SGL GROUP
THE CARBON COMPANY

COMSOL CONFERENCE ROTTERDAM2013

Hardware-Efficient Parallelized Optimization with Comsol Multiphysics® and Matlab®

Thomas Frommelt
SGL Group, Technology & Innovation

Comsol Conference 2013 – Rotterdam, 23rd-25th October 2013

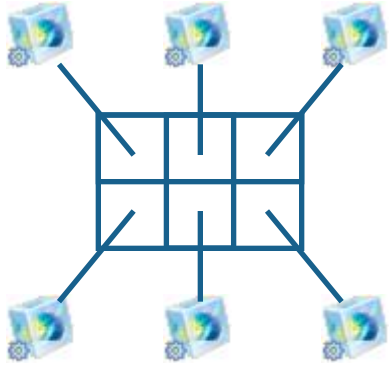
BROAD BASE. BEST SOLUTIONS.

Motivation
Efficient Optimization With State-Of-The-Art Hardware

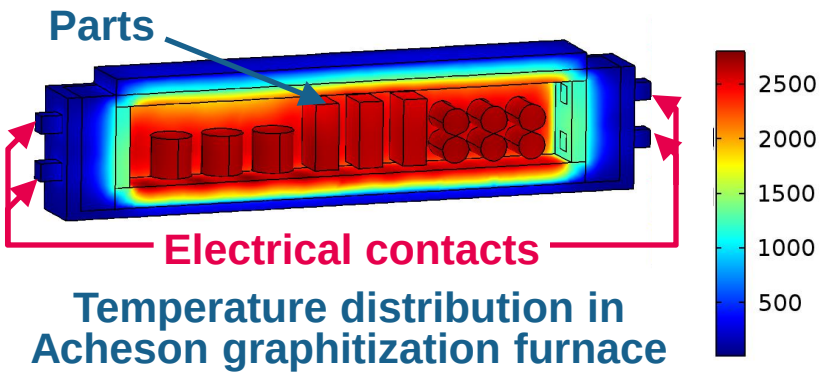
- Processor clocking frequency stagnates since years due to power dissipation
  ➔ Performance increase by more processor cores
- Optimization is high effort: Which approach efficiently uses many cores?
  - Single model with multiple cores (shared-memory parallelization)
    ➔ Normal sequential optimizers
  - Multiple models with single cores
    ➔ Optimizer must support simultaneous simulation of several models!

Single Model

Multiple Models



- Test system
  - Transient electro-thermal model to optimize positions of carbon parts in an industrial graphitization furnace
  - Hardware: 2x Intel I5 6-core CPUs, 64GB, Windows 7, Comsol 4.3b



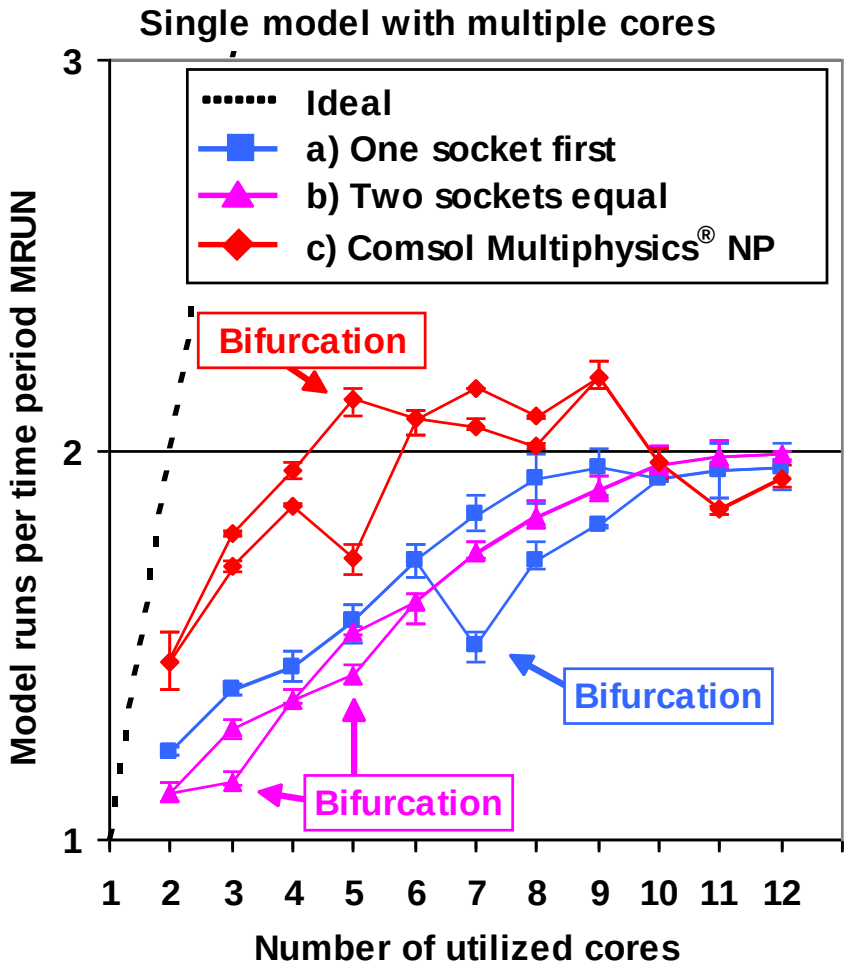
Temperature distribution in Acheson graphitization furnace

Parallelization Benchmark
Single Model Multiple Cores

- Indicator MRUN: Number of possible model simulations per iteration

Single model with multiple cores

- Several strategies:
  a) Use one CPU first, then the other
  b) Use both CPUs equally
  c) Automatic by NP option
- Parallelization performance by far not ideal
- Bifurcations detected ➔ No reliable performance for a setting
- Stagnation beyond 1 CPU
- Reasons: hardware architecture, memory allocation, parallelization overhead



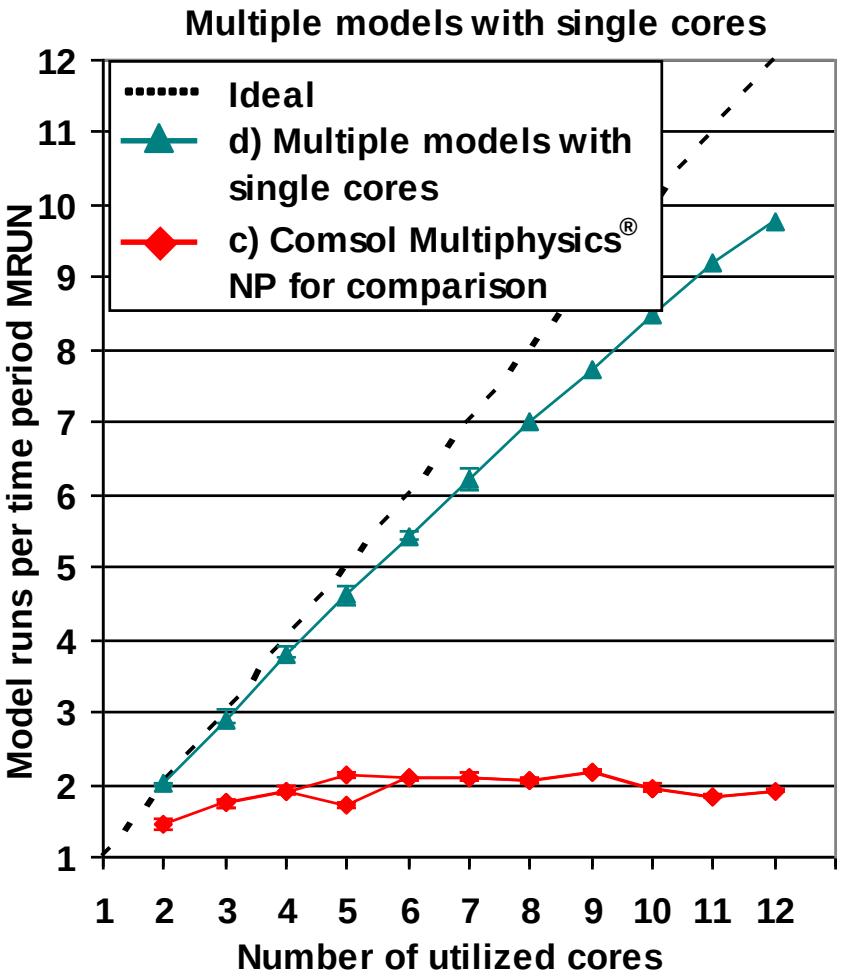
Parallelization Benchmark
Multiple Models Single Cores

- Indicator MRUN: Number of possible model simulations per iteration

Multiple models with single cores

- Almost ideal parallelization
- By far better MRUN than best single model result ➔ More than a factor 4 at full hardware utilization

➔ A parallelized optimizer has the potential for 400% speed-up by more efficient hardware utilization as compared to sequential optimization with single models

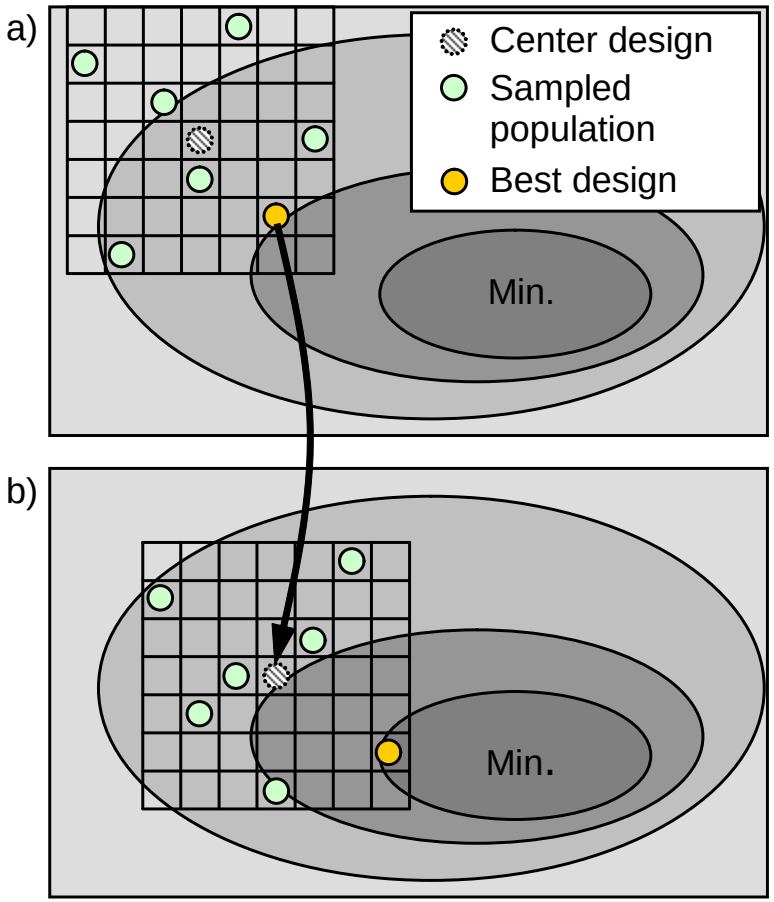


Optimization Benchmark
Investigated Optimizers

- **Fminsearch: Sequential Nelder-Mead Simplex algorithm in Matlab®**
- **LHSOpt: Developed parallelized optimizer based on latin hybercube sampling (implemented in Matlab®)**

  a) Samples parameter space around center point
  b) Best design is center point of next generation

  – **Parallelized optimization workflow working with single user license**



Principle of LHSOpt

Optimization Benchmark
Results

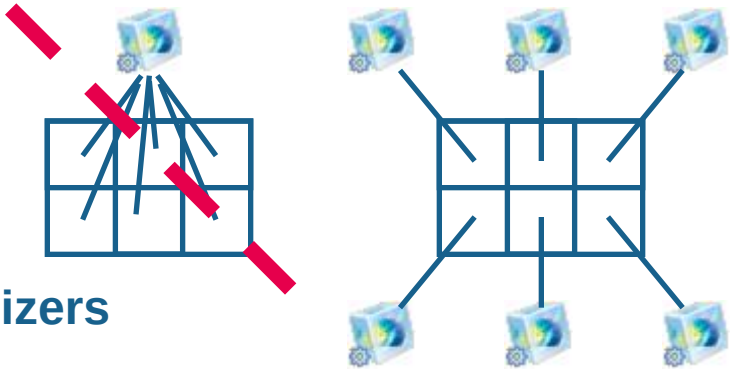
- Comparative optimization starting at 10 different points
- LHSOpt is
  - more robust ➔ always convergence
  - reliably finds the global optimum ➔ 100% result for each starting point
  - provides a speed-up by up to 300% ➔ 75% of potential from parallelization benchmarks

Starting Point	Fminsearch		LHSOpt		Speed-Up
	Duration (h)	Result	Duration (h)	Result	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

Summary & Outlook

- **Shared-Memory Parallelization cannot uses multi-core hardware efficiently**
- **Simulation of multiple models with single cores accesses hardware potential efficiently**

➔ Huge speed-up potential for suitable optimizers



- **A simple population based optimizer LHSOpt is able to access already up to 300% of 400% speed-up potential, including typical robustness and reliability**
- **Optimization workflow of LHSOpt works with single user license**
- **Even several single user licenses (workstations) can operate on the same task**

- **In the future, more sophisticated parallelized optimizers will access a large fraction of speed-up potential reliably**