Performance Models

D. Pleiter

Jülich Supercomputing Centre and University of Regensburg

January 2016

Overview

Content

Model

- **Model** = Simplification of another entity
  - Should contain the characteristics and properties of the modelled entity
  - It is expected to cover only those characteristics that are relevant to a particular task
- **Motivation for using models**
  - Performance modelling
  - Functional modelling and specification
  - Validation and verification
  - ...
- **Types of models**
  - Mathematical models
  - Event simulation
  - Signal level simulations (in soft- or hardware)
  - ...

Performance

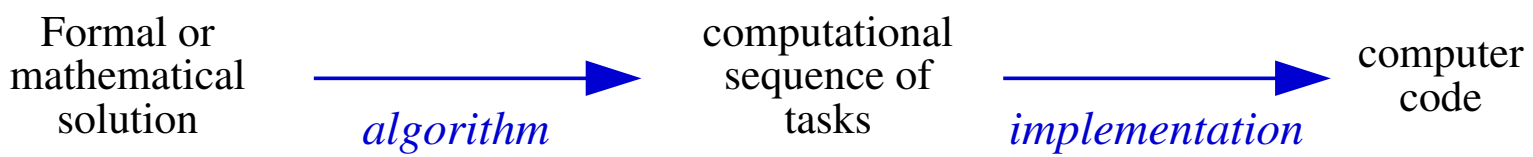
- Measure of performance

$$b = \frac{W}{\Delta t(W)}$$

- **Work-load W** = Amount of work that needs to be executed
  - Also called **problem size**
- **Latency $\Delta t(W)$** = Time required to execute work-load W
  - Ambigiouties in definition of latency:
    - Wall-clock time = latency to complete task
    - Resource occupation time, e.g. CPU time

Performance Model

- **Performance model** = Estimate of latency Δt as a function of different parameters
- Parameters
  - Problem to be solved
  - Chosen algorithm
  - Implementation
  - Computer architecture
  - Problem size



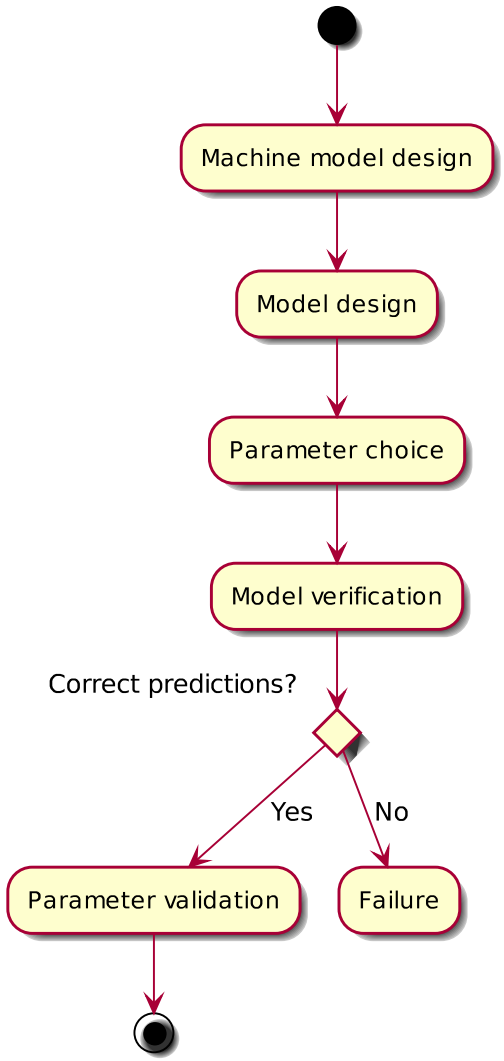
Motivation for Performance Modelling

[Hoefler et al., 2011]

- **Modelling for performance optimisation**
  - Identify optimization opportunities
  - Choose the right algorithm
- **Modelling during software development**
  - E.g., software design following the waterfall model including the steps: analysis, design, implementation, testing, maintenance
- **Modelling during system design and proposal**
  - Create basis for trade-off decisions
- **Modelling during deployment and testing**
  - Compare expected and obtained performance
- **Modelling during operation**
  - Compare expected and operational performance (e.g., to identify performance degradation)

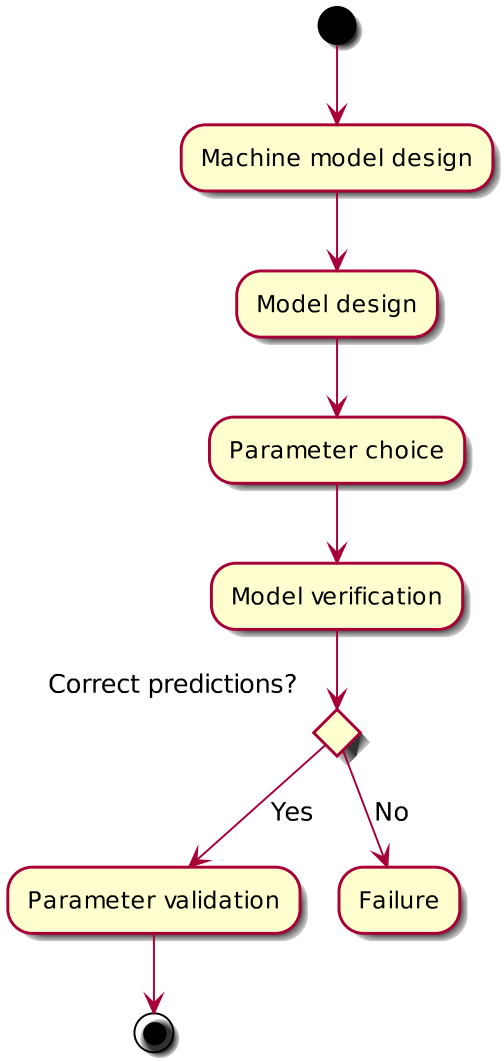
Modelling

- Machine model design
  - Any performance model is based on a simplified model of the hardware (and/or software) architecture
- Model design, e.g.
  - Analytic model
  - Semi-empirical model
- Model parameter choice/determination, e.g.
  - Determine parameters using
  - Fit parameters to performance data
  - Apply training/learning methods



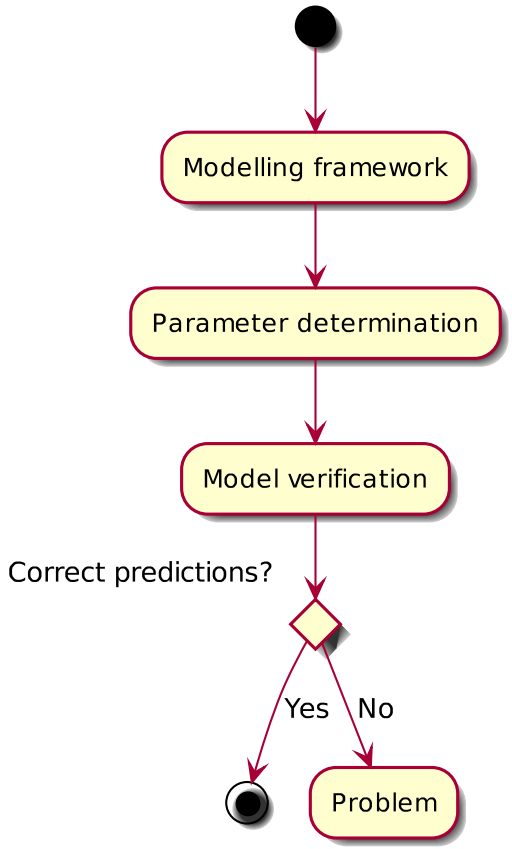
Modelling (cont.)

- Model verification
  - Use model to make predictions
  - Compare predictions to measurements
- Optional: Parameter validation
  - Do parameters have expected size?



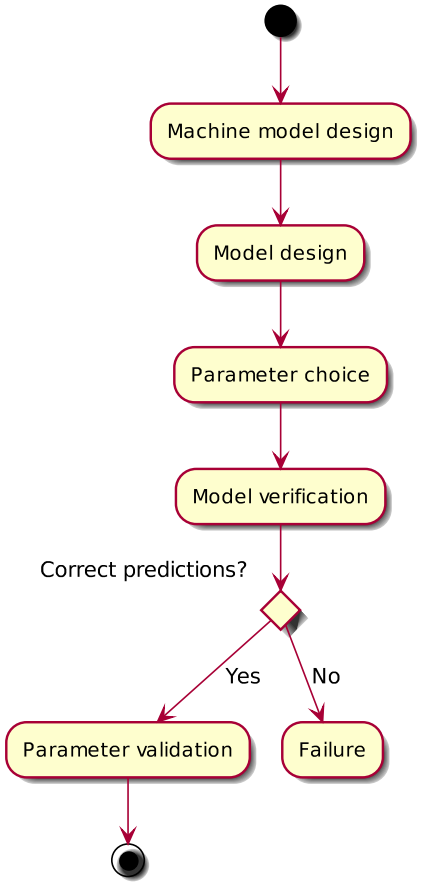
Black Box Modelling

- **Black box modelling** approaches
  - Statistical methods to parametrize performance data
  - Machine learning techniques
- Black box modelling may be independent from architecture and application
  - Rely on pattern matching and curve fitting
- Advantage
  - Simple to apply
- Disadvantages
  - Does not create understanding
    - Failure of validation step merely showcase limitation of approach



White Box Modelling

- **White box modelling** approaches
  - Choice/design of both, machine model and performance model based on knowledge and/or assumptions
  - Example: Knowledge about memory hierarchy
- In case of failures
  - Select new or adjust machine model
  - Select new or adjust performance model
- Disadvantages
  - Much more difficult
- Advantage
  - Creation of new insight
    - Model assumptions are challenged



(White Box) Performance Model Classification

- **Analytic performance models**
  - Performance is expressed with purely analytic expressions
  - Parameters are fixed from data-sheet information
- **Semi-empirical performance models**
  - Performance is expressed with purely analytic expressions
  - Parameters are determined through fits to performance data
- **Simulation models**
  - Simulates behaviour of machine model for given input stimuli
  - May include cycle-accurate simulation of digital logic

Modelling Different Quantities

- Which quantities to model?
  - Latency Δt
    - Time-to-solution in case of latency of full application
  - Energy-to-solution ΔE
  - Resource utilization
    - Example: Number of cache misses
  - Application requirements
    - Example: Number of floating-point operations
- Classification of quantities involved in modelling
  - Deterministic quantities
    - Number of floating-point operations
  - Statistical quantities
    - Latency Δt

Content

Proxy Architectures

[Ang et al. (Sandia/LBNL), 2014]

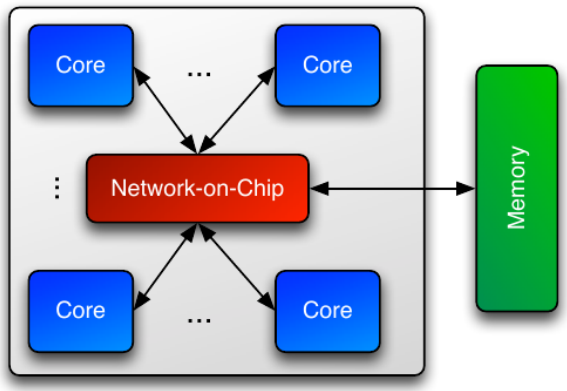
- **Proxy architecture** = Parameterized version of an abstract machine model
  - Parameter set limited to describe hardware capability and capacity of relevant components
  - Example for such parameters
    - Number of cores
    - Memory bandwidth, floating-point throughput
    - Capacity of caches
- Why parametrized model?
  - Model different version of same component, e.g. processor
  - Explore architectural design space
- Different scopes
  - Processor level
  - Node level
  - System level
- Different level of abstractions
  - As little details as possible, as many details as necessary

Proxy-Architectures for Processors

[Ang et al. (Sandia/LBNL), 2014]

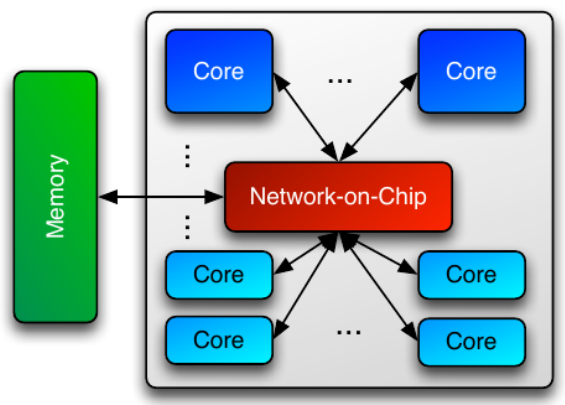
Homogeneous multi-core processor

- Multiple cores of same type
- Processor-internal network
- External memory



Heterogeneous multi-core processor

- Multiple cores of different type
- Processor-internal network
- External memory

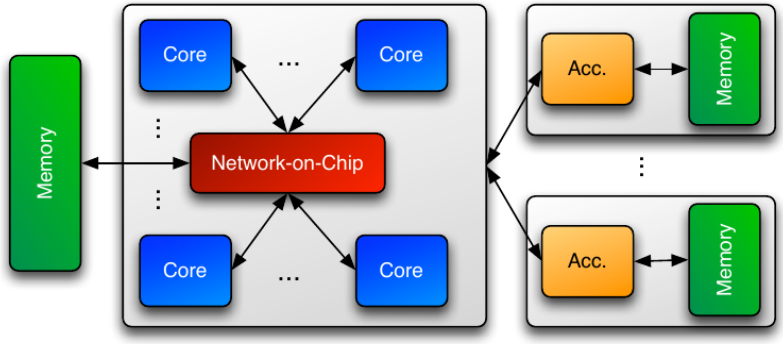


Proxy-Architectures for Processors (cont.)

[Ang et al. (Sandia/LBNL), 2014]

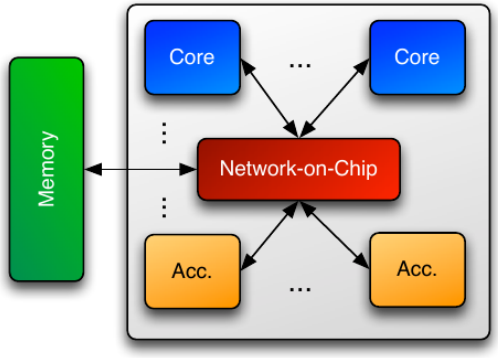
Multi-core processor plus discrete accelerator

- Example: CPU + GPU



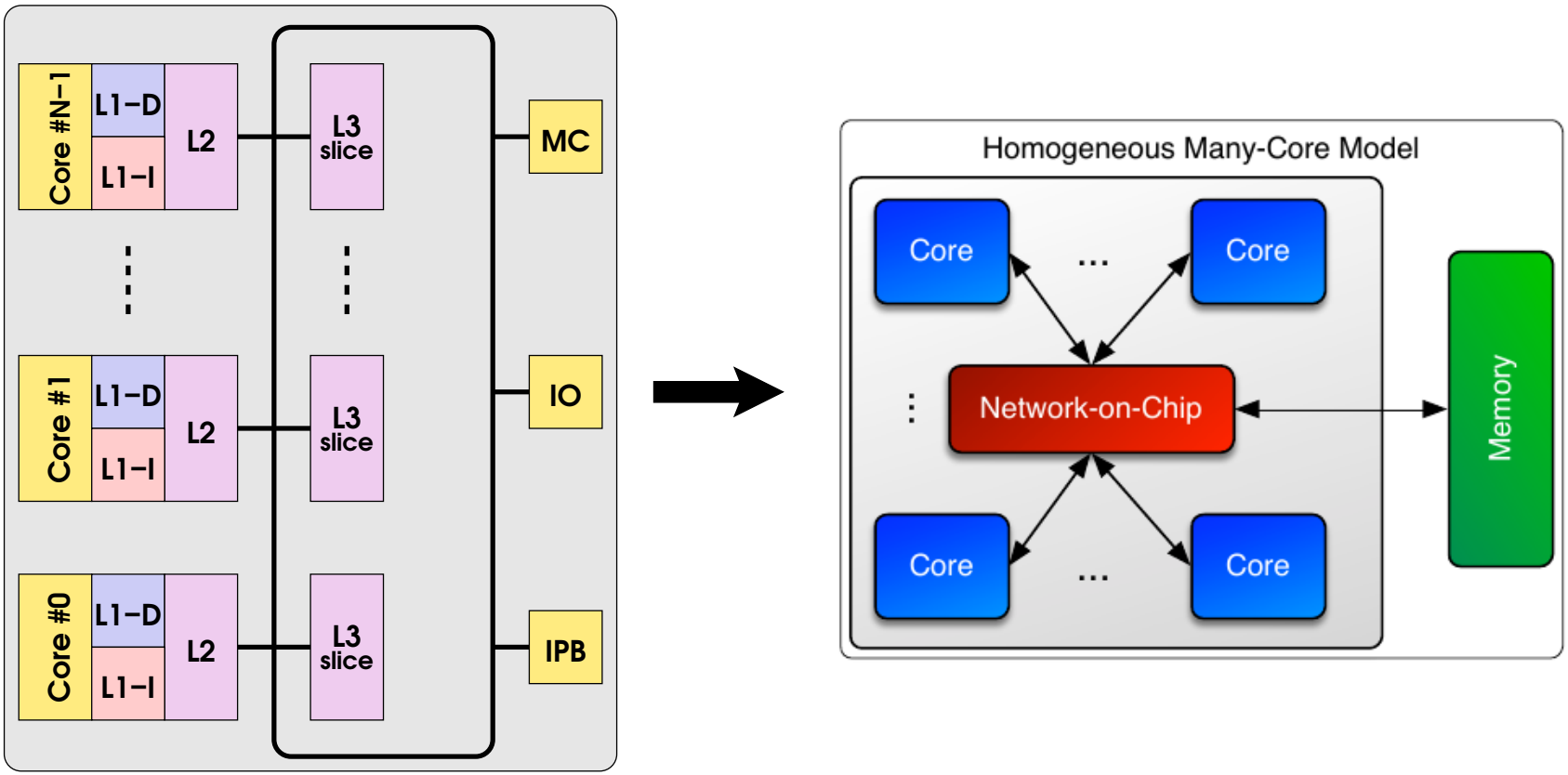
Multi-core processor plus integrated accelerators

- Example: AMD APU



Proxy-Architecture for Intel Xeon E5

[Ang et al. (Sandia/LBNL), 2014]

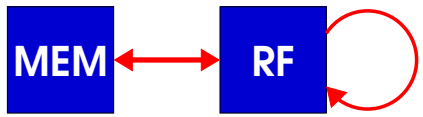


Abstract Machine Model

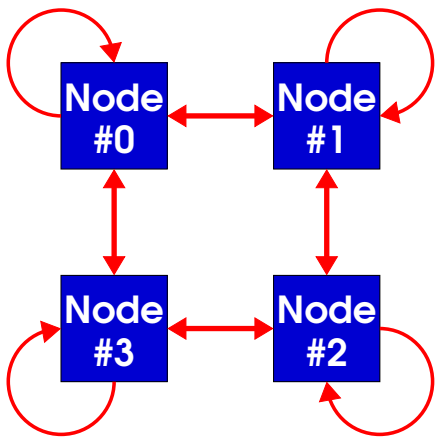
- **Machine** = set of interconnected devices
  - **Storage devices**
  - **Processing/transport devices**
- Storage devices
  - Examples
    - External memory
    - Register file
  - Parameter: capacity
- Processing/transport devices
  - Examples
    - Arithmetic pipeline
    - Memory bus
  - Possible parameters
    - Bandwidth/throughput
    - Start-up latency

Abstract Machine Model (cont.)

- **Graphical representation**
  - Vertices = storage devices
  - Edges = Processing/transport devices
- **Example 1: Processor architecture**
  - Storage devices
    - External memory (MEM)
    - Register file (RF)
  - Processing/transport devices
    - Memory bus
    - Processing pipeline



- **Example 2: Multi-node system**
  - Storage devices
    - Node-level memory
  - Processing/transport devices
    - Processing pipelines
    - Intra-node network links



Content

Report on Performance Measurements

- Avoid reporting of speed-up
  - Problem: Reference point often unclear
  - Exception: Scaling analysis
- Report units unambiguously
  - Better to use “Flop/s” instead of “Flops”
  - Base-2 vs. base-10 number qualifiers
    - Follow recommendations of IEC 60027-2 standard and use “i” to indicate base-2

$$1\,\text{kByte/s} = 10^3\,\text{Byte/s} = 1000\,\text{Byte/s}$$
$$1\,\text{kiByte} = 2^{10}\,\text{Byte} = 1024\,\text{Byte}$$

- Report latency (Δt) instead of performance ($b = W/\Delta t$)
  - Often functional behaviour of latency is easier to understand
  - Example: latency often depends linearly on work-load W

$$\Delta t = a_0 + a_1 W + \cdots \quad \Rightarrow \quad b = \frac{W}{a_0 + a_1 W + \cdots}$$

Analysis of Statistical Measurements

Check distribution of statistical quantities

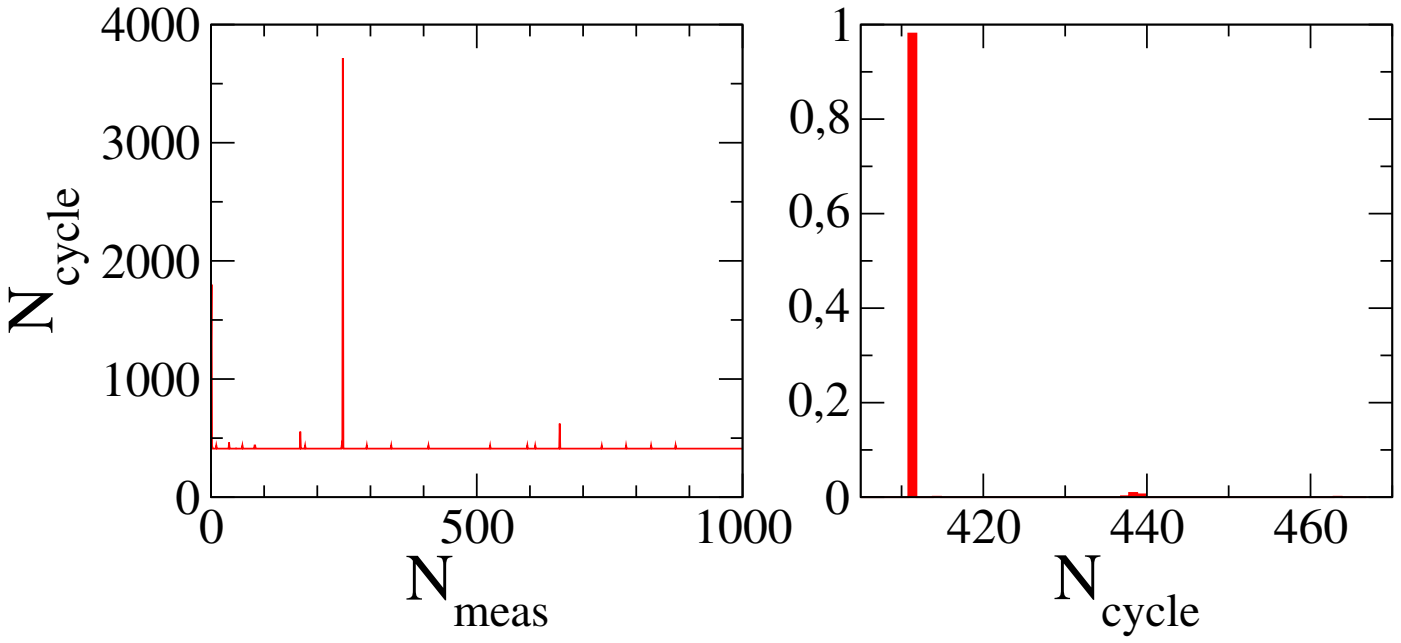
- Reasons for statistical fluctuations
  - Non-deterministic hardware (e.g. caches, memory management)
  - Non-uniform data paths (e.g. NUMA nodes)
  - Interference with other processes or (kernel-)threads (“OS jitter”)
  - Dynamic change of clock speed

☞ Do not expect normal distribution

Analysis of Statistical Measurements (cont.)

Check distribution of statistical quantities (cont.)

- Example for non-normal distribution
  - Execute a sequence of M loads and stores of double precision floats N times (here $N = 10$, $M = 16$)
  - Tested processor: TI OMAP4 with ARM Cortex-A9 cores



Analysis of Statistical Measurements (cont.)

Carefully select which central value to report

- Arithmetic mean $A(x) = (1/n) \sum_{i=1}^{n} x_i$
  - Use only in case of symmetric distributions
- Harmonic mean $H(x) = n / \sum_{i=1}^{n} (1/x_i)$
  - Use only in case of ratios where denominator is primary quantity
- Minimum value
  - Useful to analyse hardware related performance data
  - Example: data transfer
    - Latency bound by below due bandwidth bounds
    - Latency not bound from above
- Median value
  - Robust metric as long as number of outliers is small enough
  - Easy to determine with knowledge of underlying distribution

Content

Semi-Empirical Performance Modelling

[Hoefler, 2011]

Procedure

- **Analytic steps**
  - Identify application kernels
  - Identify input parameters that influence runtime
    ☞ Parametrization of work-load W
  - On basis of prior knowledge formulate scaling formulae describing dependence of latency $\Delta t(W)$ as function of work-load W
- **Empirical steps**
  - Measure $\Delta t(W)$ for different W
  - Fit scaling formulae to measurements
- **Model verification**
  - Compare predicted and measured values
- **Parameter validation**
  - Check fitted parameters for plausibility

Semi-Empirical Performance Modelling: Example

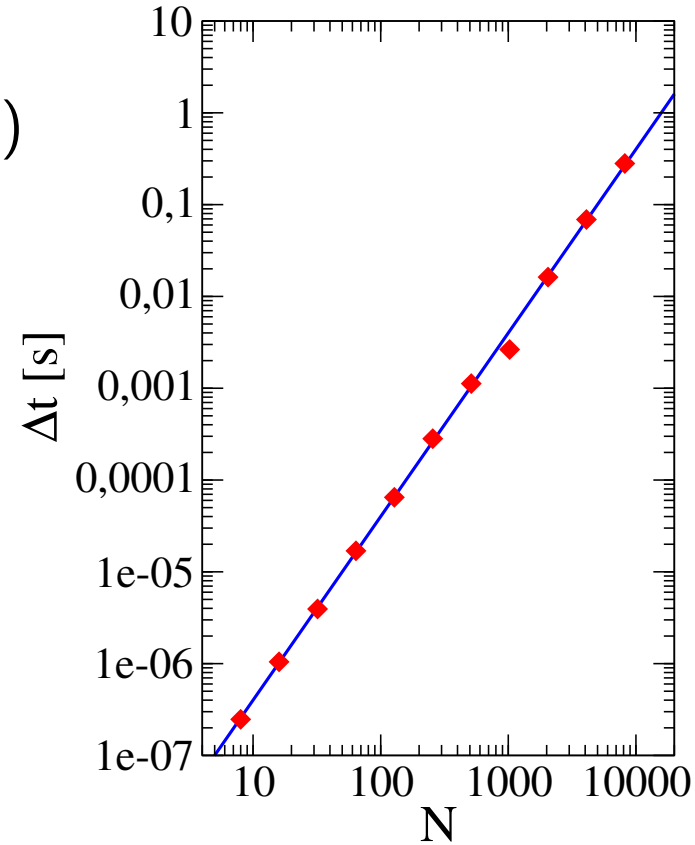
- Matrix-vector multiplication

$$y_i \leftarrow \sum_{j=0}^{N-1} A_{ij}\, x_j \quad (i = 0, \ldots, N-1)$$

- Ansatz

$$\Delta t(N) = a_0 + a_2 N^2$$

- Results obtained on Intel i7-2620M using a single thread without compiler optimization
  - $a_0 \simeq 0\,\mathrm{ns}$, $a_2 \simeq 4\,\mathrm{ns}$
- Plausibility analysis
  - Clock $f = 2.7\,\mathrm{GHz} \Rightarrow 1/f = 0.4\,\mathrm{ns}$
  - Need about 10 cycles per (i, j)



Information Exchange Function

- Each task on the computer eventually implies that information is transferred from a storage device x to a storage device y
  - Machine model = abstract machine model
- **Information Exchange Function**:

$$I^k_{x,y}(W) \quad = \quad \text{data transferred between computer sub-systems for specific computational task } k$$

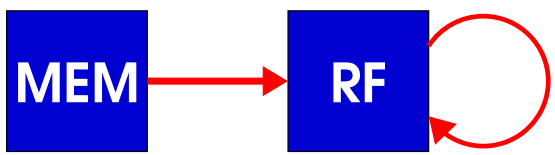
x ... source storage device (e.g. memory)
y ... destination storage device (e.g. register file)
W ... workload

Information Exchange Function (cont.)

- Consider scalar product

$$c \leftarrow \sum_{i=0}^{N-1} a_i \, b_i$$

  - a_i, b_i, c ... double-precision numbers (64 bit)
  - Work-load W parametrized by vector length N
- Abstract machine model

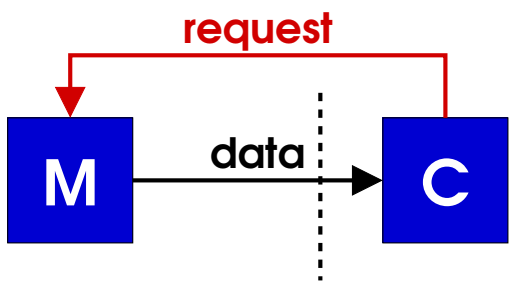


- Information exchange functions

$$\begin{aligned} I^{\mathrm{sp}}_{\mathrm{mem,rf}}(N) &= 16N\,\mathrm{Byte}, \\ I^{\mathrm{sp}}_{\mathrm{rf,rf}}(N) &= 2N\,\mathrm{Flop}. \end{aligned}$$

Bandwidth or Throughput vs. Latency

- Definition **bandwidth or throughput**
  - Bandwidth = amount of items passing a particular interface per time unit
  - Throughput = amount of work done in a given time
- Definition **latency (or response time)** Δt = Time between events indicating start of an task and the event signaling its completion
- Example: Memory read operation

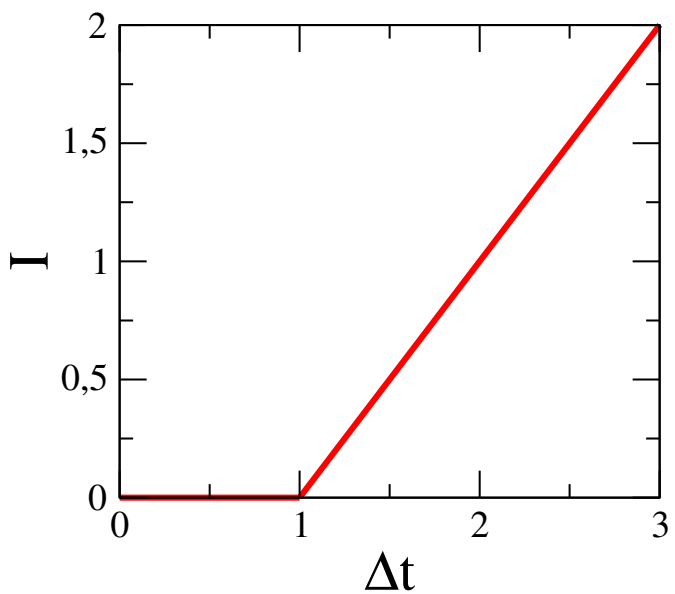


  - Bandwidth: Amount of data passing memory interface per time unit
  - Latency: Time from starting memory read operation until last data item arrived in register file

Latency-Bandwidth Model

- Simple model describing **latency** Δt as a function of the information exchange I
- Model parameters:
  - **Start-up latency** λ = Time between event trigger and start of arrival of response
  - **Bandwidth** β: Parametrizes time to exchange information I
- Model:

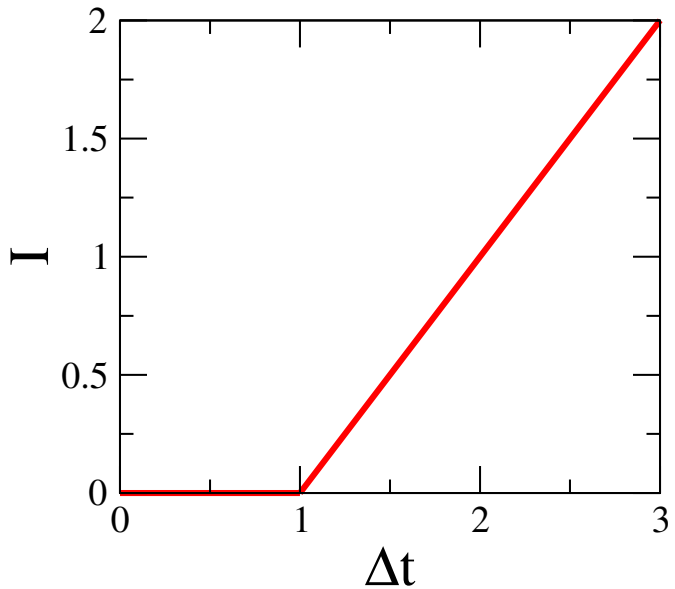
$$\Delta t(I) = \lambda + \frac{I}{\beta}$$

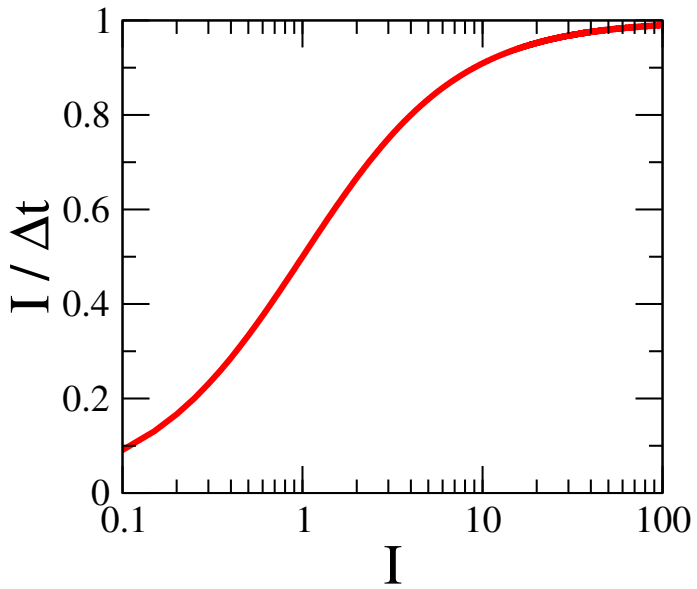


Latency-Bandwidth Model (2)

- **Effective bandwidth** $\tilde{b} = I/\Delta t(I)$
- Example: Read I Bytes from memory or network

$$\Delta t(I) = \lambda + \frac{I}{\beta} \quad \Rightarrow \quad \tilde{b}(I) = \left(\frac{\lambda}{I} + \frac{1}{\beta}\right)^{-1}$$



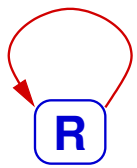


Information Exchange Model: Latency Predictions

Ansatz to predict latency:

$$\Delta t^k_{x,y} \simeq \lambda_{x,y} + \frac{I^k_{x,y}}{\beta_{x,y}}$$

Example:



arithmetic unit

register file

- x, y = R
- $\beta_{R,R}$ = throughput arithmetic unit
- $I^k_{x,y}$ = number of input (or output) operands

Beware of limitations of this ansatz:

- Transfer mechanism may depend on task size N
- Bandwidth changes due to resource congestion is ignored
- ...

Information Exchange Model: Simple Example

- Example operation: $y_i \leftarrow \alpha \cdot x_i$ $(i = 1, \ldots, N)$
  x_i, y_i ... vectors of single precision floating-point numbers
  α ... single-precision floating-point number
- Information exchange for each i:

load x_i	4 Bytes
multiply α and x_i	1 Flop
store y_i	4 Bytes

- Assume the following hardware parameters:

Memory bandwidth β_{mem}	1 Byte/clock cycle
Floating-point unit throughput β_{fp}	1 Flop/clock cycle

- Latency predictions:

Δt_{mem}	$N \cdot 8$ clock cycles
Δt_{fp}	$N \cdot 1$ clock cycles

Information Exchange Model: Simple Example (cont.)

Latency for full operation:

- No overlap of memory load/store and arithmetic operations:

$$\Delta t(N) = \Delta t_{\text{mem}} + \Delta t_{\text{fp}} = N \cdot 9$$

- Perfect overlap of memory load/store and arithmetic operations:

$$\Delta t(N) = \max\left(\Delta t_{\text{mem}}, \Delta t_{\text{fp}}\right) = N \cdot 8$$

☞ **Memory (bandwidth) limited** problem

- Opposed to **compute (throughput) limited** problems

Memory Bandwidth

- Definitions
  - Measured bandwidth b_{mem}
    - Amount of transferred data l_{mem} over measured latency Δt
  - Latency-bandwidth model parameter β_{mem}
  - Effective bandwidth $\tilde{b}_{\text{mem}}$
  - **Nominal hardware bandwidth** B_{mem}
    - Data rate on hardware link/bus
- Relations
  - Upper limit for measured bandwidth: $b_{\text{mem}} \leq B_{\text{mem}}$
  - Upper limit for effective bandwidth: $\tilde{b}_{\text{mem}} \leq \beta_{\text{mem}}$
  - Note: $\beta_{\text{mem}} \leq B_{\text{mem}}$ is not guaranteed
    - But expected if bandwidth-latency model is a good model

Semi-Empirical and Information Exchange Modelling

- Ansatz: Use Information Exchange to create semi-empirical ansatz, e.g.

$$\Delta t = a_0 + a_1 I(W)$$

  - a_1 has dimension of inverse throughput/bandwidth
- Advantages
  - No need to determine start-up latency λ and asymptotic bandwidth β
  - Allow for easy comparison with hardware performance parameters

Example

- Double-precision matrix-vector multiplication

$$y_i \leftarrow \sum_{j=0}^{N-1} A_{ij}\, x_j \quad (i = 0, \ldots, N-1)$$

- Information exchange

$$\begin{aligned} I_{\text{mem,rf}} &= (N^2 + 2\,N)\,8\,\text{Byte} \\ I_{\text{rf,rf}} &= (2\,N - 1)\,N\,\text{Flop} \end{aligned}$$

- Re-use previously presented results for Intel i7-2620M and comparison to nominal throughput/bandwidth B

Storage devices	$1/a_1$	B
mem,rf	$2\,\text{GByte/s}$	$21.3\,\text{GByte/s}$
rf,rf	$0.5\,\text{GFlop/s}$	$43.2\,\text{GFlop/s}$