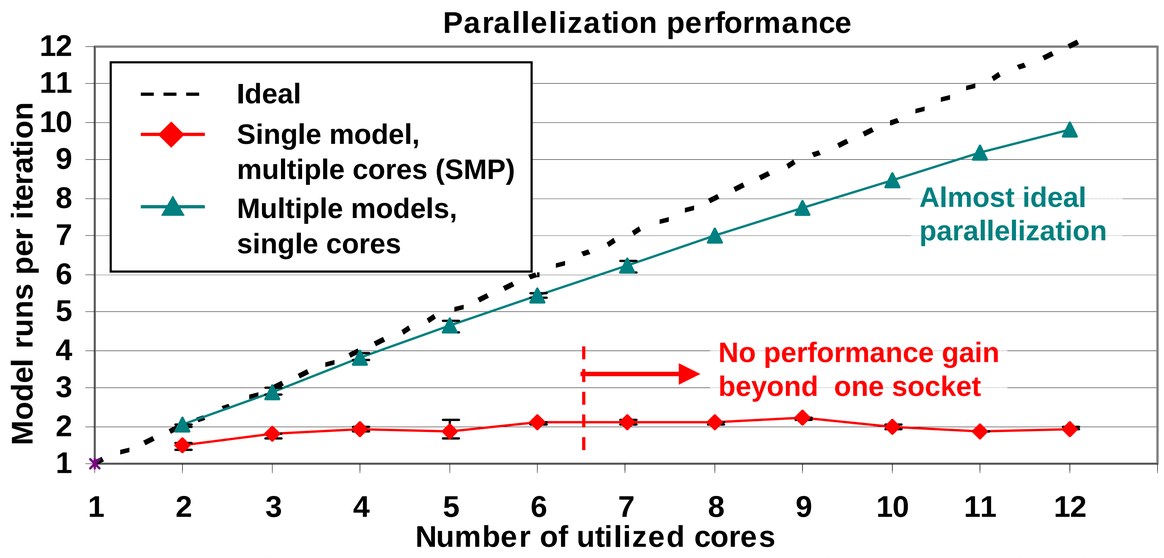

Figure 2. Performance comparison: Simulating multiple models single core allows to run more models per iteration than multi-core simulation of single models (SMP).

Latin hypercube optimizer LHSOpt: In each iteration, latin hypercube sampling determines a uniformly distributed set of designs in the window of the parameter space (Fig. 3a). The best design of an iteration determines the center point for the next iteration (Fig. 3b). A scheduler and several Comsol servers simulate multiple models with single cores, using one Comsol single user license.

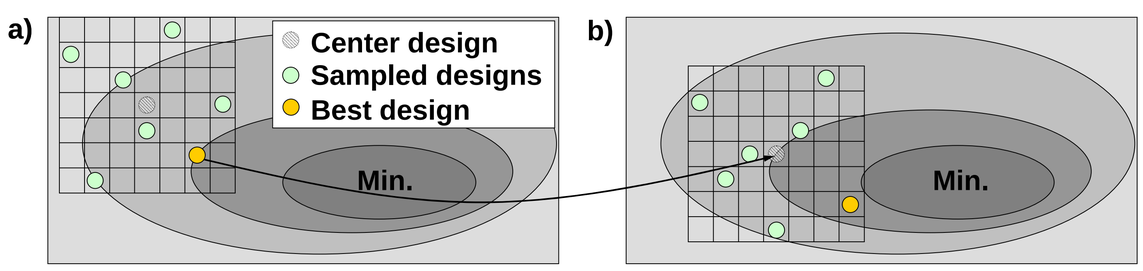

Figure 3. a) Latin hypercube sampling around a center design, b) best design is the center design of the next iteration.

Test Model System: In a transient electro-thermal model of an industrial graphitization furnace, part positions are optimized (Fig. 1).

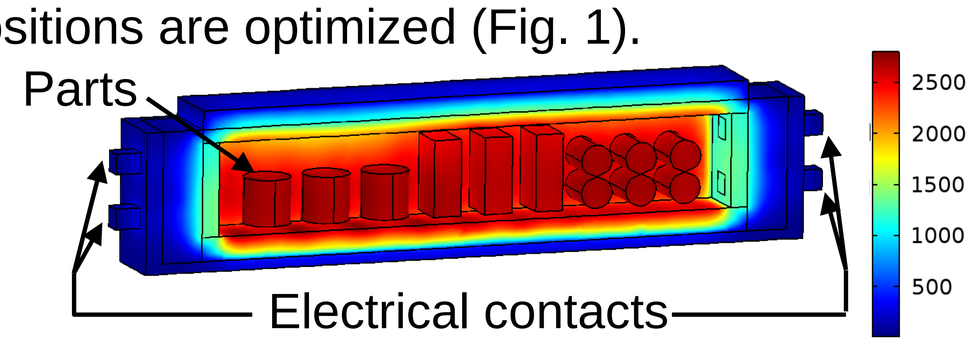

Figure 1. Temperature distribution in an industrial Acheson graphitization furnace.

Optimizer Benchmark: LHSOpt was benchmarked against Nelder-Mead Simplex fminsearch with SMP for 10 homgeneously distributed starting points in parameter space. The quality of the returned result is expressed in percent, with 100% as the global optimum. In comparison to fminsearch with SMP, Tab. 1 highlights that LHSOpt is faster (speed-up factor), more robust (convergence for all starting points) and reliably finds the optimum.

Starting Point	Fminsearch Duration (h)	Fminsearch Result	LHSOpt Duration (h)	LHSOpt Result	Speed-Up Factor
1	9.6	✗	2.5	100%	
2	22.0	✗	5.7	100%	
3	6.0	✗	6.8	100%	
4	10.9	100%	3.7	100%	2.97
5	9.0	✗	4.4	100%	
6	4.9	93%	3.6	100%	1.35
7	8.8	✗	2.5	100%	
8	5.4	97%	5.1	100%	1.05
9	5.3	100%	3.7	100%	1.43
10	3.9	100%	3.7	100%	1.04
✗: No convergence				Average:	1.57

Table 1. Benchmark results for fminsearch and LHSOpt

Conclusions: In our tests, the simple LHSOpt algorithm accesses between 1% and 75% of the hardware performance benefits for simulation of multiple models with single cores. In future, more sophisticated optimization algorithms can increase this fraction and reduce the observed variation of the speed-up factor.
Moreover, the implemented optimization workflow enables even a joint operation of several single user licenses on one optimization task. This approach is particularly interesting for the common situation of modeling teams with several existing single user licenses instead of a floating network license and a cluster for distributed computing.

References:

1. D. Molka, D. Hackenberg, R. Schöne, M. S. Müller: Memory Performance and Cache Coherency Effects on an Intel Nehalem Multiprocessor System, *18th International Conference on Parallel Architectures and Compilation Techniques* (2009)