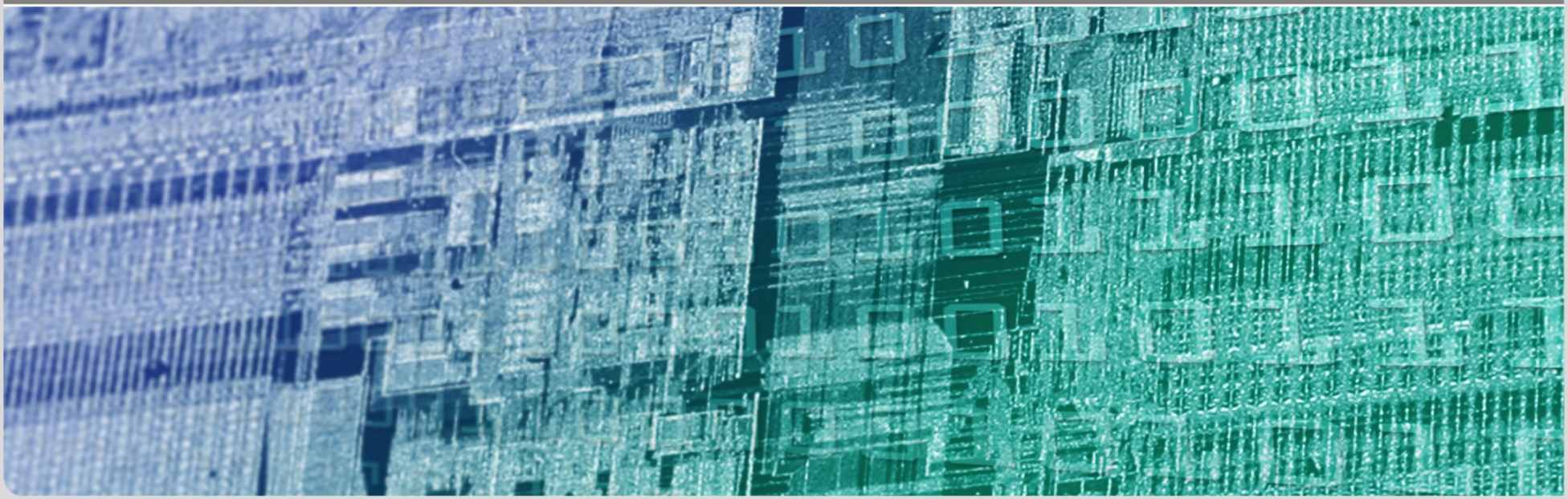
Karlsruhe Institute of Technology

Online Tuning of Stream Programs or How To Get The Most Ouf Of Your Multicore

Walter F. Tichy

Institute for Program Structures and Data Organization

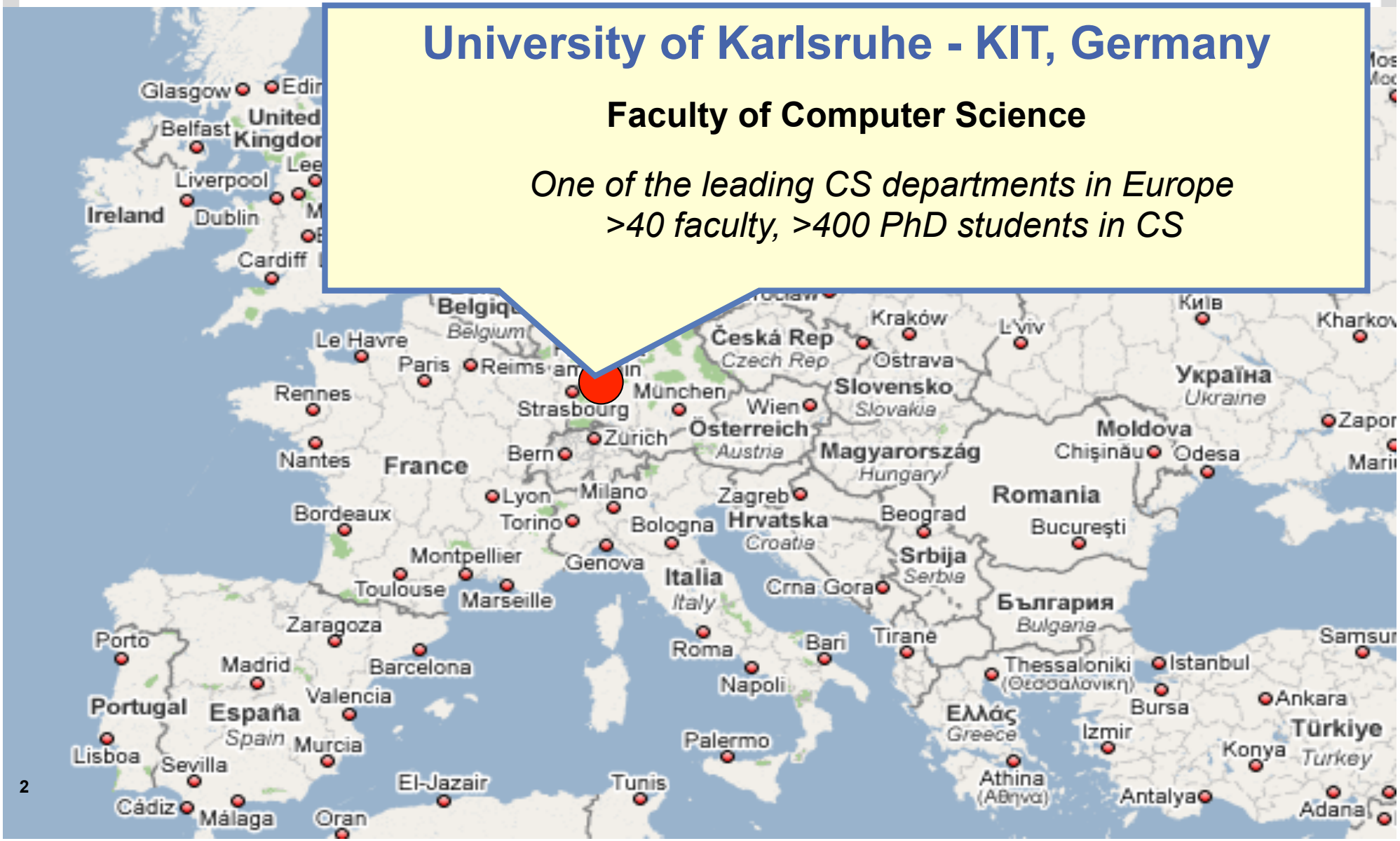


KIT – University of Baden-Württemberg and National Research Center of the Helmholtz Association

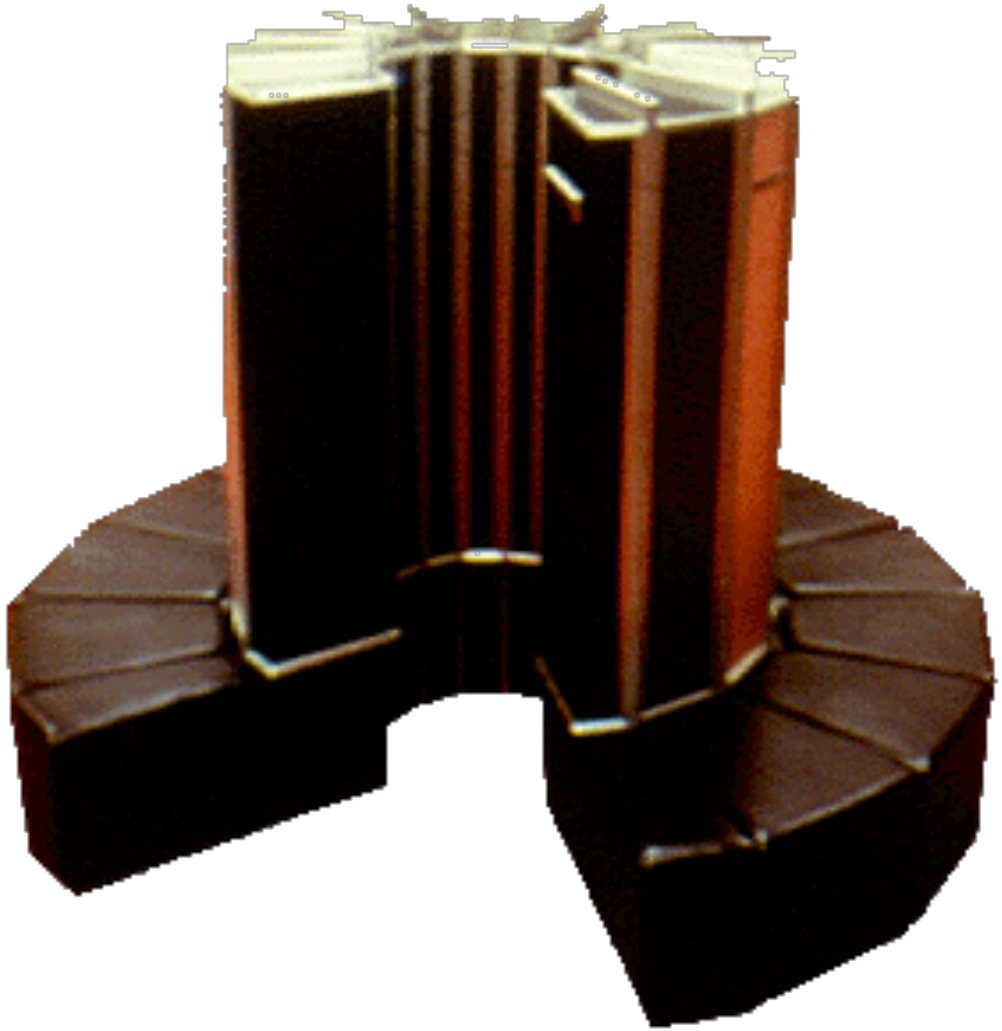
Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Where is Karlsruhe?



The changing parallel computing landscape

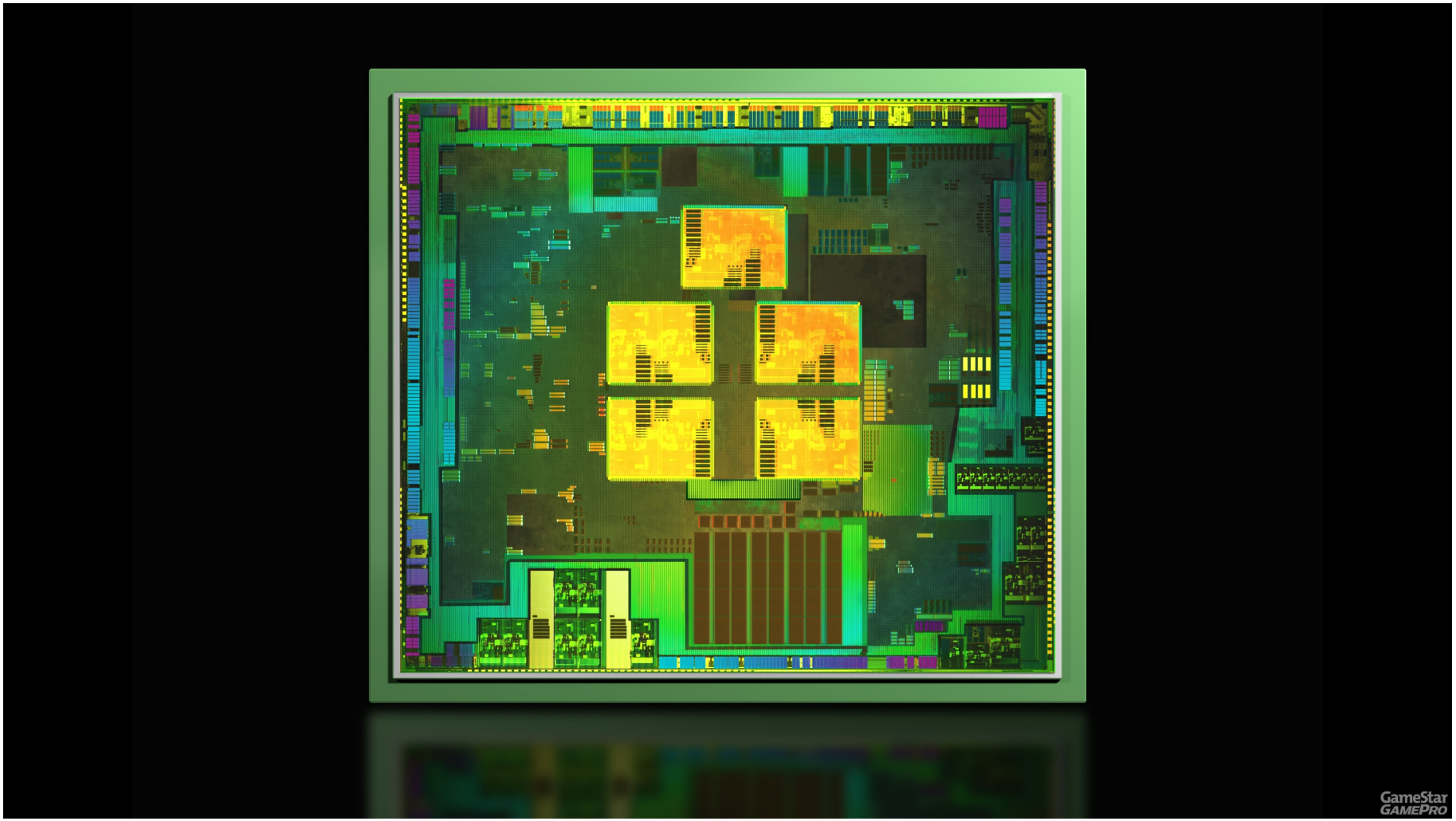


Cray vector computer 1976

The first five-core mobile phone

HTC One X, Feb. 1912,
Powered by Nvidia Tegra 3

Nvidia Tegra 3



Nvidia Tegra 3 Schematic

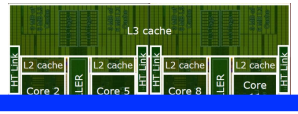
1 core at 500 MHz (battery saver)

4 cores at 1.5 GHz

1 GPU

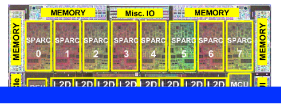
AMD Opteron 12 cores

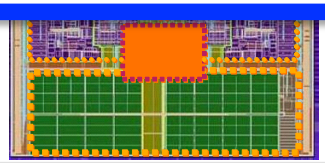
~1.8 Bill. T. on 2x3.46cm^2



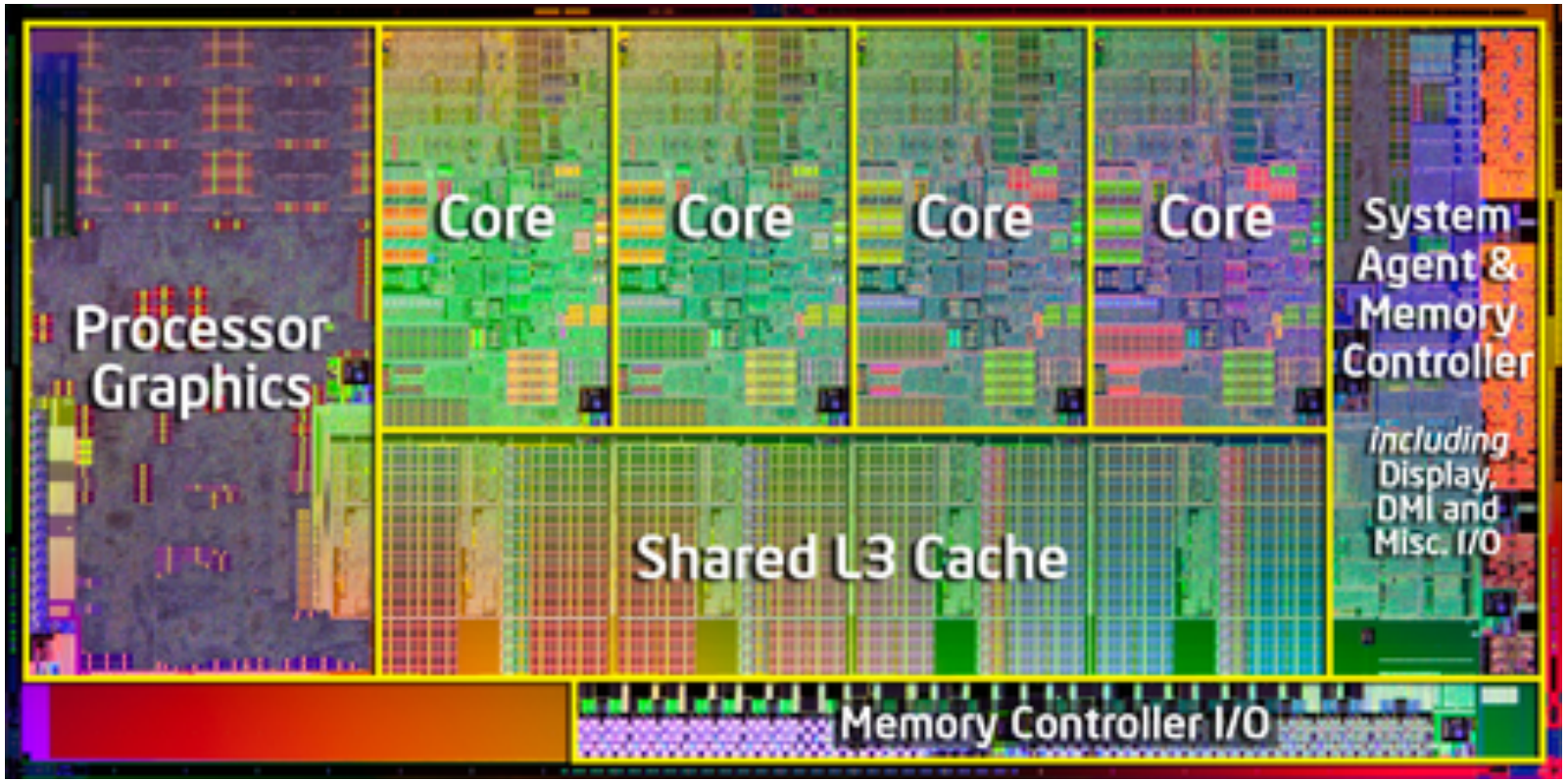
Sun Niagara3 16 cores

~1 Bill. T. on 3.7cm^2





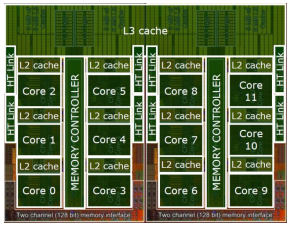
The 2011 Intel Sandy Bridge



Currently: 4 CPUs, 6 graphics Execution Units
Later: 8 CPUs, 12 graphics Execution Units

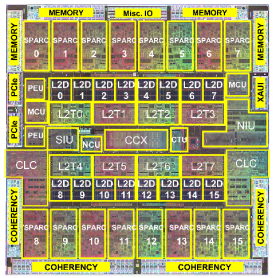
AMD Opteron 12 cores

~1.8 Bill. T. on 2x3.46cm²



Sun Niagara3 16 cores

~1 Bill. T. on 3.7cm²



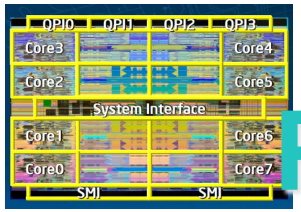
Karlsruhe Institute of Technology

Intel SCC 48 cores

~1.3 Bill. T. on 5.6 cm²

Intel 8 cores

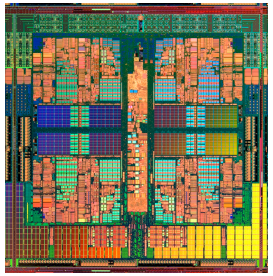
~2.3 Bill. T. on 6.8cm²



Parallelism Everywhere

Intel 4 cores

~582 Mio. T on 2.86cm²

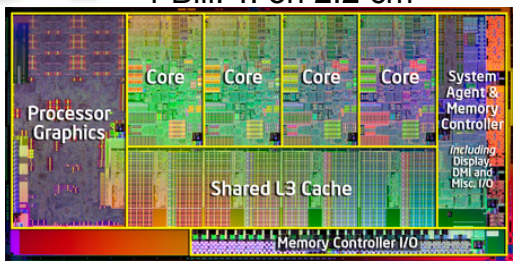


What is Missing?

Software!

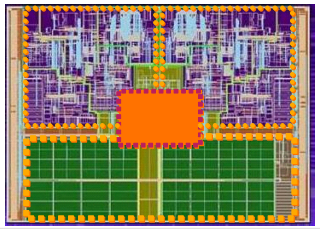
Intel Sandy Bridge 4+6 cores

~1 Bill. T. on 2.2 cm²



Intel 2 cores

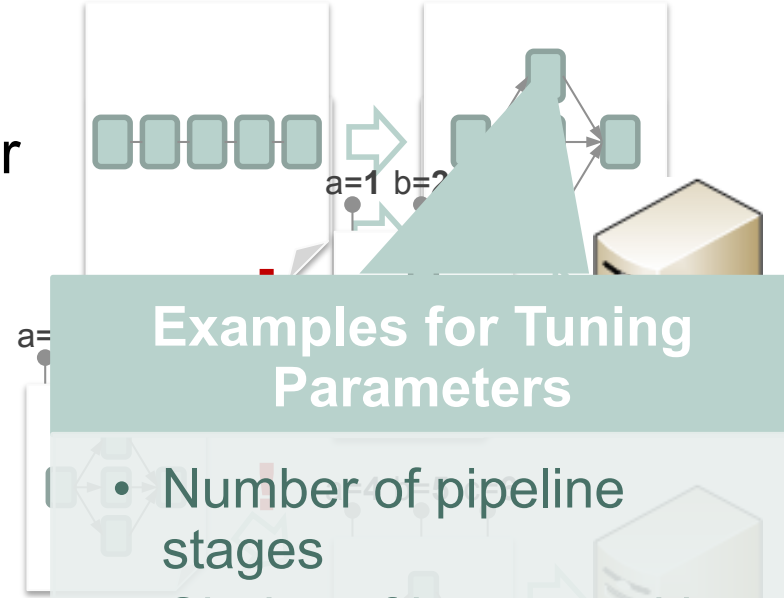
~167 Mio. T. on 1.1cm²



Fixing Parallel Performance Problems

- Parallelization is complex and error-prone
- Parallel programs contain a number of tuning parameters
- Manual optimization difficult and time-consuming
- Each target platform may require re-tuning

- **Auto-Tuning**: Let the computer do the tuning!

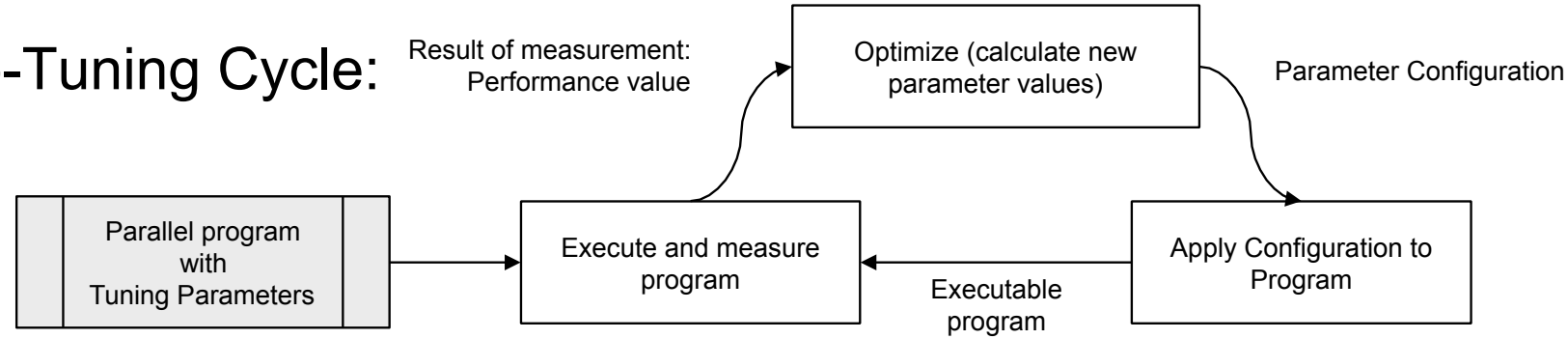


Examples for Tuning Parameters

- Number of pipeline stages
- Choice of best algorithm implementation
- Order of execution
- Size of data partitions
- Number of workers
- Type of core
- Load balancing strategy

Online Auto-Tuning

- Auto-Tuning Cycle:



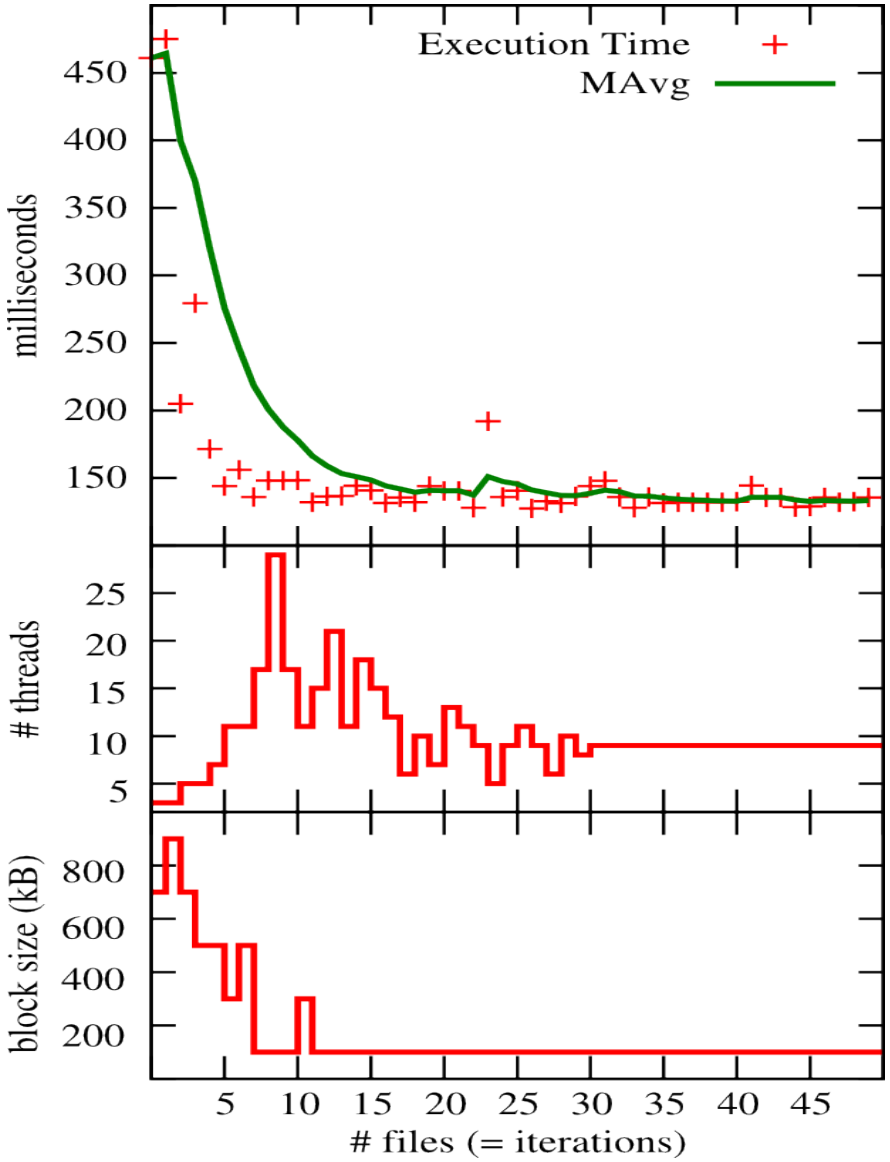
- Example (pseudo code)

```
TuningParameter numthreads(3, 64);
TuningParameter blocksize(100, 900, 100);

for(int i=0; i<numfiles; ++i) {
    startMeasurement();

    compress(files[i], blocksize, numthreads);

    stopMeasurement();
}
```

Tuning Parameter

Measurement Section

Auto-Tuning: BZip2 example



Parallelized BZip2, compressing 50 files on a machine with 8 cores

Initial tuning parameter values: 3 threads, block size 700 kB

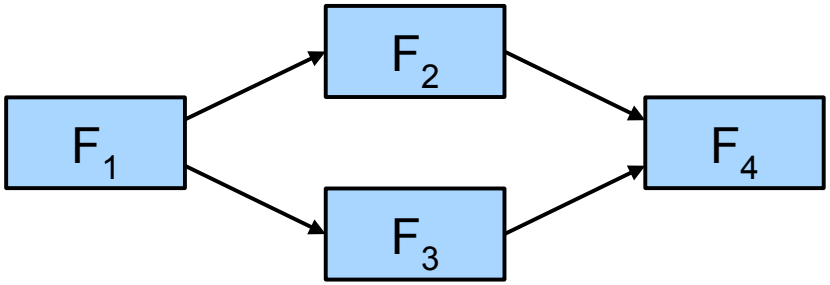
Runtime without tuning: 22,9 s

Runtime with Auto-Tuner: 8 s

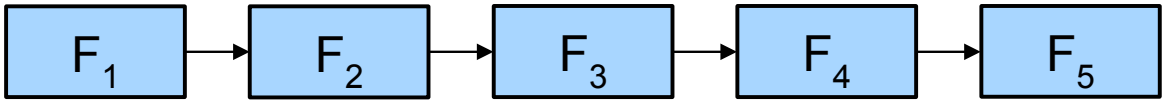
Best possible time (start with best configuration): 6,5 s

Stream Programming Paradigm

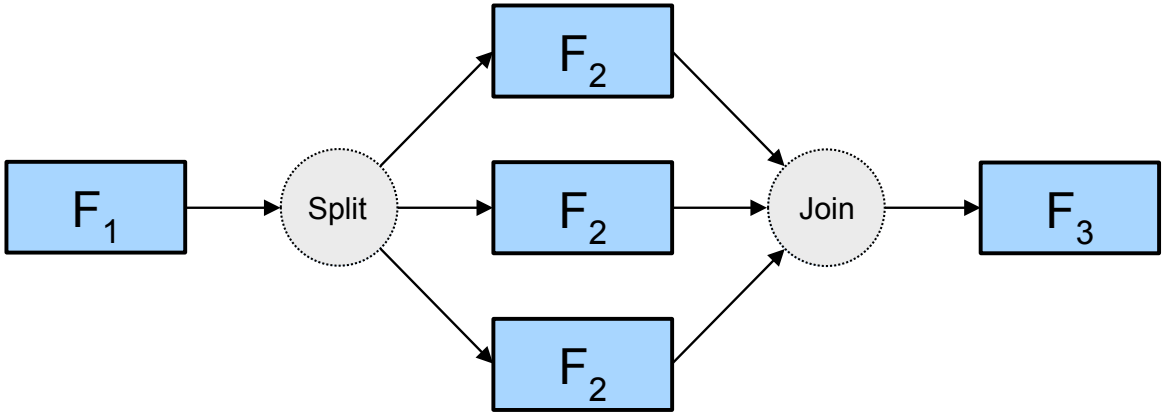
- A stream of elements flows through a graph of processing modules called *filters*.
- Task parallelism



- Pipeline parallelism

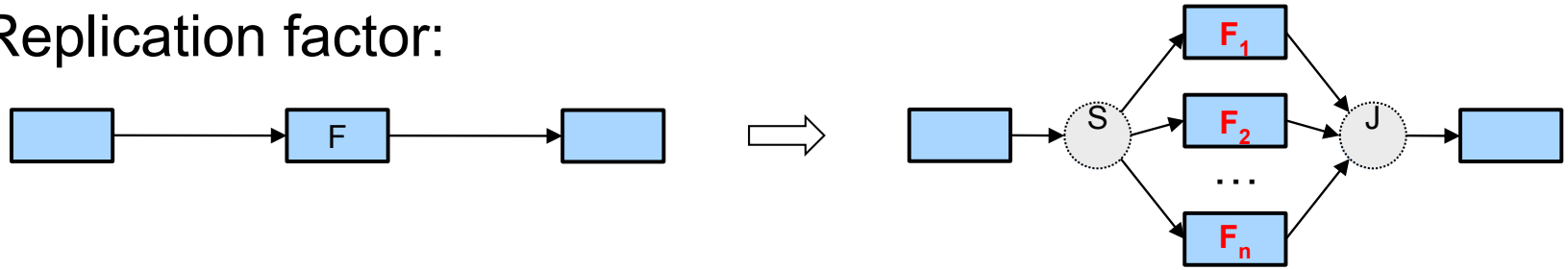


- Data parallelism
  (by filter replication)

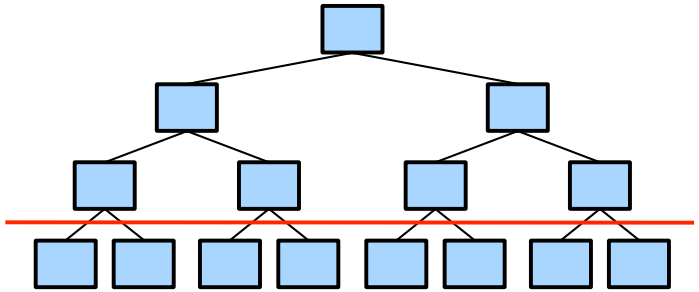


(Some) Implicit Tuning Parameters

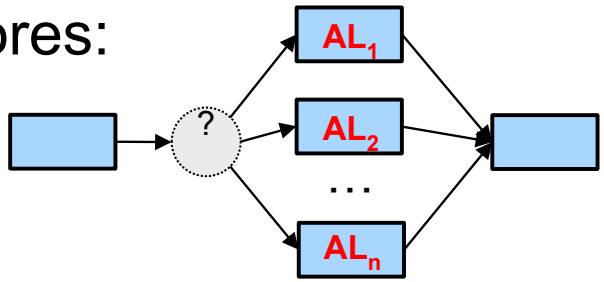
- Replication factor:



- Cut-off depth:

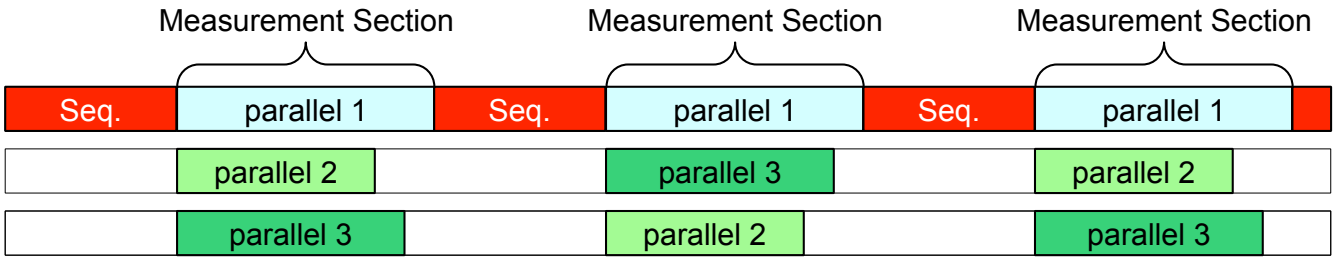


- Alternative Algorithms/Cores:

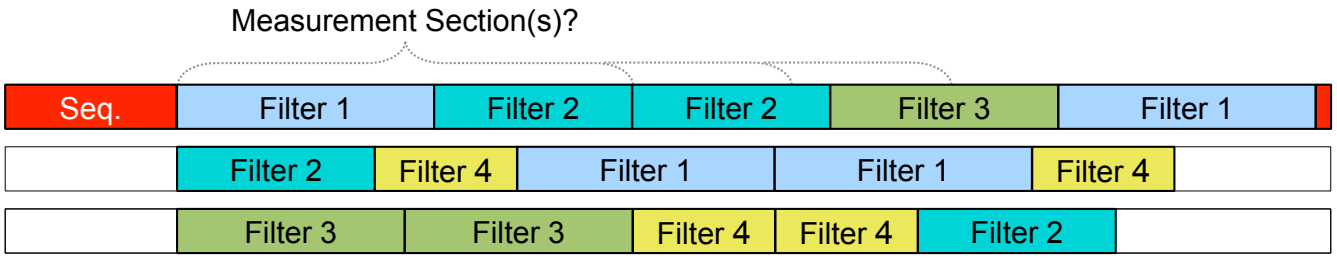


Measurement Sections in Stream Programs

- „Classic“ Fork/Join pattern:



- Stream program:

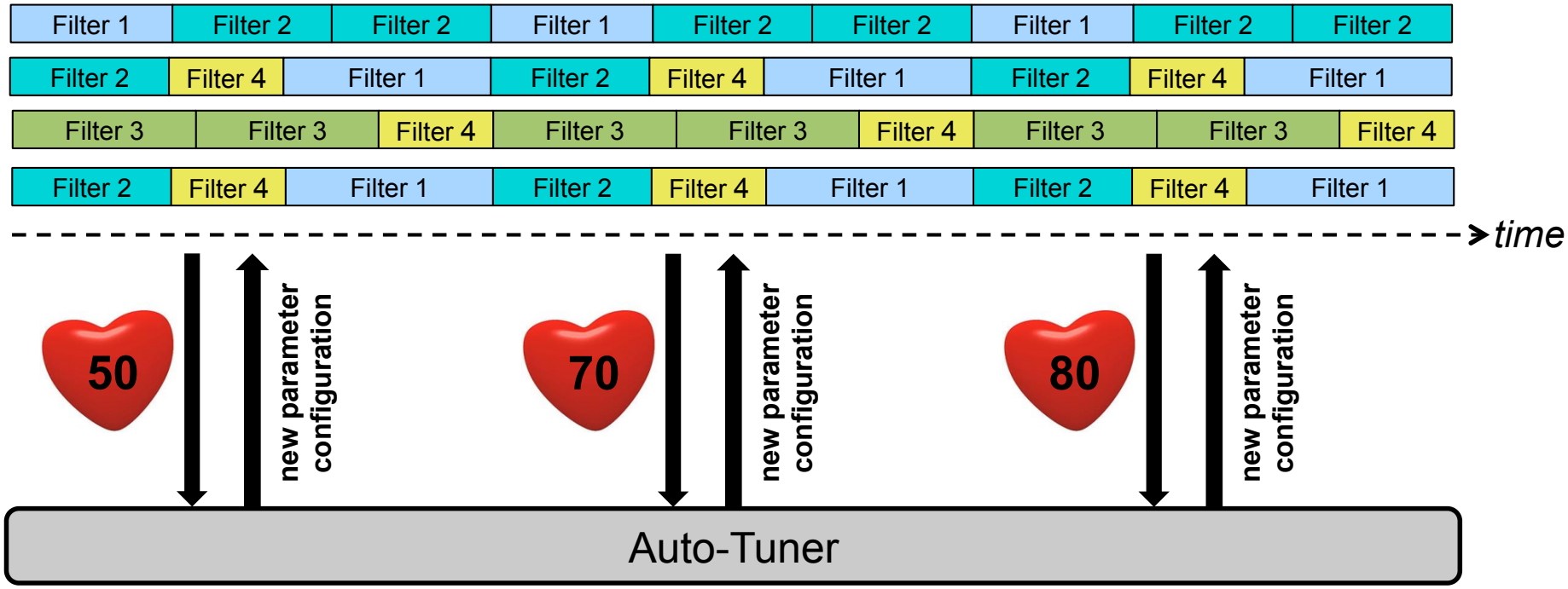


- Solution:
  - Count „heart beats“ (events triggered by stream elements)
  - Use heart beats to evaluate performance

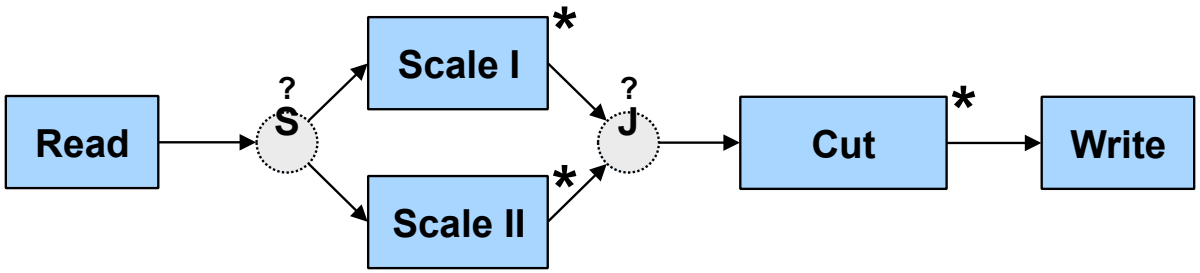
Using Heartbeats for Online Tuning

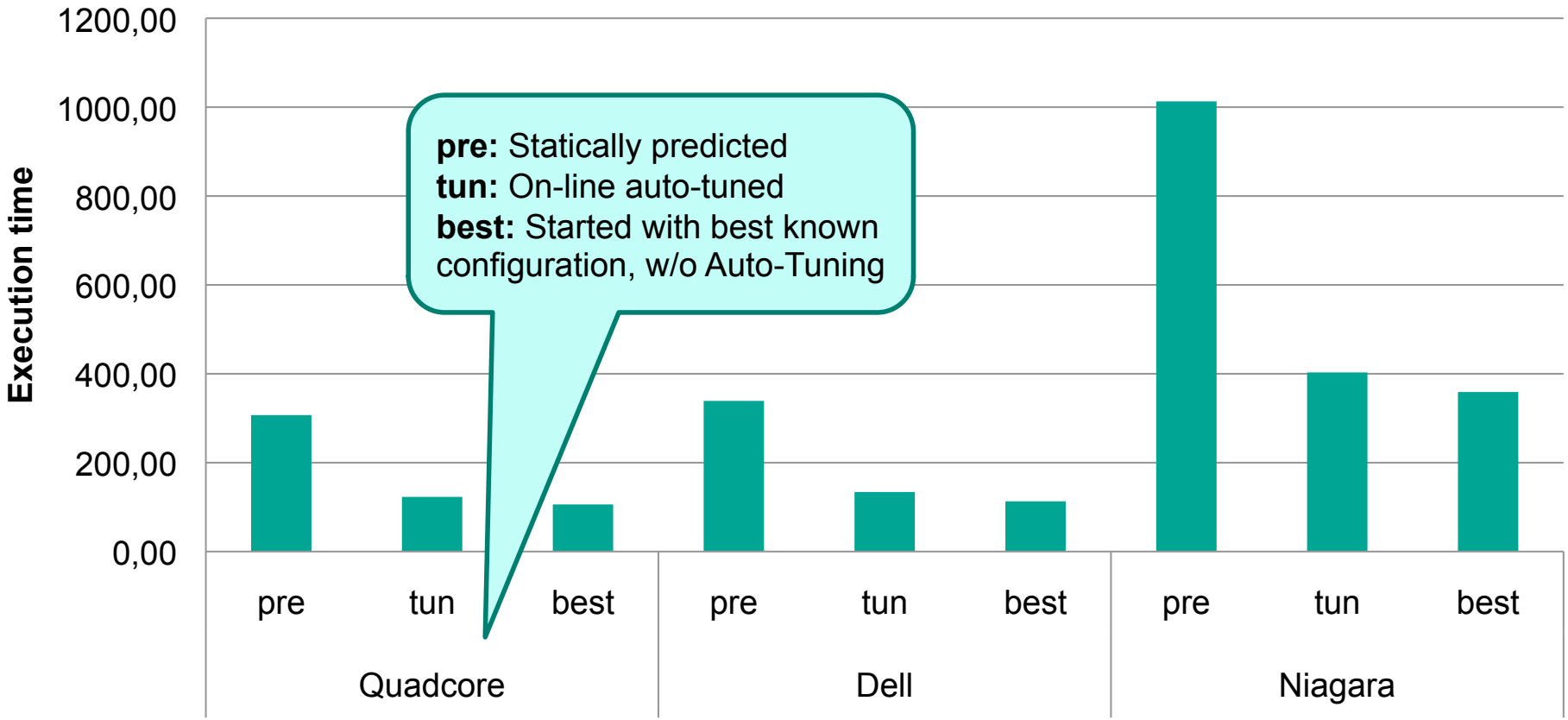
- Heartbeats are emitted by sink filters
- The faster the heartbeat, the better the performance
- Heartbeats serve as an input signal for online auto-tuners

Illustrating Example:



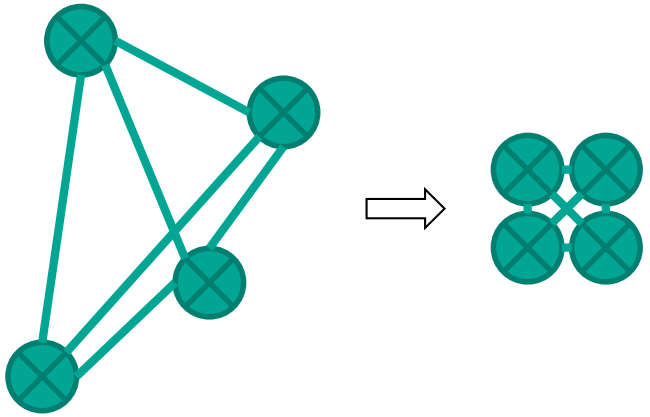
Benchmark 1: Video zoom

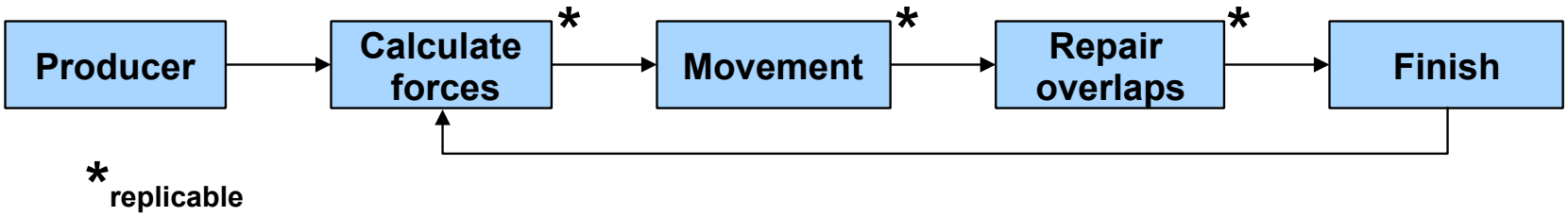




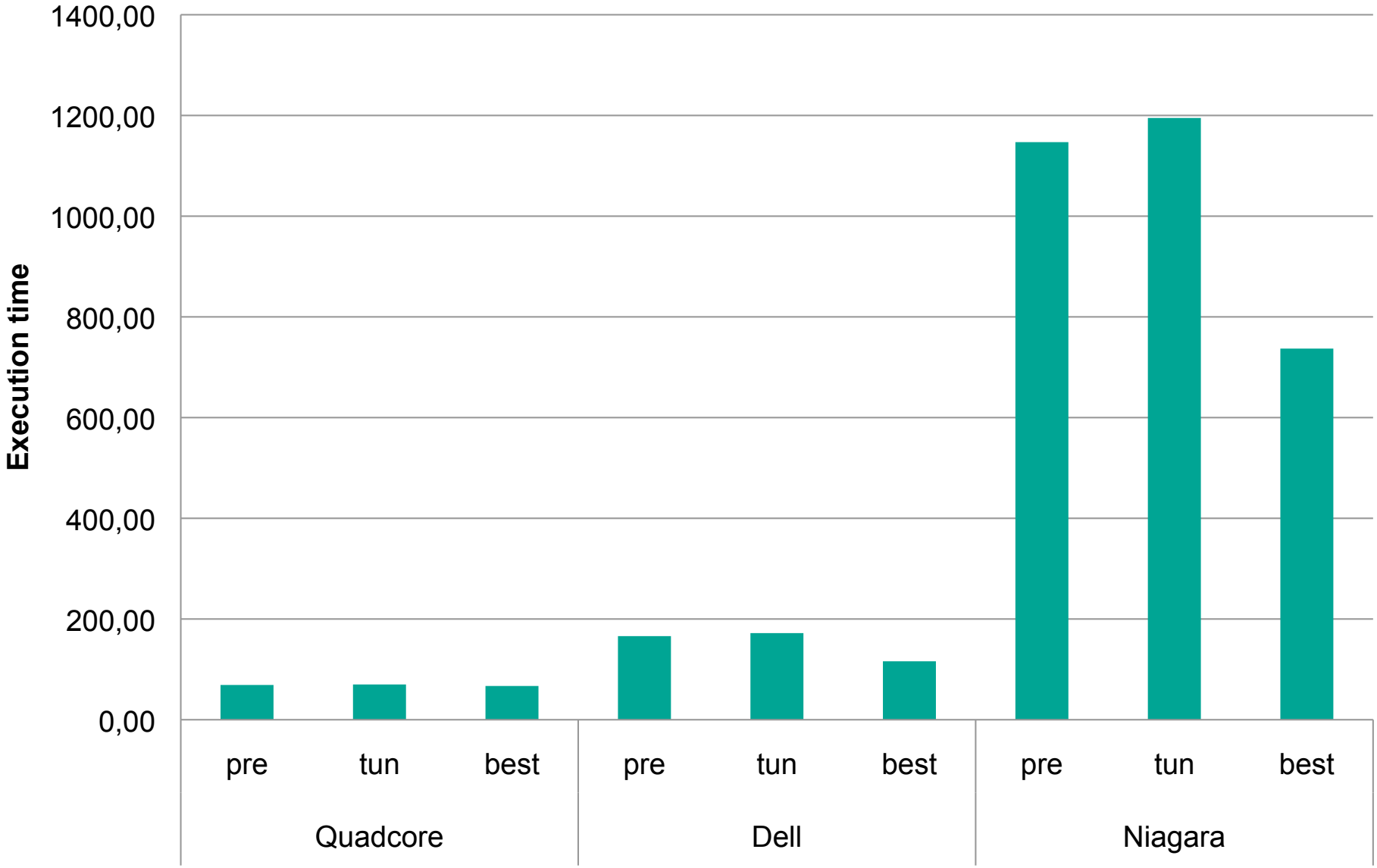
Benchmark 2: Electric (Placement of circuits on a die)

- Part of VLSI design application
- 5 Filters with feedback loop and teleports
- 4 Tuning parameters

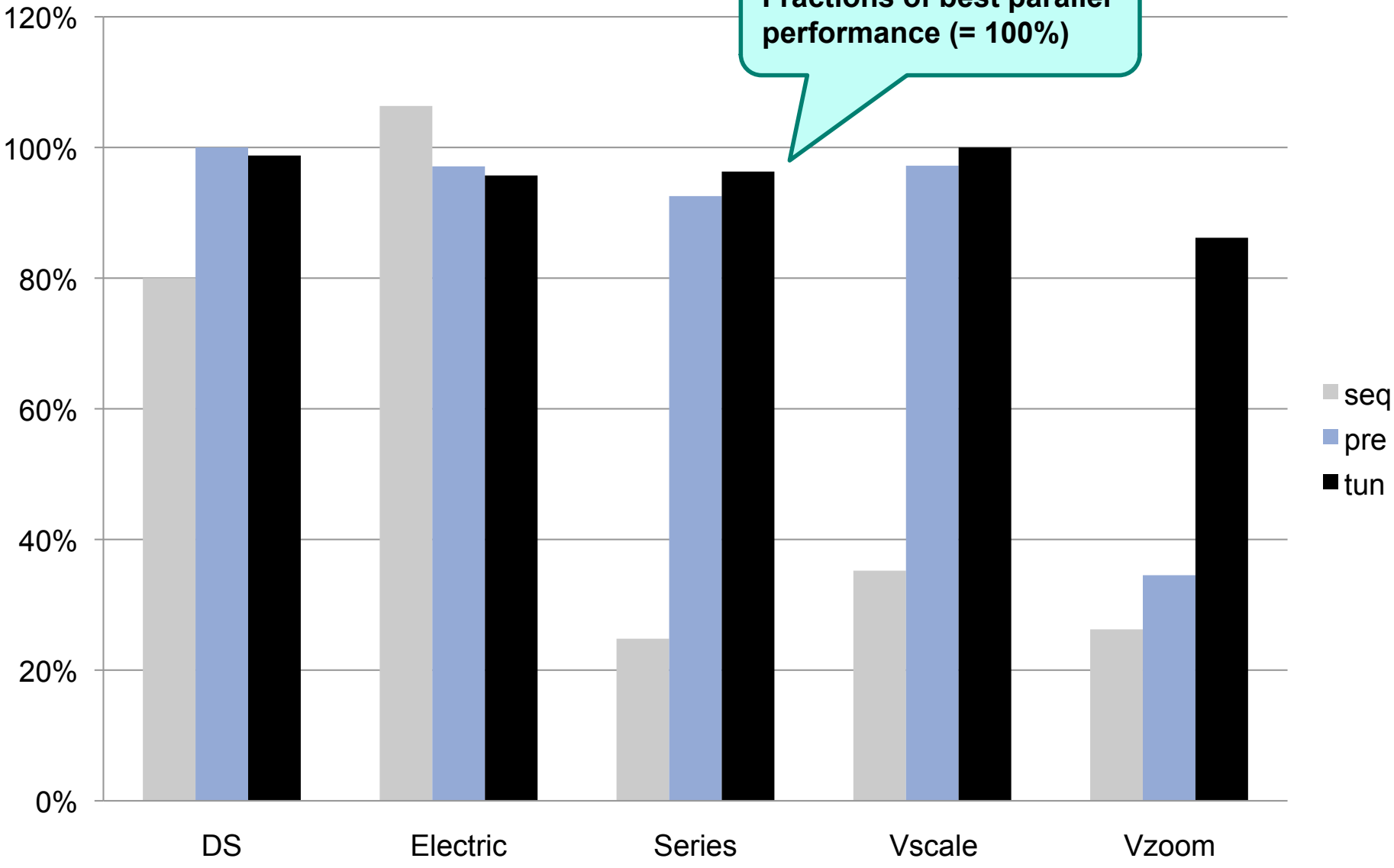




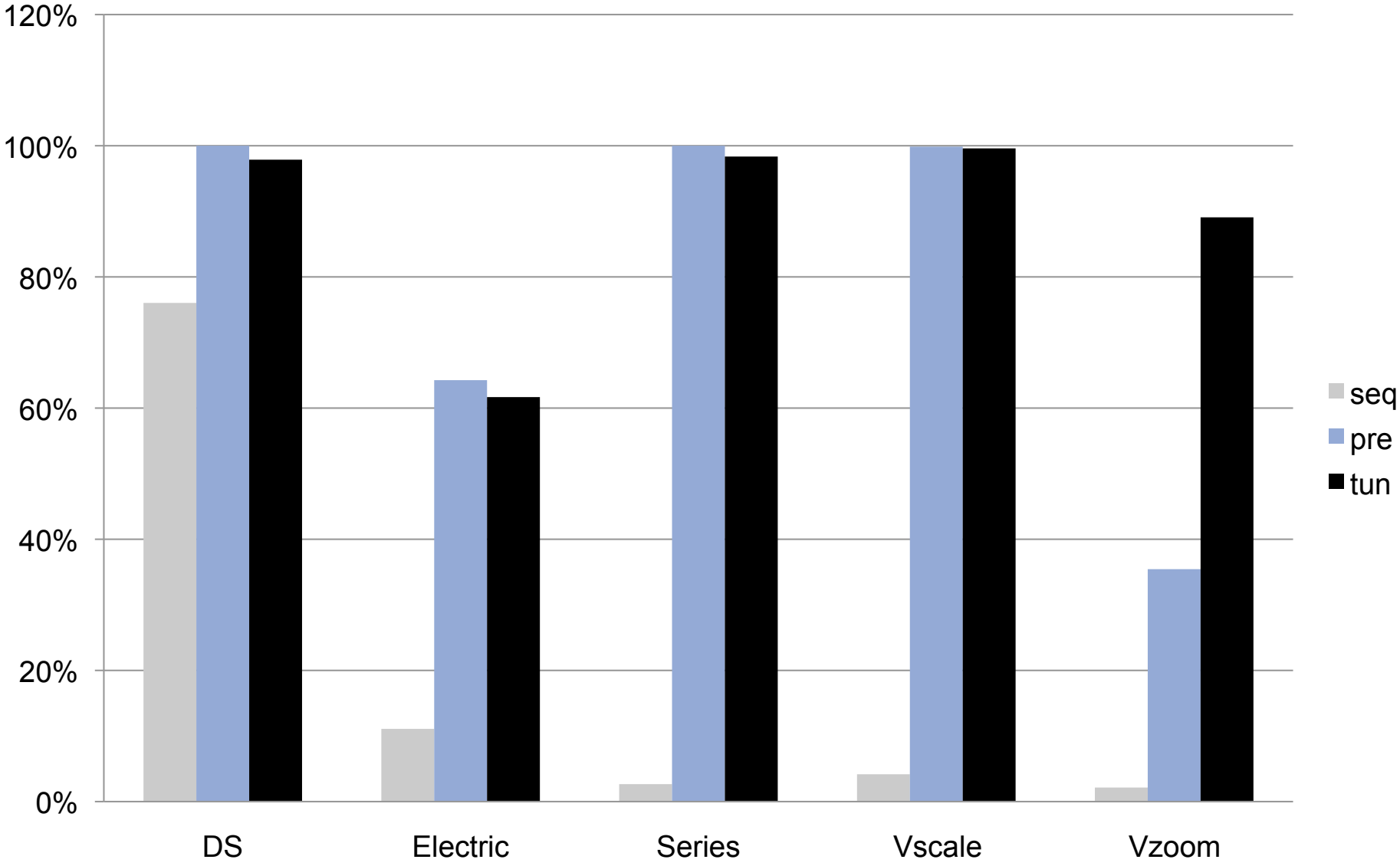
Electric: Results



Benchmarks on 4 cores



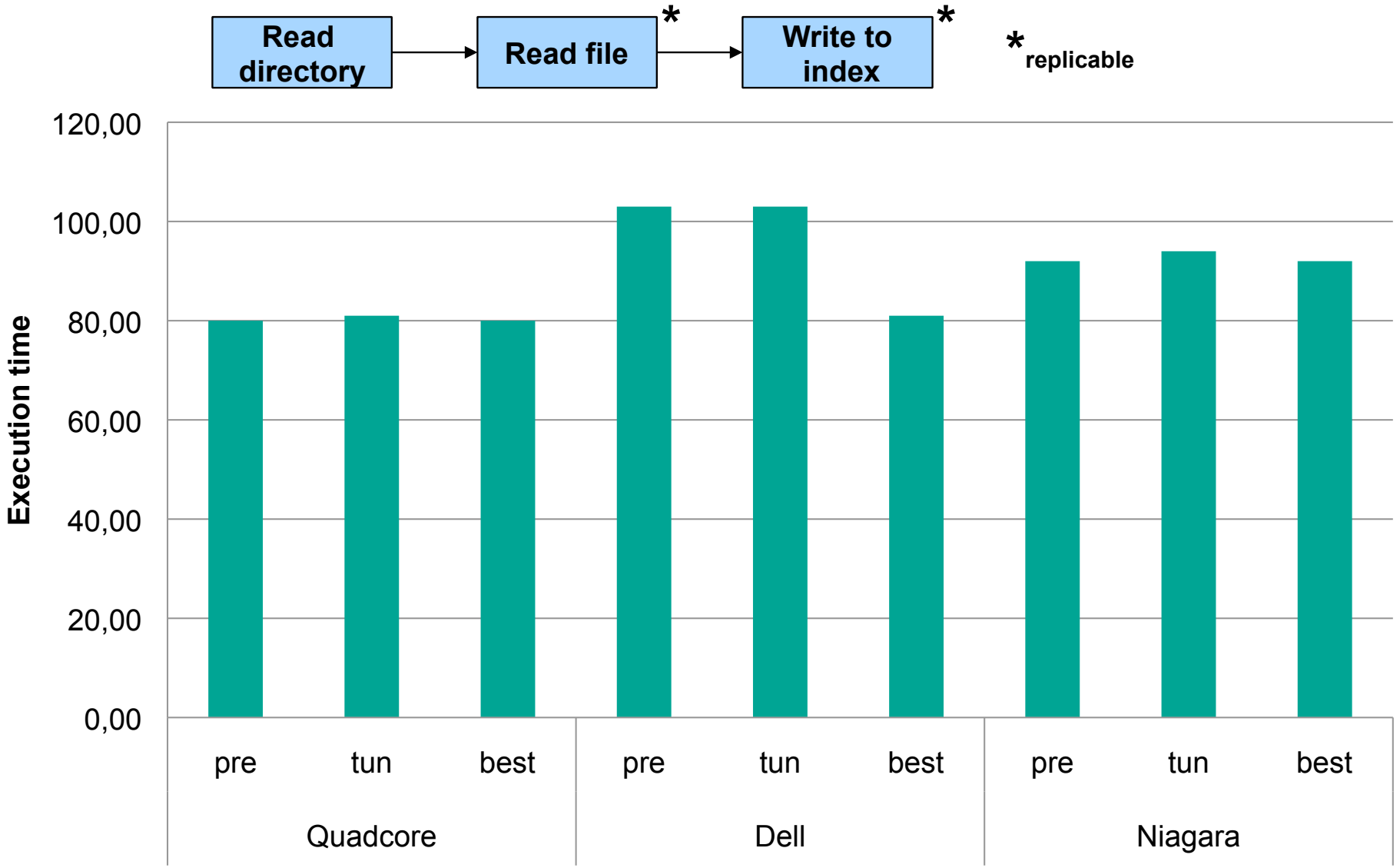
Benchmarks on 64 cores (Niagara)



Related Work (Selection)

- *ATLAS/AEOS* (Whaley et al., 2000)
  - Auto-tuning system for algebraic operations and algorithms
  - Domain specific approach
  - No support for parallel programs
- *Active Harmony* (Tapus et al., 2002)
  - Search-based auto-tuning system for library optimization
  - Comprehensive analysis of search algorithms
  - Not applicable for parallel programs
- *MATE* (Morajko et al., 2007)
  - Model-based tuning system for distributed PVM programs
  - Provides good performance predictions
  - Limited to special program structures
- *ATUNE* (Schaefer, Tichy, 2010)
  - General-purpose auto-tuner
  - Offline tuner (trial runs)
  - Pattern language for expressing parallel patterns (TADL)

Benchmark 3: Desktop search



Summary

- Computers are not the bottleneck.
- Programmers are!
- Stream programming simplifies parallel programming
  - Typical parallel patterns easy to write
  - Auto-tuning finds optimal operating conditions
  - Saves lots of tuning work

- Further research
  - Improved online search algorithms
  - Use static model to predict good starting values
  - Use auto-tuning to distribute work over heterogeneous cores

THANK YOU!
QUESTIONS?

With many thanks to
Frank Otto, Thomas Karcher, Jonas Thedering, Victor Pankratius

For more information, see: http://www.ipd.kit.edu/Tichy/

Karlsruhe Institute of Technology

BACKUP SLIDES

Benchmarks on 8 cores