TECHNISCHE UNIVERSITÄT DARMSTADT

System and Parallel Programming

Prof. Dr. Felix Wolf

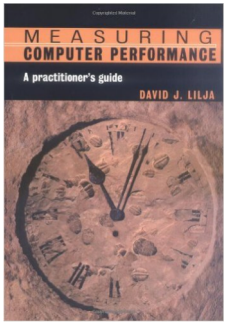
PERFORMANCE ANALYSIS

Outline

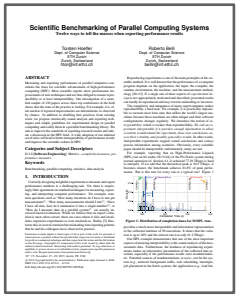
- Introduction
- Performance measurement
- Performance analysis
- Performance modeling

Literature

- Measuring Computer Performance - A Practitioner's Guide
  David J. Lilja, Cambridge University Press, 2000



- Torsten Hoefler, Roberto Belli: Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results. Proc. of the SC15 Conference, 2015.



Charles Babbage, 1791 - 1871
(inventor of the first mechanical computer)

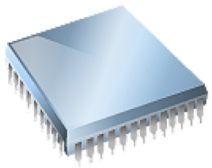
TECHNISCHE UNIVERSITÄT DARMSTADT

"The most constant difficulty in contriving the engine has arisen from the desire to reduce the time in which the calculations were executed to the shortest which is possible."

Performance

TECHNISCHE UNIVERSITÄT DARMSTADT

$$\textbf{Performance} \sim \frac{1}{\text{Resources to solution}}$$



Hardware

Time

Energy

… and ultimately

Money

Performance optimization pays off

TECHNISCHE UNIVERSITÄT DARMSTADT

Example: HPC Service RWTH Aachen (2012)

~300 TFlops Bull/Intel cluster

Total cost of ownership (TCO) per year: **5.5** M€

Resource type	Fraction of TCO
Hardware	1/2
Energy	1/4
Staff	1/8
Others	1/8

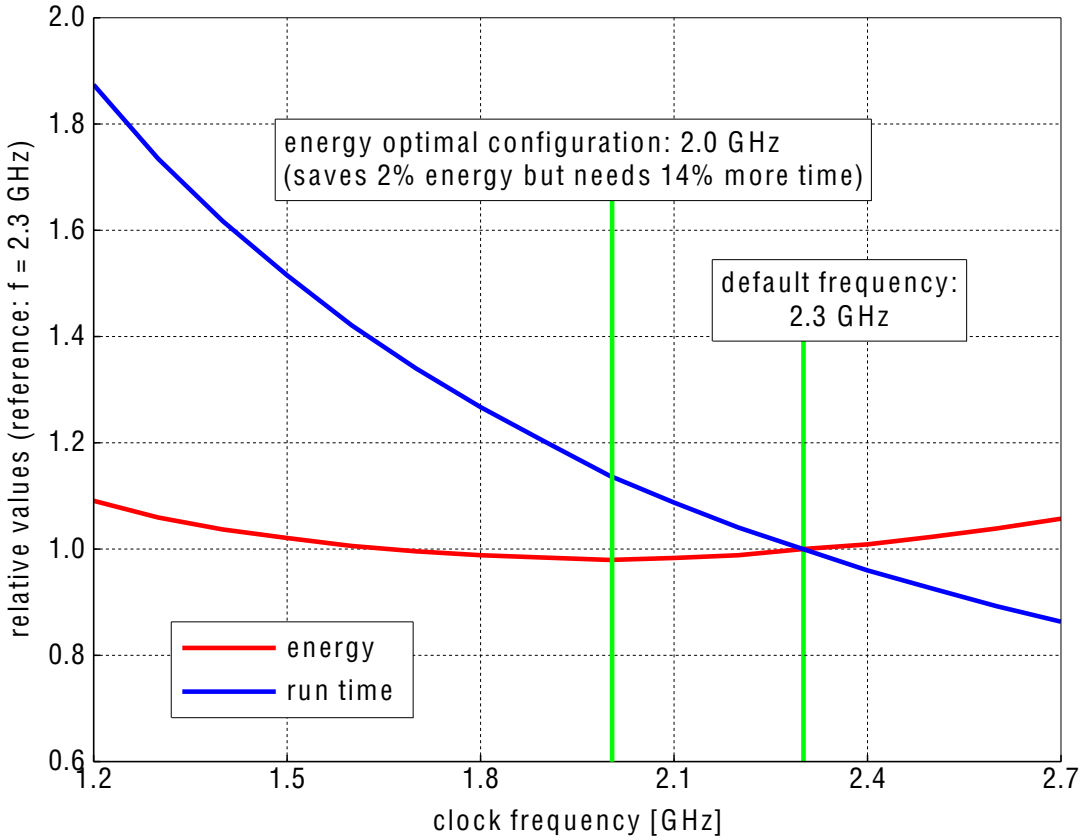
Source: Bischof, an Mey, Iwainsky: Brainware for green HPC, Computer Science-Research and Development, Springer, 2012.

Tuning the workload by 1% will “save” 55k€ per year ~ 1 FTE

Trading runtime for energy

TECHNISCHE UNIVERSITÄT DARMSTADT

Runtime vs. energy for CFD code INDEED at different clock speeds



Are 2% less energy worth 14% more time?

Courtesy of GNS mbH

Energy delay product (EDP)

TECHNISCHE UNIVERSITÄT DARMSTADT

Weighing runtime vs. energy largely a policy decision

$EDP = E * T^w$ - where E = energy, T = runtime, w = 1,2, or 3

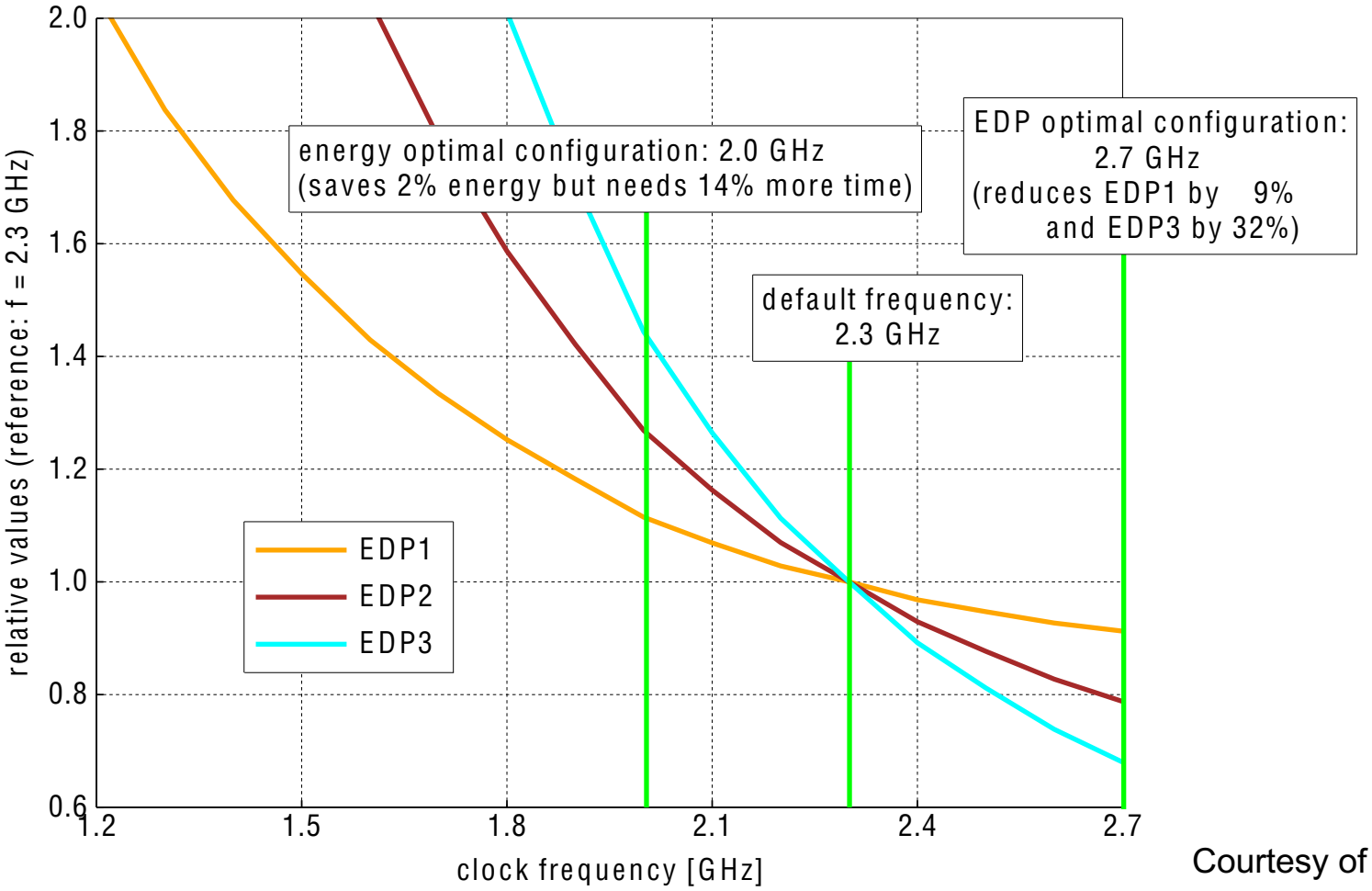
The value of w determines how important performance is

- w=2 generally suggested for HPC
- Sometimes 2 not enough – especially for short runtimes

Energy-Efficient High Performance Computing: Measurement and Tuning. James H. Laros III, Kevin Pedretti, Suzanne M. Kelly, Wei Shu, Kurt Ferreira, John Van Dyke, Courtenay Vaughan, Spinger, 2013.

Different EDPs for INDEED

TECHNISCHE UNIVERSITÄT DARMSTADT



Courtesy of GNS mbH

Goals of performance analysis

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Compare alternatives** - which configurations are best under which conditions?
- **Determine the impact of a feature** – before-and-after comparison
- **System tuning** – find parameters that produce best overall performance
- **Identify relative performance** – which program / algorithm is faster?
- **Performance debugging** – search for bottlenecks
- **Set expectations** – provide information for users

Serial vs. parallel performance

TECHNISCHE UNIVERSITÄT DARMSTADT

Serial programs

- Memory behavior
- Instruction-level parallelism

Parallel programs

- Amount of parallelism
- Granularity of parallel tasks
- Inter-task communication
- Inter-task synchronization
- Contention (e.g., for memory bandwidth)

Three basic performance analysis methods

- Analytical modeling
- Simulation
- Measurement

Analytical modeling

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mathematical description of the system
  - Allows quick change of parameters
  - Rapid solution
- Often requires restrictive assumptions, limiting accuracy
- Allows insights into fundamental properties such as scalability
- Can be used to set expectations for simulations or measurements
- Example
  - Memory delay $t_{avg} = ht_c + (1-h)t_m$
  - Parameters obtained from manufacturer or measurement

Simulation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Program written to model important features of the target system
- Can be easily modified to study the impact of changes
- Cost
  - Writing the program
  - Running the program
- Impossible to model every small detail
  - Simulation refers to “ideal” system; sometimes low accuracy
- Example
  - Cache simulator
  - Parameters: size, block size, associativity, cache and memory delays

Measurement

TECHNISCHE UNIVERSITÄT DARMSTADT

- No simplifying assumptions
- Highest credibility
- Information only on specific system being measured
- Harder to change system parameters in a real system
- Difficult and time consuming
- Software tools can help
- Should be used in conjunction with modeling
  - Can aid the development of performance models
  - Performance models set expectations against which measurements can be compared

Comparison

TECHNISCHE UNIVERSITÄT DARMSTADT

Measurement – Simulation – Model

Adapted from: T. Hoefler, W. Gropp, W. Kramer, and M. Snir. Performance modeling for systematic performance tuning. In State of the Practice Reports, SC '11, pages 6:1–6:12. ACM, 2011.

Comparison (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

	Analytical modeling	Simulation	Measurement
Flexibility	High	Medium	Low
Accuracy	Low	Medium	High
Cost / difficulty	Using a model is easy and cheap, developing one can be hard	The cost of running a simulator depends on the simulation, developing one is usually hard	Obtaining measurements requires skills & experience and can be expensive in terms of compute time

Metrics of performance

TECHNISCHE UNIVERSITÄT DARMSTADT

What can be measured?

- A count of how many times an event occurs
  - Example: number of input / output requests
- The duration of some time interval
  - Example: duration of these requests
- The size of some parameter
  - Example: number of bytes transmitted or stored

Derived metrics

- E.g., rates / throughput
- Needed for normalization

Primary performance metrics

TECHNISCHE UNIVERSITÄT DARMSTADT

- Execution time, response time
  - Time between start and completion of a program or event
  - Most consistent and reliable measure of performance
  - Wall-clock time vs. CPU time
- Throughput
  - Total amount of work done in a given time
- Problem: execution time often subject to run-to-run variation and therefore hard to reproduce exactly
  - Summarize multiple measurements

Alternative performance metrics

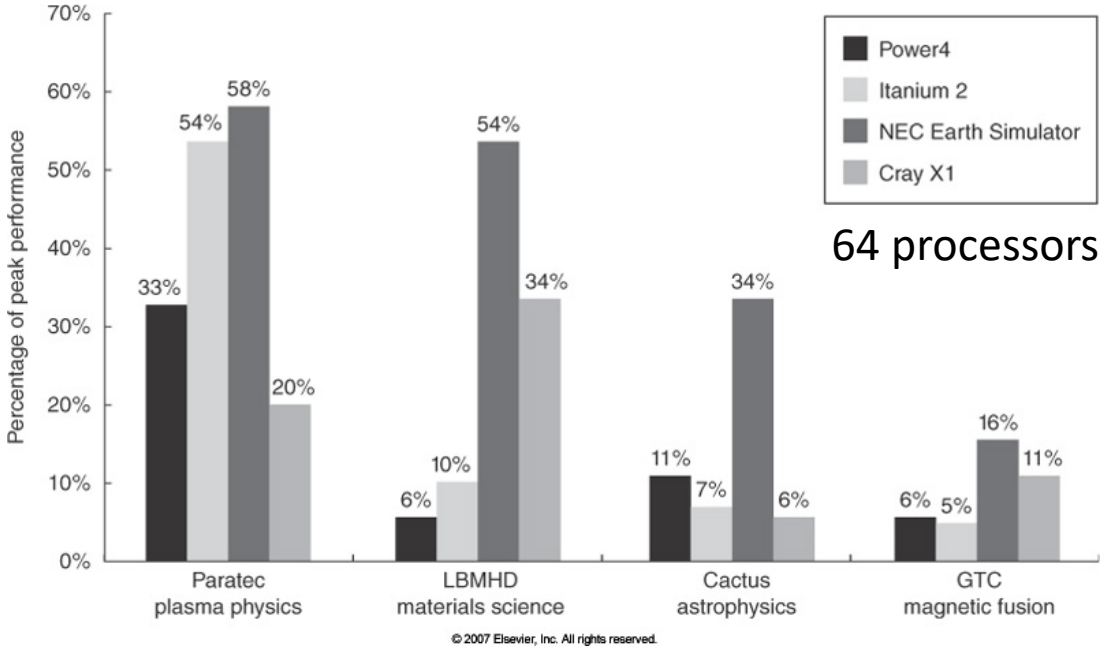
TECHNISCHE UNIVERSITÄT DARMSTADT

- Clock rate
- Instructions executed per second
- FLOPS – floating-point operations per second
- Benchmarks
  - Standard test program(s) + standardized methodology
  - E.g., SPEC, Linpack
- QUIPS / HINT [Gustafson and Snell, 95]
  - Quality improvements per second
  - Quality of solution instead of effort to reach it

Peak performance

TECHNISCHE UNIVERSITÄT DARMSTADT

- Peak performance is the performance a computer is guaranteed not to exceed



Source: Hennessy, Patterson: Computer Architecture, 4th edition, Morgan Kaufmann

How to measure time?
UNIX gettimeofday()

TECHNISCHE UNIVERSITÄT DARMSTADT

```
#include <sys/time.h>
#include <stdio.h>

double get_seconds()
{
  struct timeval tv;
  gettimeofday(&tv, NULL);
  return tv.tv_sec + tv.tv_usec * 1.0e-6;
}

int main()
{
  double start = get_seconds();
  /* do some work */
  double end = get_seconds();
  printf("Time: %f seconds\n", end-start);
}
```

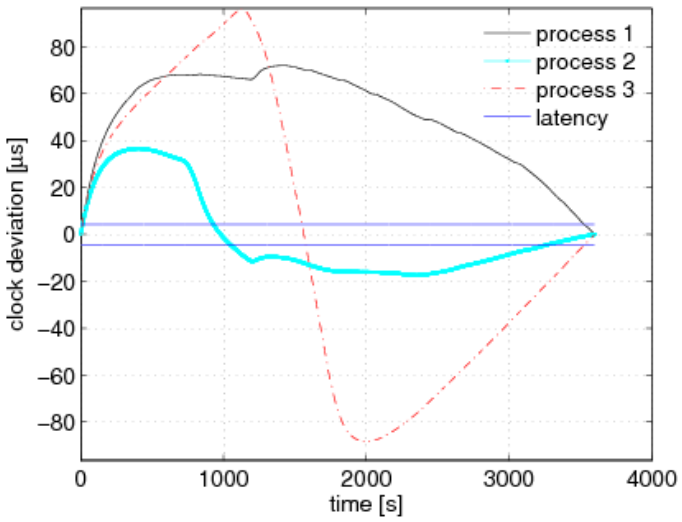
Further timers

Function	Description	Standard
clock()	Approximation of processor time	ANSI-C
times()	Wall-clock time; system and user time	UNIX / IEEE
getrusage()	User time, system time, memory usage, page faults, etc.	UNIX / BSD
MPI_Wtime()	Wall-clock time	MPI
`omp_get_wtime()`	Wall-clock time	OpenMP

Insufficiently synchronized clocks on clusters

TECHNISCHE UNIVERSITÄT DARMSTADT

- Clocks on a cluster may differ in terms
  - Offset at a given time
  - Drift (i.e., speed)
- Drift often unstable
  - Due to temperature variation or NTP synchronization
  - Linear interpolation may not be enough to preserve logical event order



Clock deviation after linear interpolation

Summarizing (deterministic) results

TECHNISCHE UNIVERSITÄT DARMSTADT

What?	Example	How?
Costs	Flop count; execution time (in certain scenarios)	• Arithmetic mean $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
Rates	FLOP/s	• Ideally arithmetic mean of numerator divided by arithmetic mean of denominator • Harmonic mean otherwise $\bar{x} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$

Torsten Hoefler, Roberto Belli: Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results. Proc. of the SC15 Conference, 2015.

Summarizing (deterministic) results (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

What?	Example	How?
Unitless ratios	Speedup	• Don't. Summarize the costs or rates that the ratio is based on instead • Only if these are not available use the geometric mean $\bar{x} = \sqrt[n]{\prod_{i=1}^{n} x_i}$

harmonic mean ≤ geometric mean ≤ arithmetic mean

Torsten Hoefler, Roberto Belli: Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results. Proc. of the SC15 Conference, 2015.

Non-deterministic results

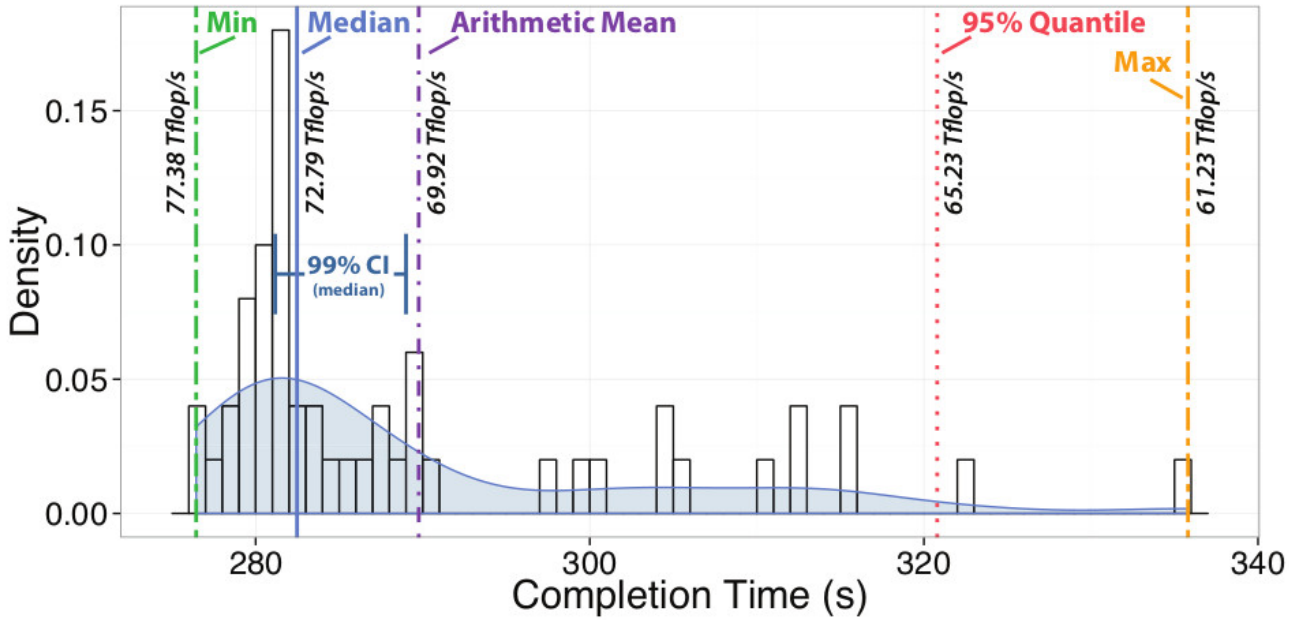
TECHNISCHE UNIVERSITÄT DARMSTADT

- Performance varies significantly on today's supercomputers
  - A single number may not represent the actual performance well
  - Some measure of spread or stability of the results is required (e.g., standard deviation)
- Report whether the measurement values are deterministic
  - For nondeterministic data, report confidence intervals of the measurement
  - Allows number of required measurements to be computed
- Do not assume normality of collected data (e.g., based on the number of samples) without diagnostic checking

Torsten Hoefler, Roberto Belli: Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results. Proc. of the SC15 Conference, 2015.

Non-normal results

- For non-normal data use rank-based measures
  - Example: median, percentiles

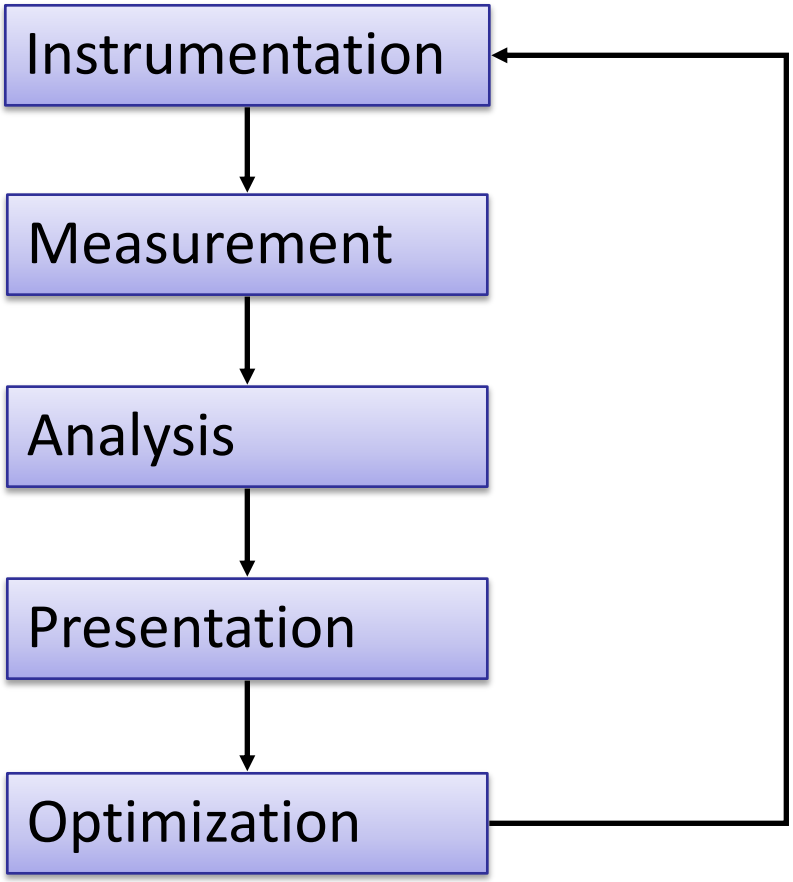


Distribution of completion times for 50 HPL runs

Torsten Hoefler, Roberto Belli: Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results. Proc. of the SC15 Conference, 2015.

Performance tuning cycle

TECHNISCHE UNIVERSITÄT DARMSTADT



Instrumentation techniques

TECHNISCHE UNIVERSITÄT DARMSTADT

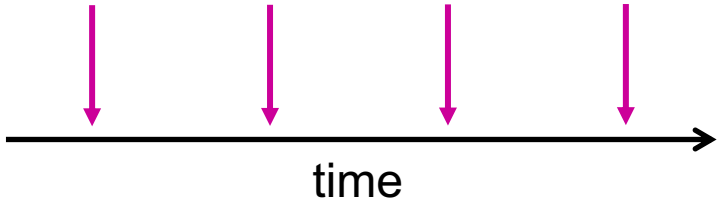
Direct instrumentation

- Measurement code is inserted at certain points in the program
- Example: function entry/exit, dispatch or receipt of messages

```
void foo(int i) {
  func_enter(" foo ");
    for (i = 0; i<end; ++i)
        do_something ( ) ;
  func_exit(" foo ");
}
```

Sampling

- Measurement performed only in certain intervals - usually implemented with timer interrupt



- Based on the assumption that a subset of a population being examined is representative for the whole population

Comparison

TECHNISCHE UNIVERSITÄT DARMSTADT

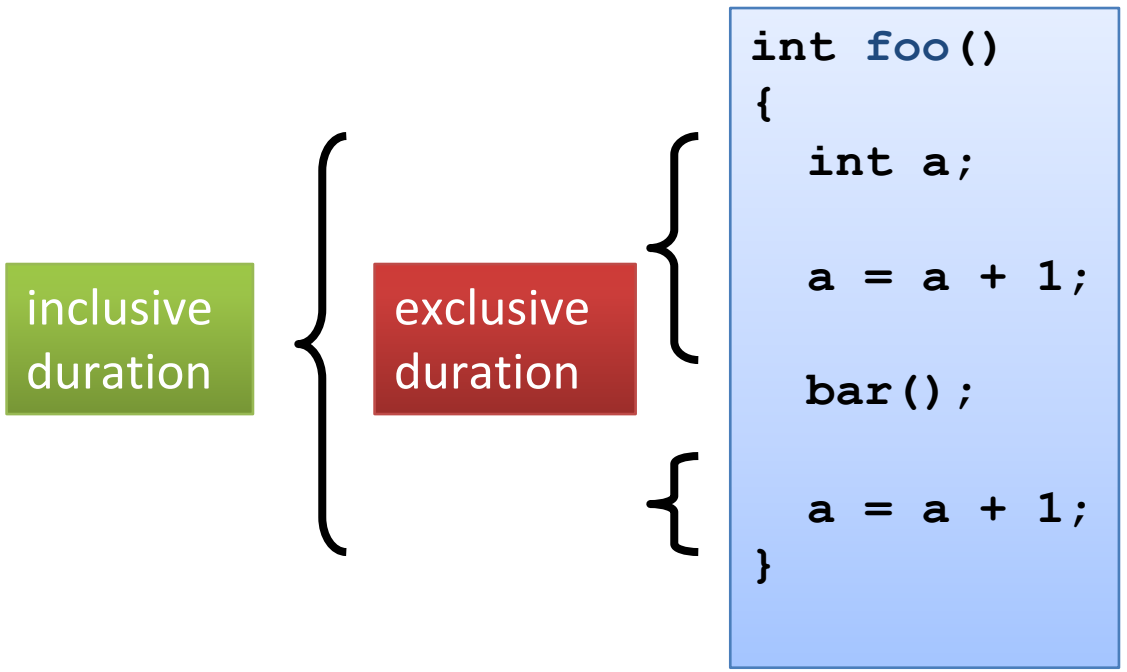
	Direct instrumentation	Sampling
Advantage	Captures all instrumented events	Overhead can be easily controlled
Disadvantage	Overhead more difficult to control	Incomplete information, harder to access program state

Measurement

TECHNISCHE UNIVERSITÄT DARMSTADT

Typical performance data include counts and/or durations of

- Computation
- Communication
- Synchronization
- Input / output
- Hardware events (e.g., cache misses)



Critical issues

TECHNISCHE UNIVERSITÄT DARMSTADT

Accuracy

- Perturbation – measurement alters program behavior
- Example: memory access pattern
- Intrusion overhead – measurement itself needs time and thus lowers performance
- Accuracy of timers, counters

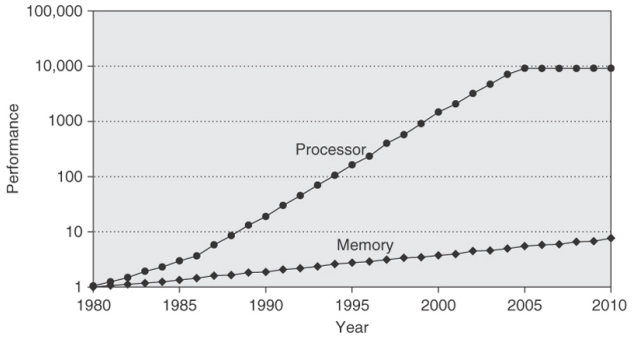
Granularity

- How many measurements
- Pitfall: short but frequently executed functions
- How much information / work during each measurement?

Single-node performance

TECHNISCHE UNIVERSITÄT DARMSTADT

Huge gap between CPU and memory speed



Source: Hennessy, Patterson: Computer Architecture, 5th edition, Morgan Kaufmann

Internal operation of a microprocessor potentially complex

- Pipelining
- Out-of-order instruction issuing
- Branch prediction
- Non-blocking caches

Hardware counters

- Small set of registers that count events
- Events are signals related to the processor's internal function
- Original purpose – design verification and performance debugging for microprocessors
- Idea – use this information to analyze the performance behavior of an application as opposed to a CPU
- Can be measured with PAPI
  http://icl.utk.edu/papi/

Typical hardware counters

TECHNISCHE UNIVERSITÄT DARMSTADT

Cycle count	
Instruction count	All instructions Floating point Integer Load / store
Branches	Taken / not taken Mispredictions
Pipeline stalls due to	Memory subsystem Resource conflicts
Cache	I/D cache misses for different levels Invalidations
TLB	Misses Invalidations

Performance profiling

TECHNISCHE UNIVERSITÄT DARMSTADT

Mapping of aggregated information onto program and system entities

- Time
- Hardware counters
- Software counters
  - #function calls
  - #messages

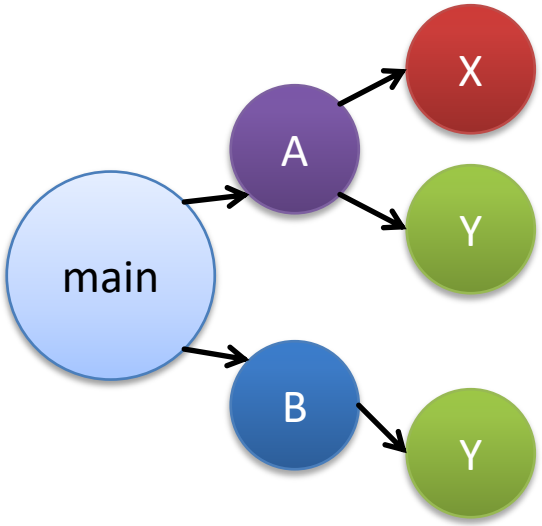
- Functions
- Loops
- Call paths
- […]

- Processes
- Threads
- Nodes
- […]

Call-path profiling

TECHNISCHE UNIVERSITÄT DARMSTADT

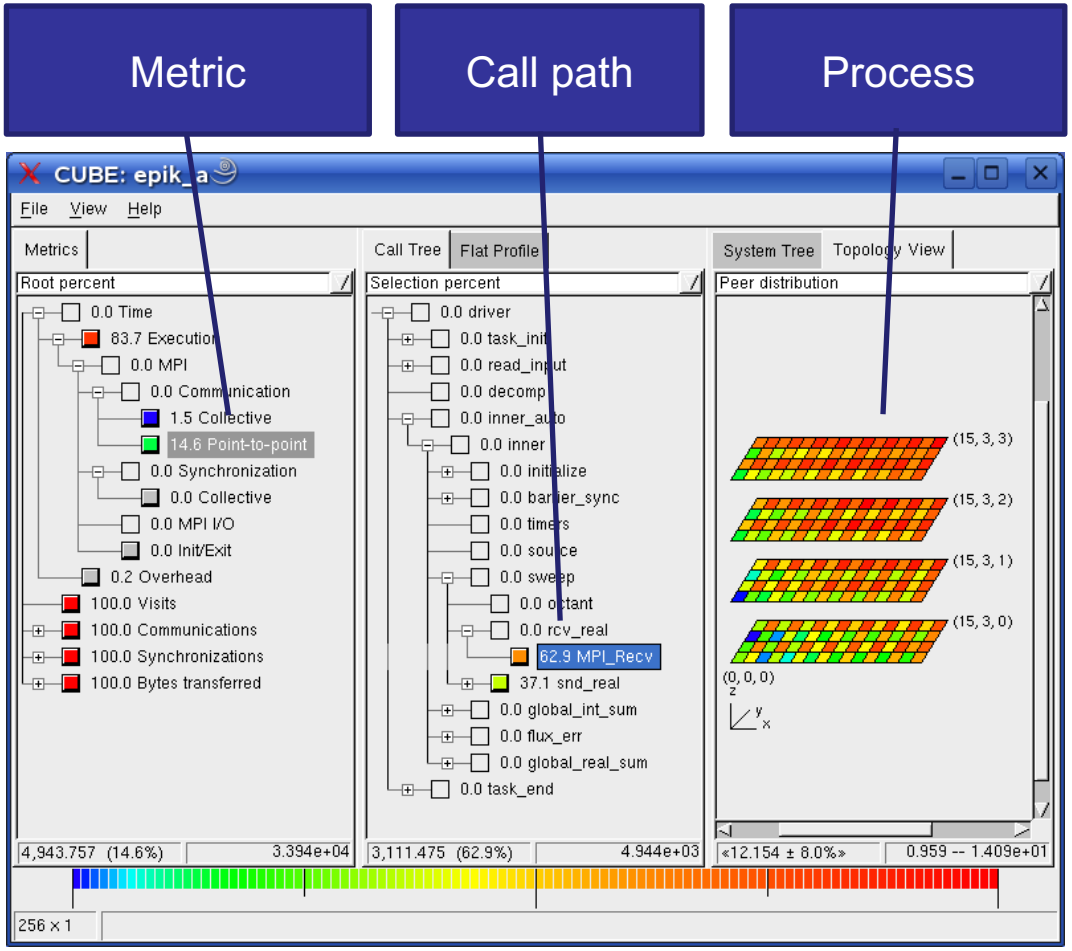
- Behavior of a function may depend on caller (i.e., parameters)
- Flat function profile often not sufficient
- How to determine call path at runtime?
  - Runtime stack walk
  - Maintain shadow stack
    - Requires tracking of function calls




```
main()
{
  A( );
  B( );
}


A( )        B( )
{           {
  X();        Y();
  Y();      }
}
```

Call-path profiling with Scalasca



www.scalasca.org

Event tracing

- Logging performance-critical events with timestamp
- Visualizing log in time-line diagram or analyzing it automatically

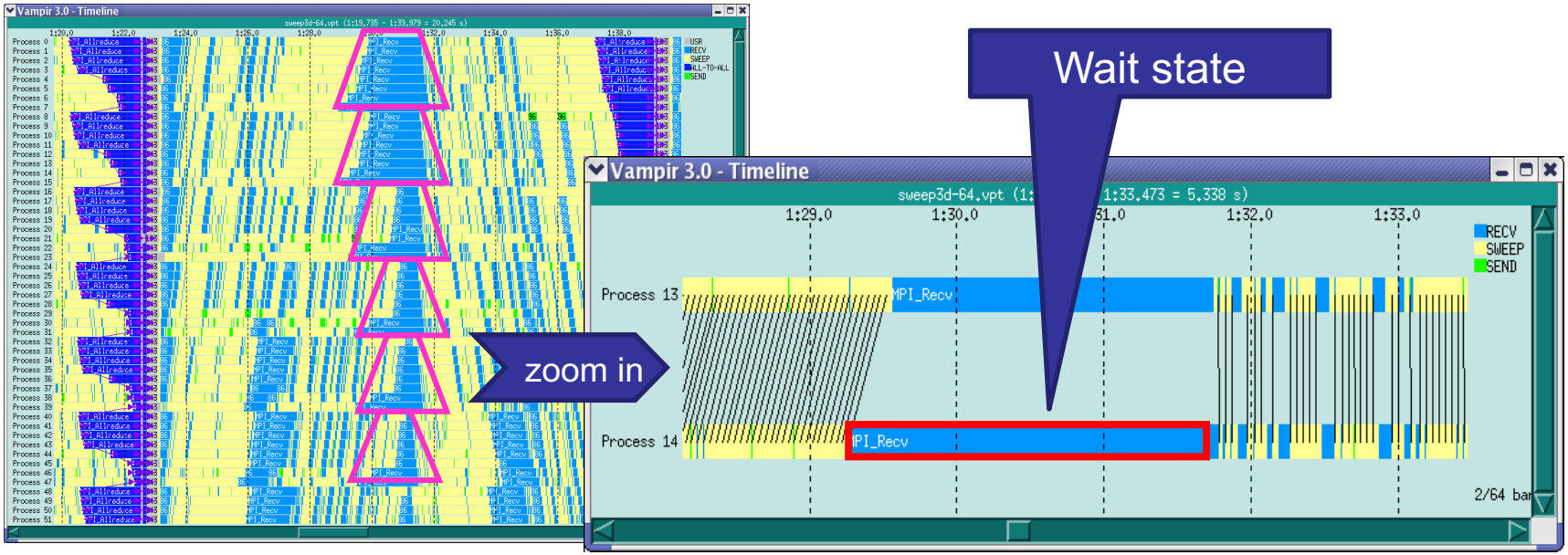
Screenshot of Vampir

Typical events

- Entering and leaving a function; sending and receiving a message

Interactive trace analysis with Vampir

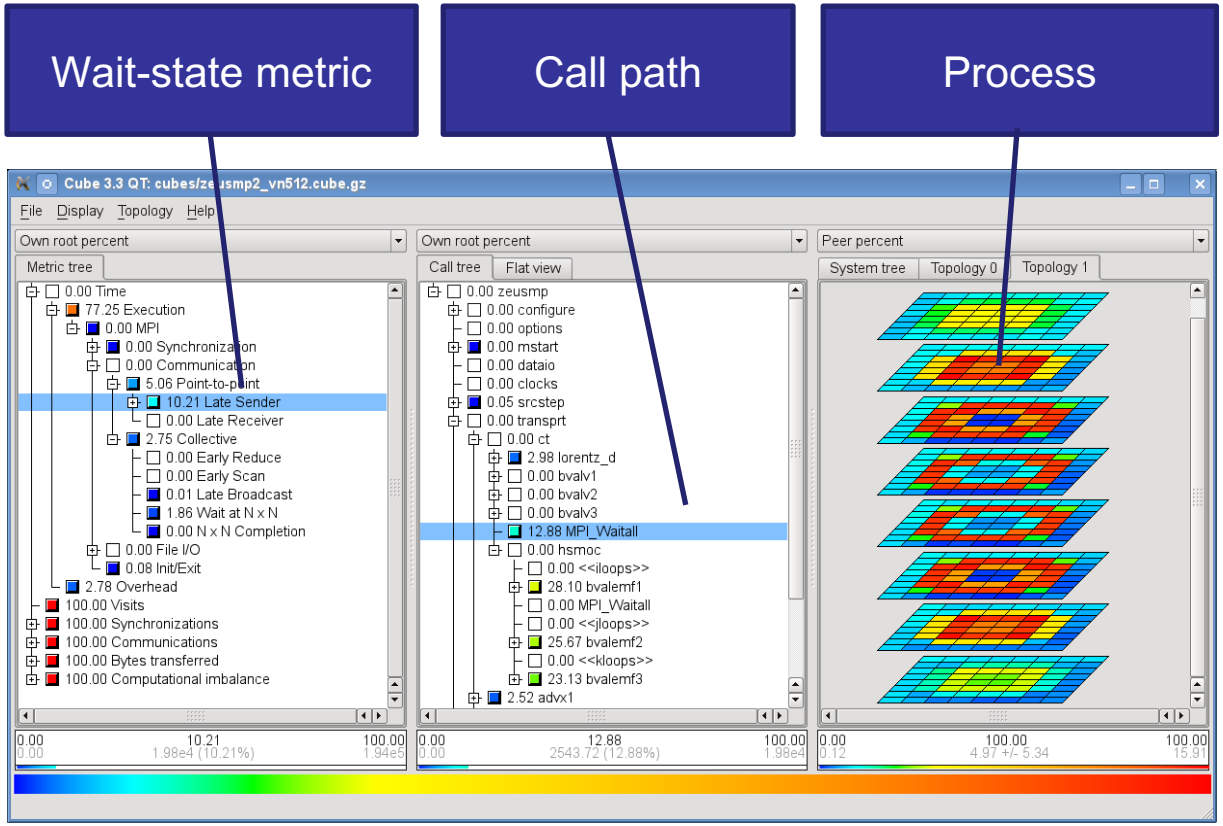
Example: Sweep3D



https://www.vampir.eu

Automatic trace analysis with Scalasca
Example: Zeus-MP/2

TECHNISCHE UNIVERSITÄT DARMSTADT

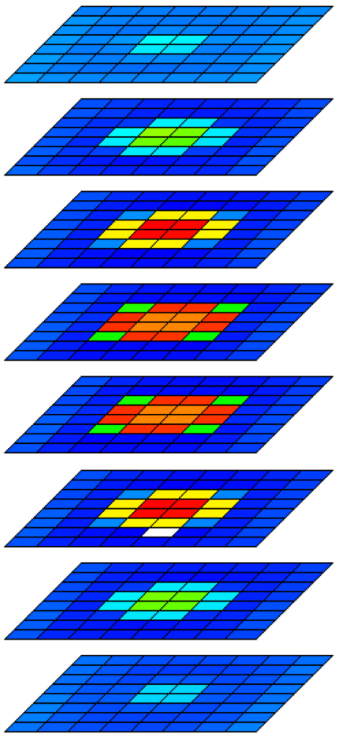


www.scalasca.org

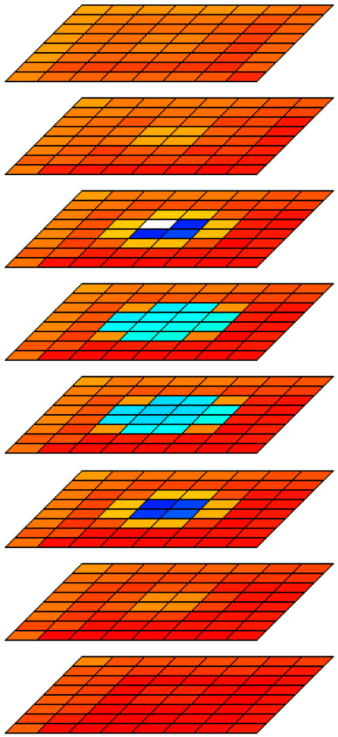
JÜLICH FORSCHUNGSZENTRUM

Origin of wait states
Example: Zeus-MP/2

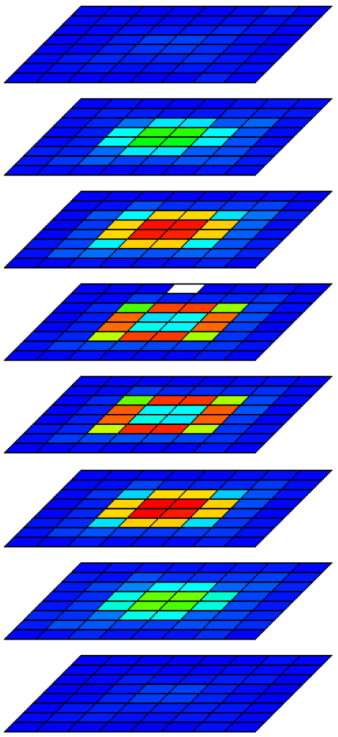
TECHNISCHE UNIVERSITÄT DARMSTADT



Computation



Waiting time

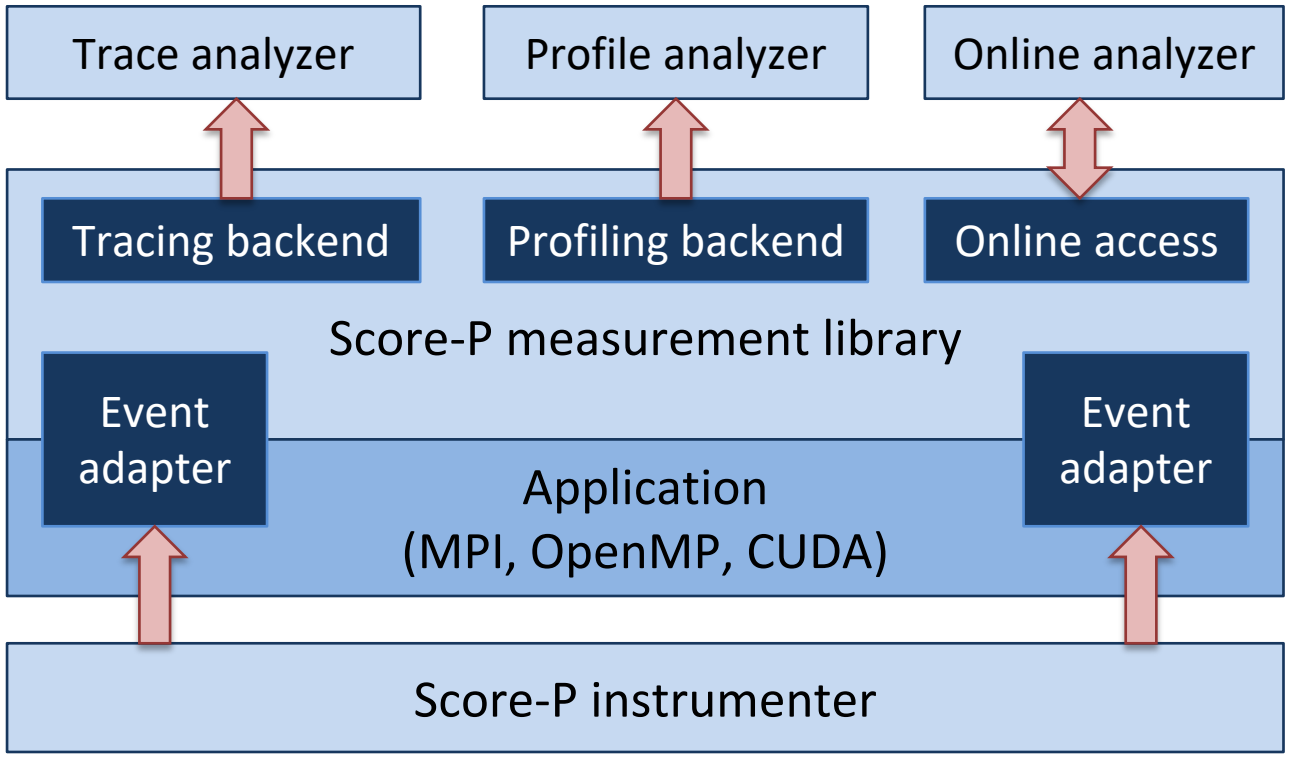


Delay costs

How to obtain profiles and traces?

TECHNISCHE UNIVERSITÄT DARMSTADT

Score-P measurement system

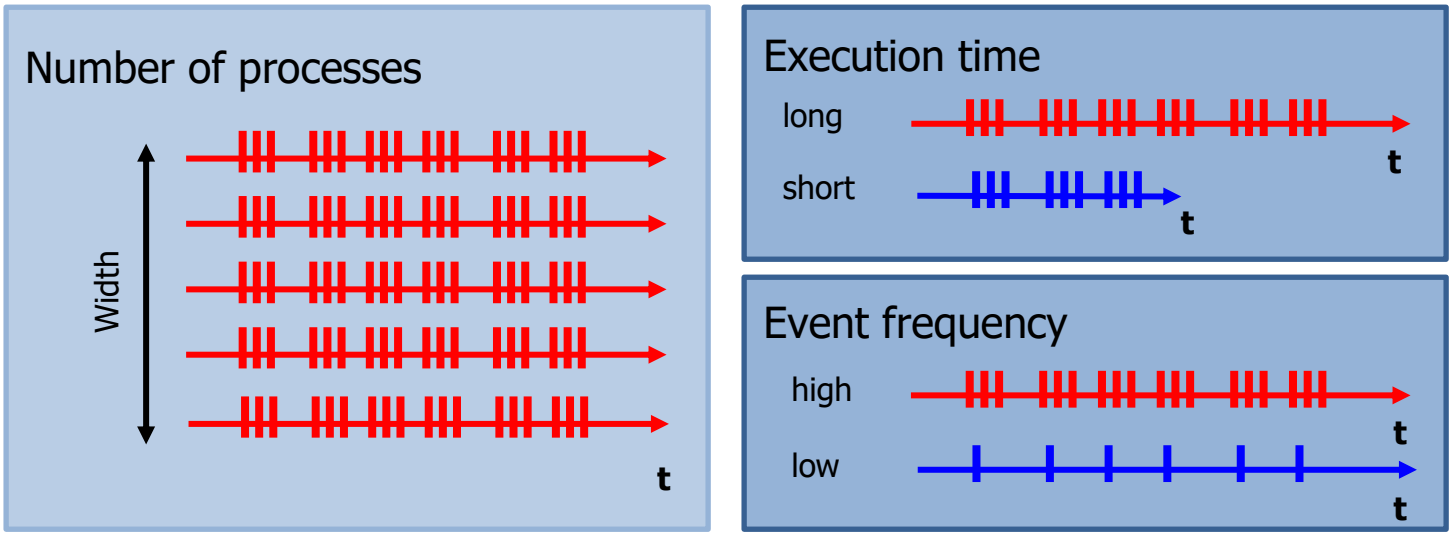


http://www.vi-hps.org/projects/score-p/

Obstacle: trace size

TECHNISCHE UNIVERSITÄT DARMSTADT

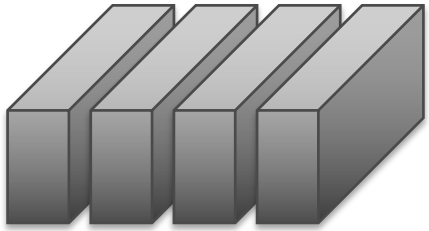
Problem – width and length of event trace

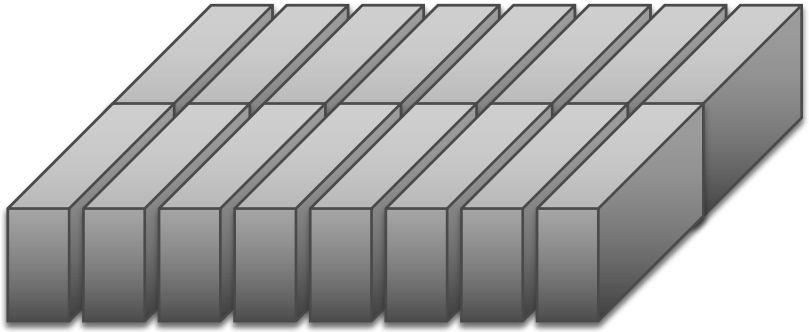


Latent scalability bugs

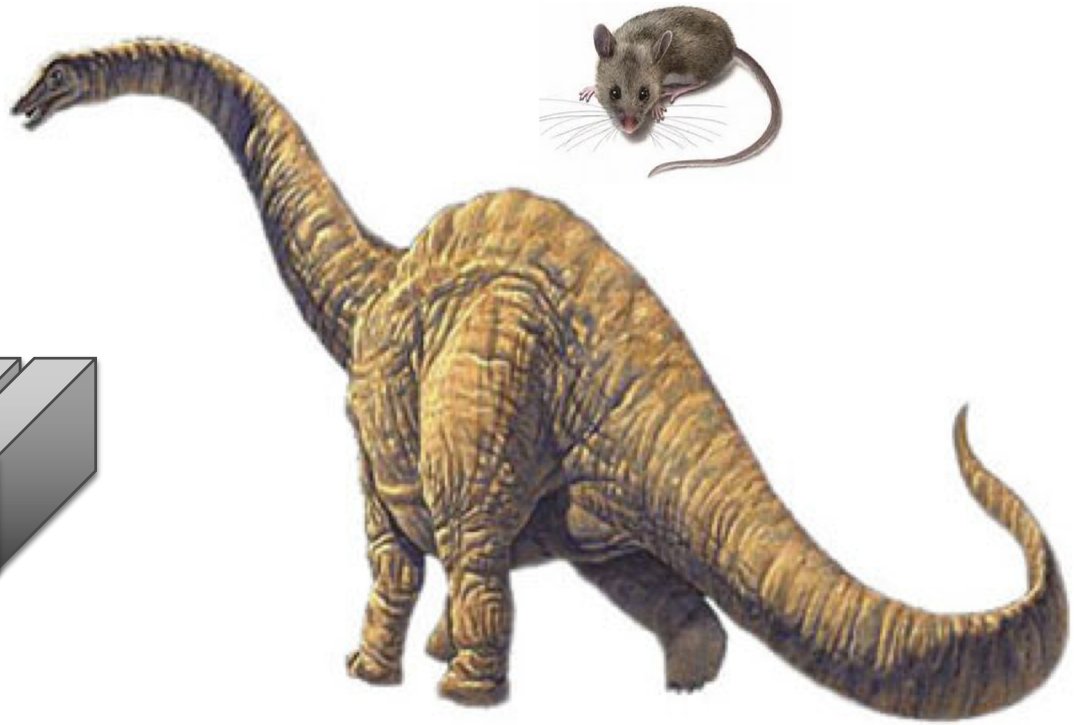
TECHNISCHE UNIVERSITÄT DARMSTADT

System size

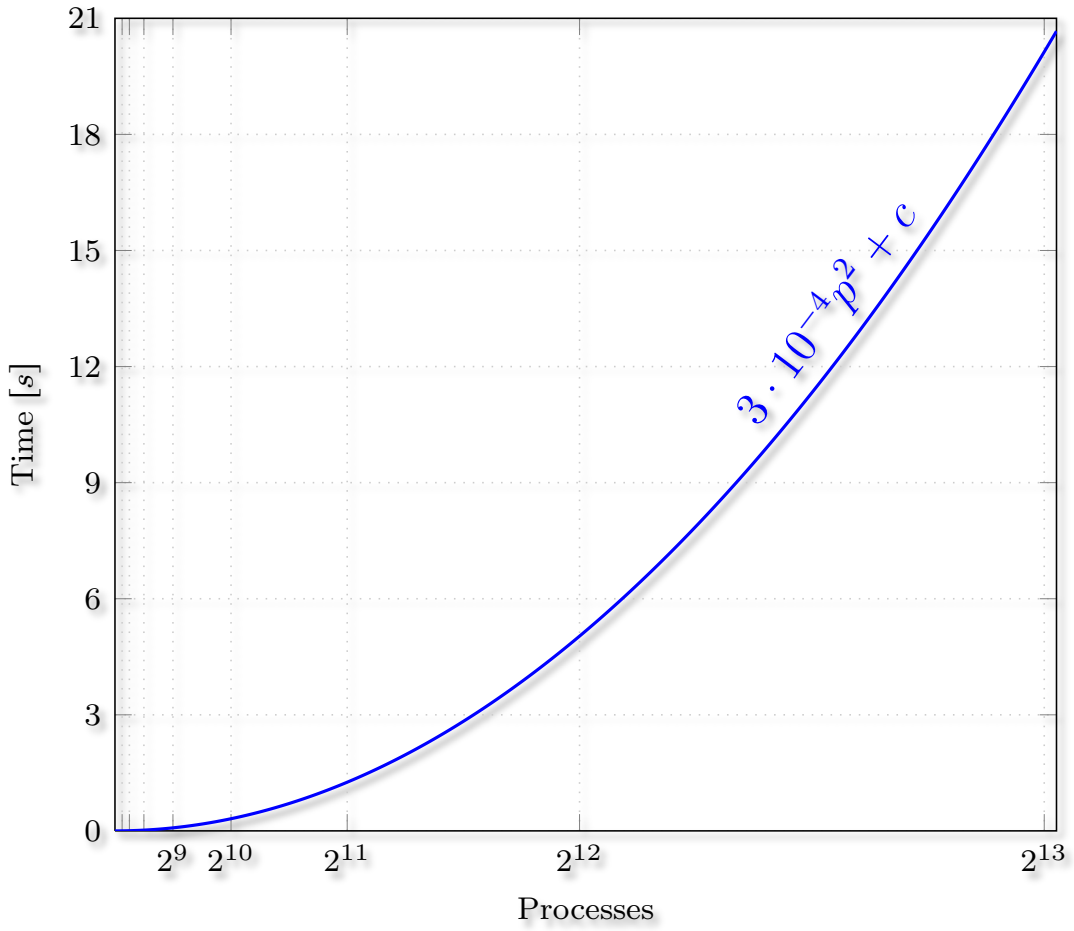




Execution time

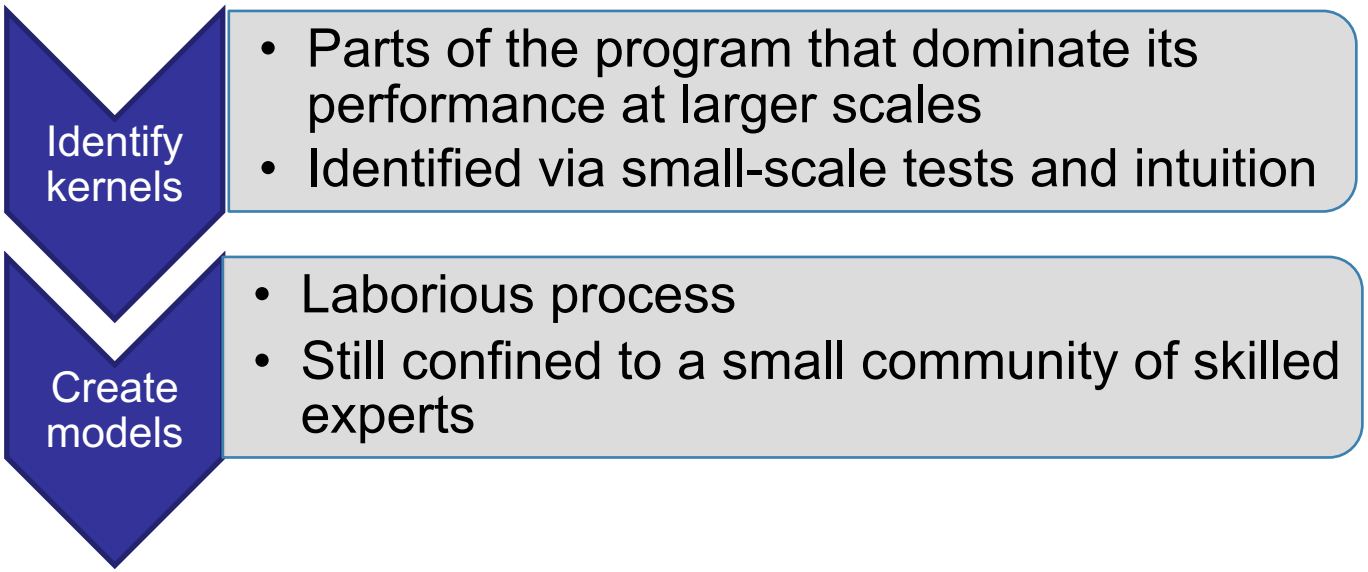


Scalability model



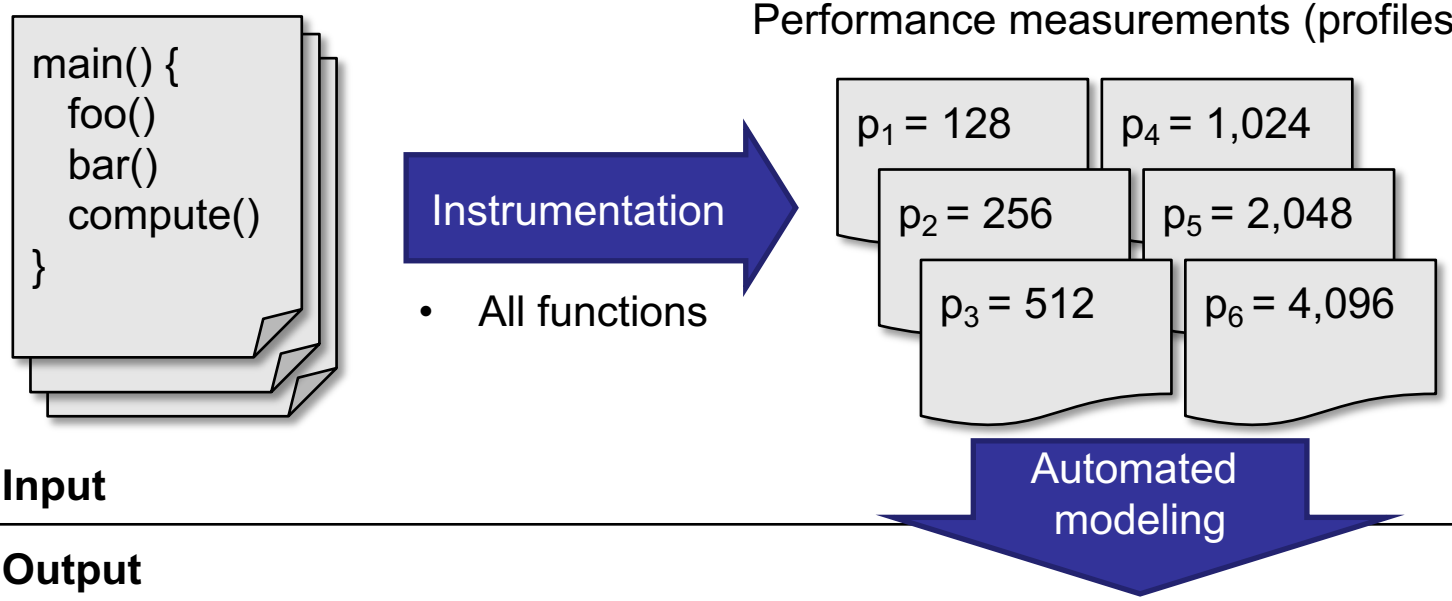
Analytical scalability modeling

TECHNISCHE UNIVERSITÄT DARMSTADT



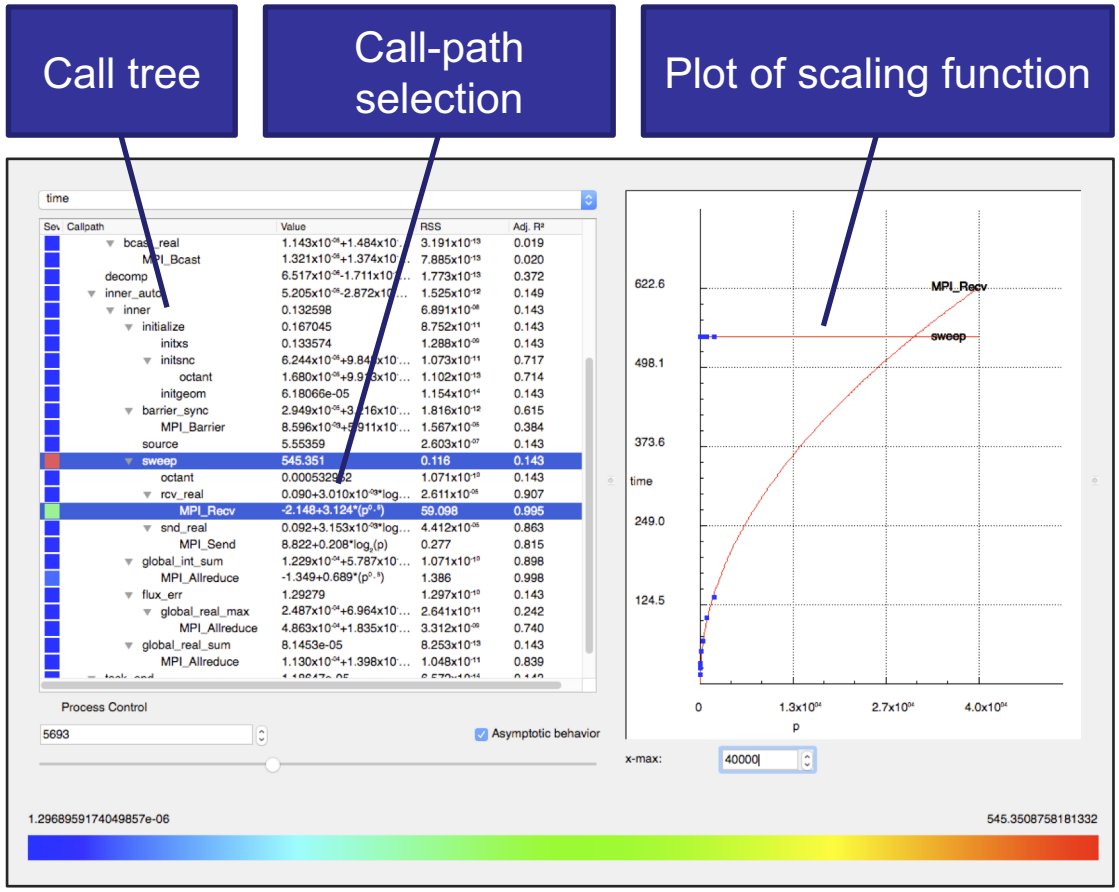
Disadvantages
- Time consuming
- Danger of overlooking unscalable code

Empirical scalability modeling



Function (with calling context)	Model [s] t = f(p)
sweep → MPI_Recv	$4.03\sqrt{p}$
sweep	582.19

Interactive scalability analysis with Extra-P

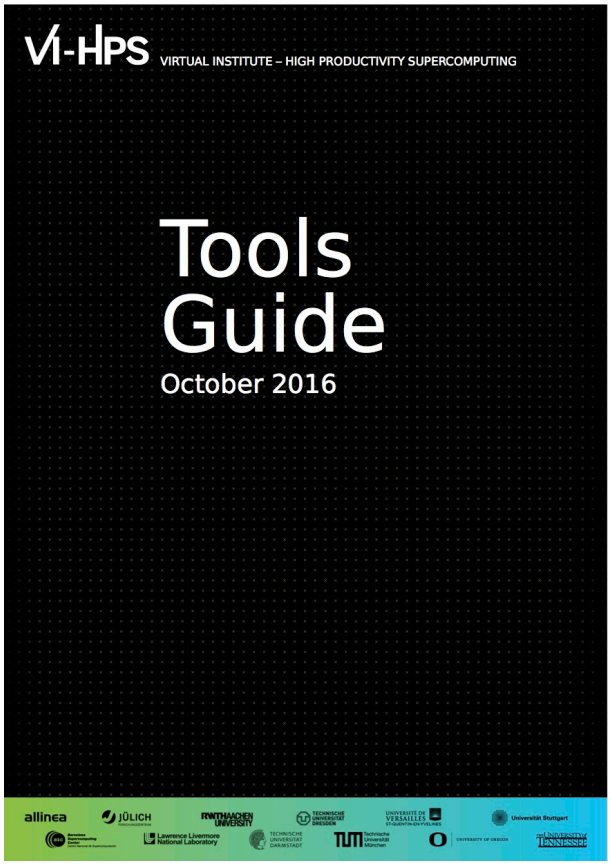


http://www.scalasca.org/software/extra-p

Virtual Institute – High Productivity Supercomputing

TECHNISCHE UNIVERSITÄT DARMSTADT

- Partnership for the development of advanced programming tools for HPC
  - http://www.vi-hps.org
- Activities
  - Tool development and integration
  - Training & support
  - Workshop series on extreme-scale programming tools (in conjunction with SC Conference)
- Publishes guide with information on available tools from VI-HPS



http://www.vi-hps.org/tools/