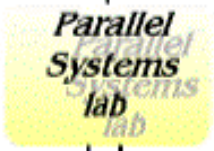


A Refinement Strategy for a User-Oriented Performance Analysis

Jan Lemeire, Andy Crijns, John Crijns, Erik Dirkx

Parallel Systems lab,
Vrije Universiteit Brussel (VUB), Brussels, Belgium
jan.lemeire@vub.ac.be
http://parallel.vub.ac.be

Talk at EuroPVM/MPI Conference, Budapest, Hungary, September 22th 2004

Parallel Performance Analysis

Current challenge

- serve user understandable results with a minimum of learning overhead

Goal

- for optimization
  - detect bottleneck
  - detect ‘abnormal’ situations
  - impact of each aspect
- prediction

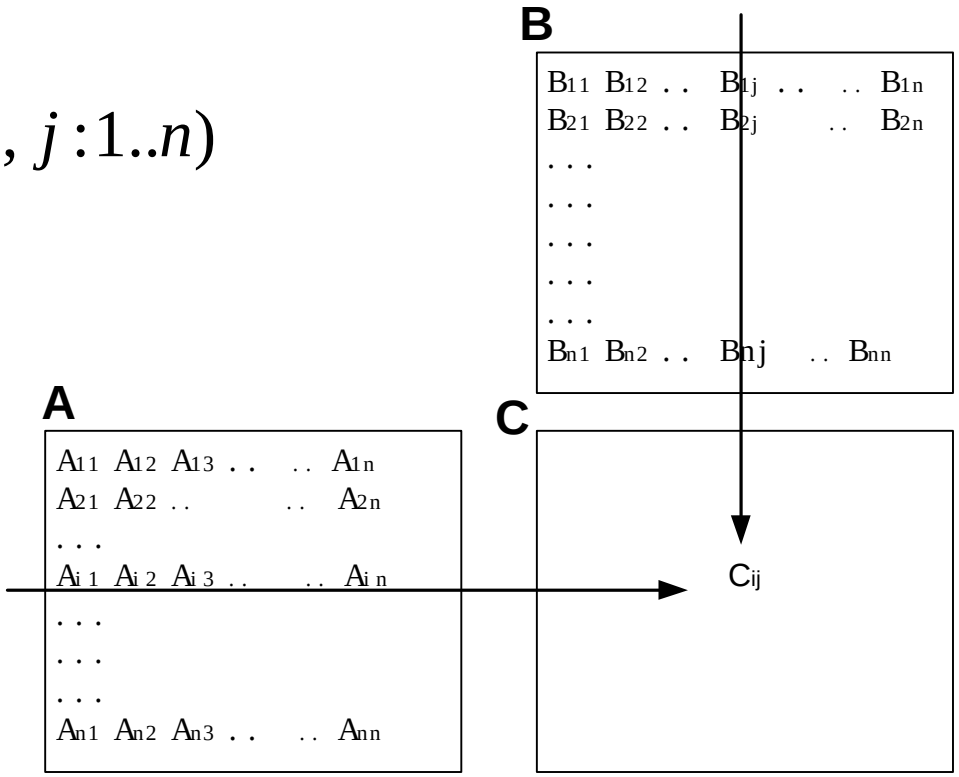
share with you our exploration towards an effective methodology

Matrix Multiplication

$$C = A \times B$$

$$C_{ij} = \sum_{k=1}^{n} A_{ik}.B_{kj} \quad (i, j : 1..n)$$

$$T_{computation} = d_{mm}.n^3$$



Parallel Matrix Multiplication

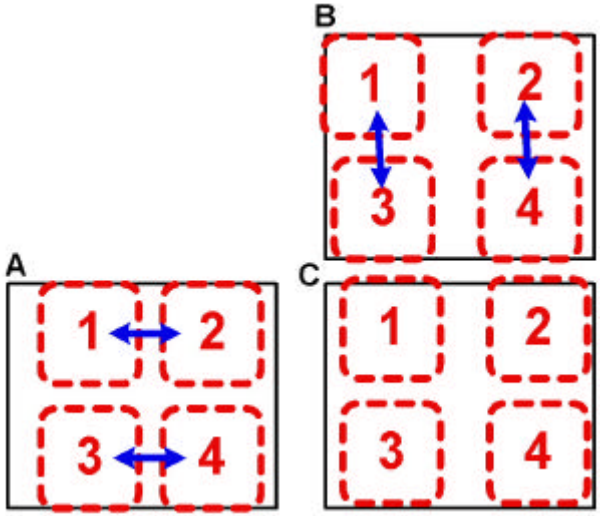
- Partitioning

p blocks of n^2/p elements

- Submatrix $C_{i,j}$:

$$C_{i,j} = \sum_{k=1}^{\sqrt{p}} A_{i,rowk} . B_{colum\,k,j}$$

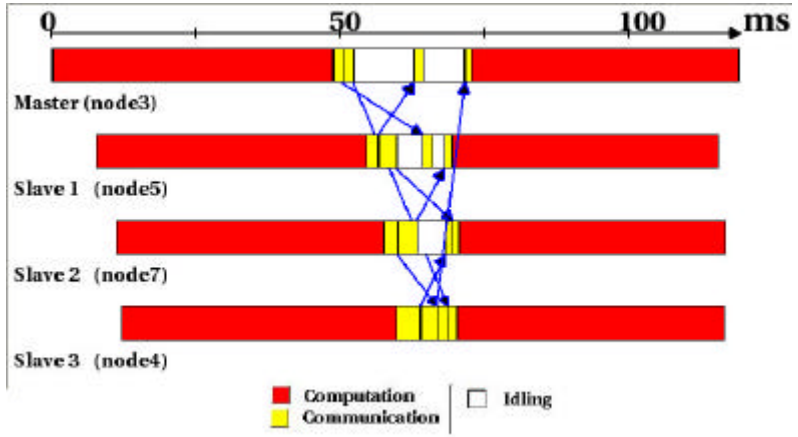
- Communication



Phases

Execution profile

matrix size = 150x150



Analysis based on the '***phases'*** of the parallel application

- UP: *group phases*
  > patterns of inefficiency (kappaPi, Kojak, …)
- DOWN: *refine phases*
  > explain phase execution times

Lost Cycles

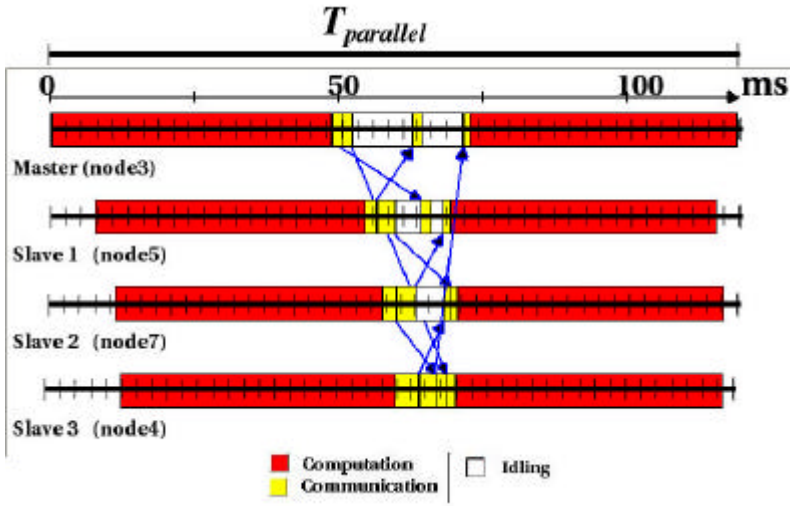
If *Speedup* is the main performance metric

$$Speedup = \frac{T_{seq}}{T_{par}}$$

- Each processor receives T_{par} time

$$T_{par} = T_{par}^{1} = T_{par}^{2} = \ldots = T_{par}^{p}$$

- ***Where are the lost cycles ?*** [Crovella '94]



Overhead Ratio

- Impact of a phase on the speedup: ***Overhead Ratio***

$$Ovh^{i}_{phase} = \frac{T^{i}_{phase}}{T_{seq} / p}.$$

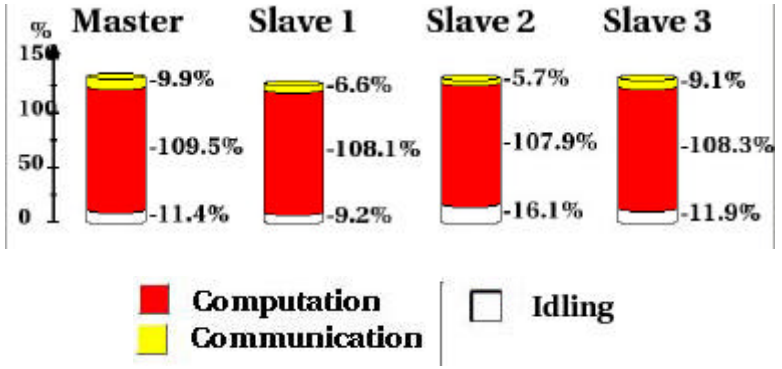
for processor i

$$Ovh_{phase} = \frac{\sum_{i}^{p} T^{i}_{phase}}{T_{seq}} = \frac{T_{phase}}{T_{seq}}.$$

totalized

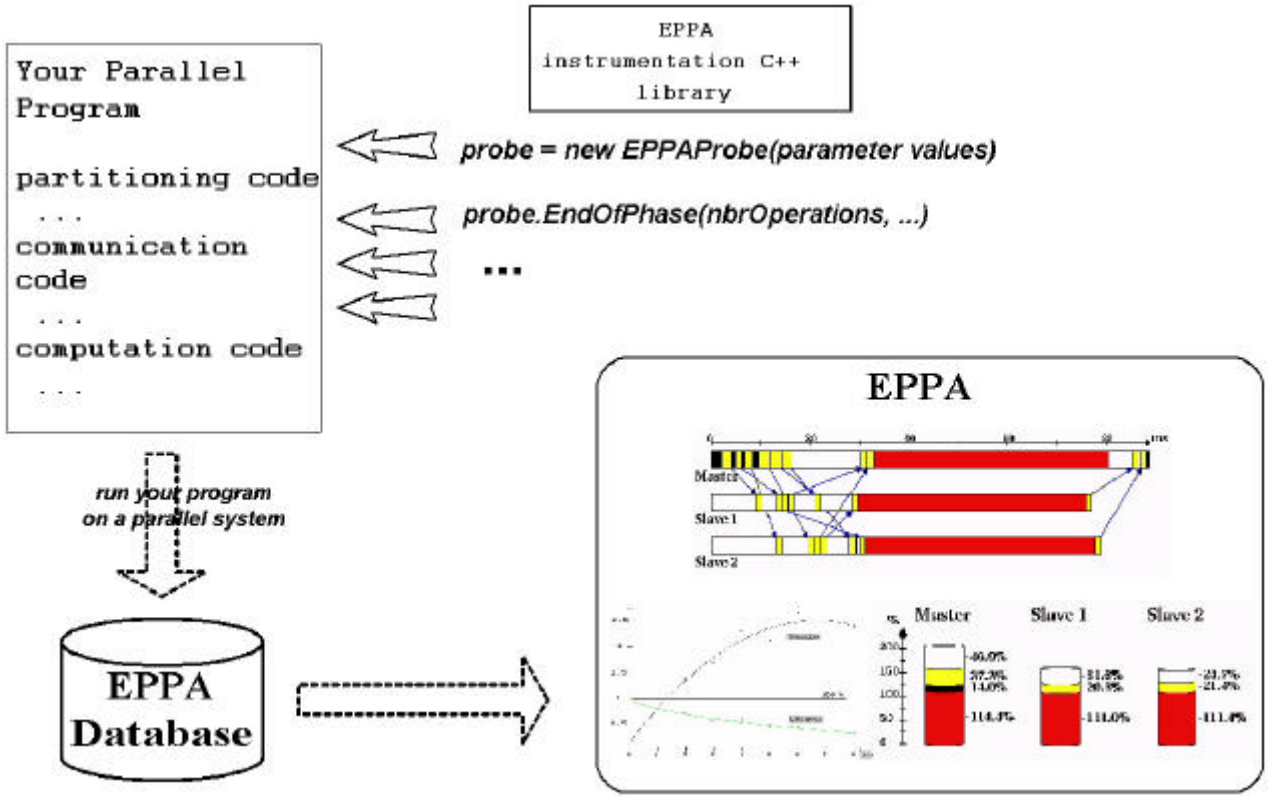
- Speedup becomes:

$$S = \frac{p}{\sum_{j}^{phases} Ovh_j}.$$



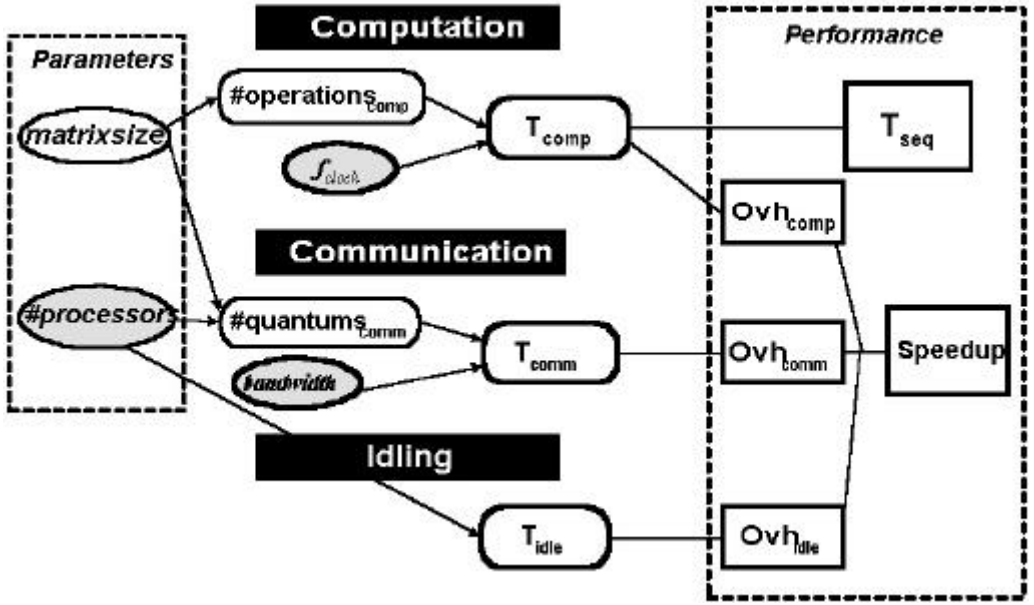
EPPA tool

Experimental Parallel Performance Analysis



Performance Model

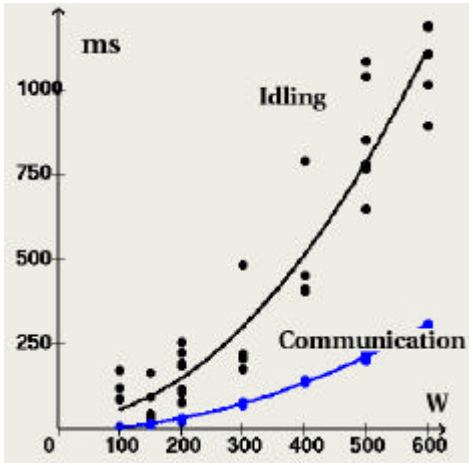
Causal model



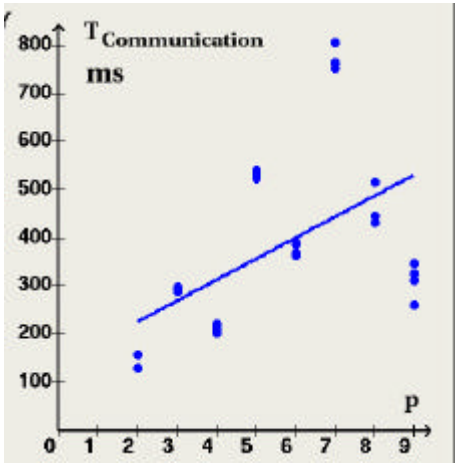
To structure the variables

Communication overhead

$T_{communication}$ In function of matrix size

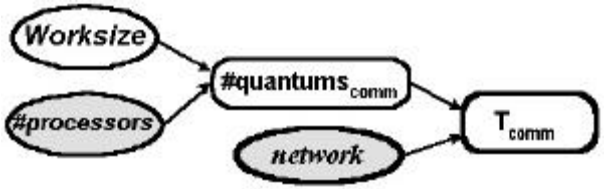


$T_{communication}$ In function of #processors



$$\#quantums_{comm} \sim n^2.\max(r,c)$$

$$\max(r,c) = \sqrt{p} \quad if\ quadratic\ values\ of\ p$$



Benefit of application info & intermediate variables

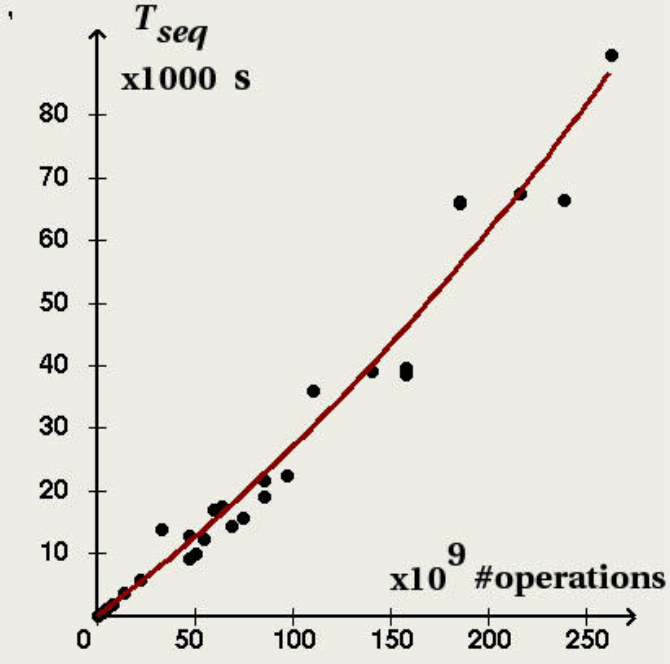
First-Order Approximation

$$T_{computation} = d_{comp}.\#operations = \frac{\#cycles_{op}.\#operations}{f_{clock}}$$

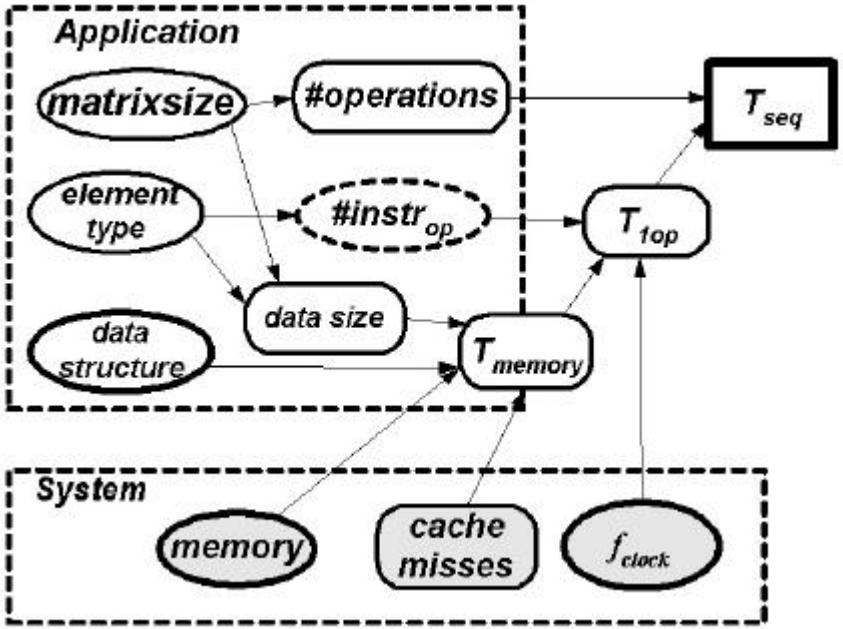
- **Utility**
  - in high-level terms
  - simple & understandable
  - for estimation
  - to start with, refine when necessary

Experiments

Sequential matrix multiplication



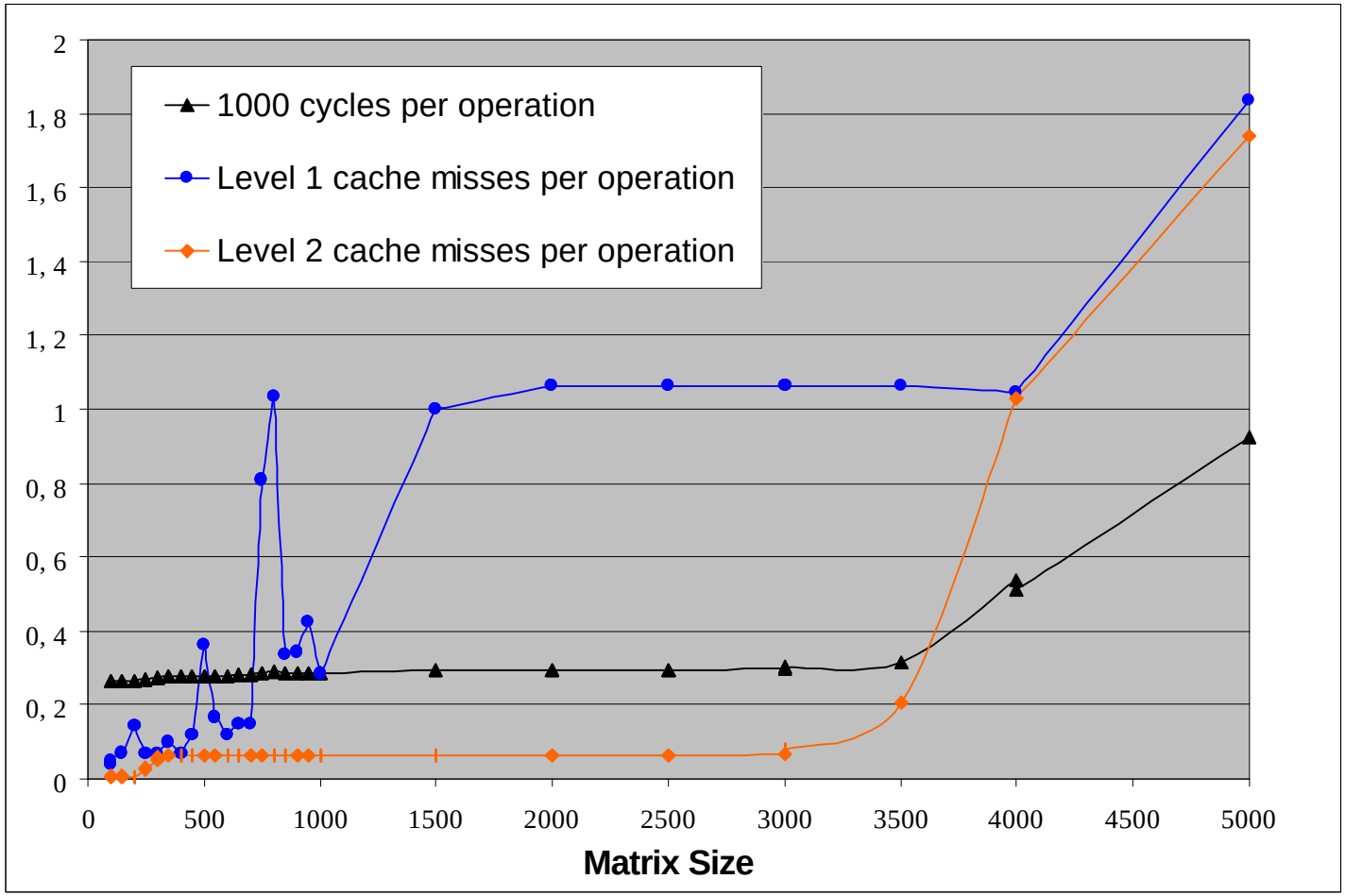
Refinement



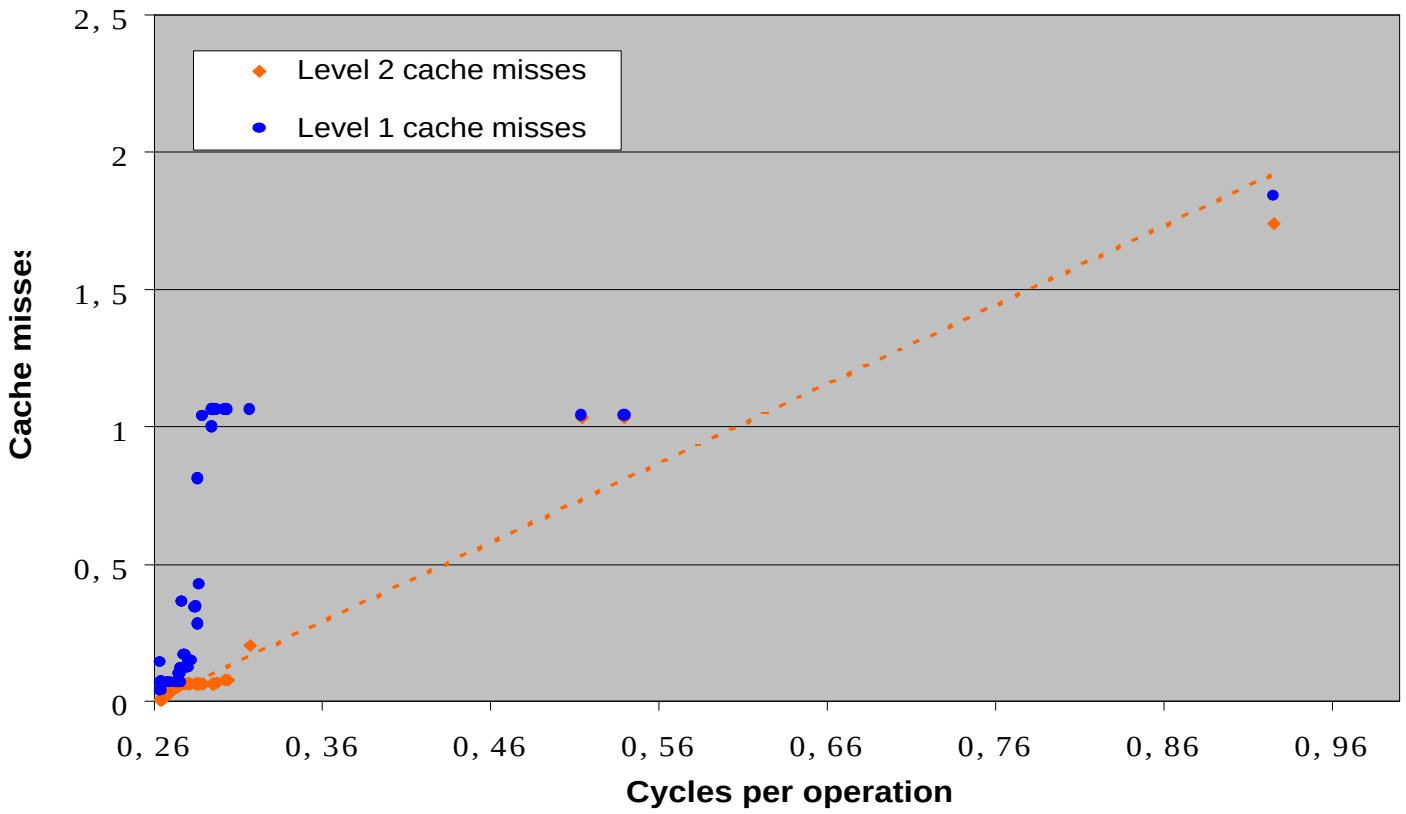
First-order model

Higher-order model

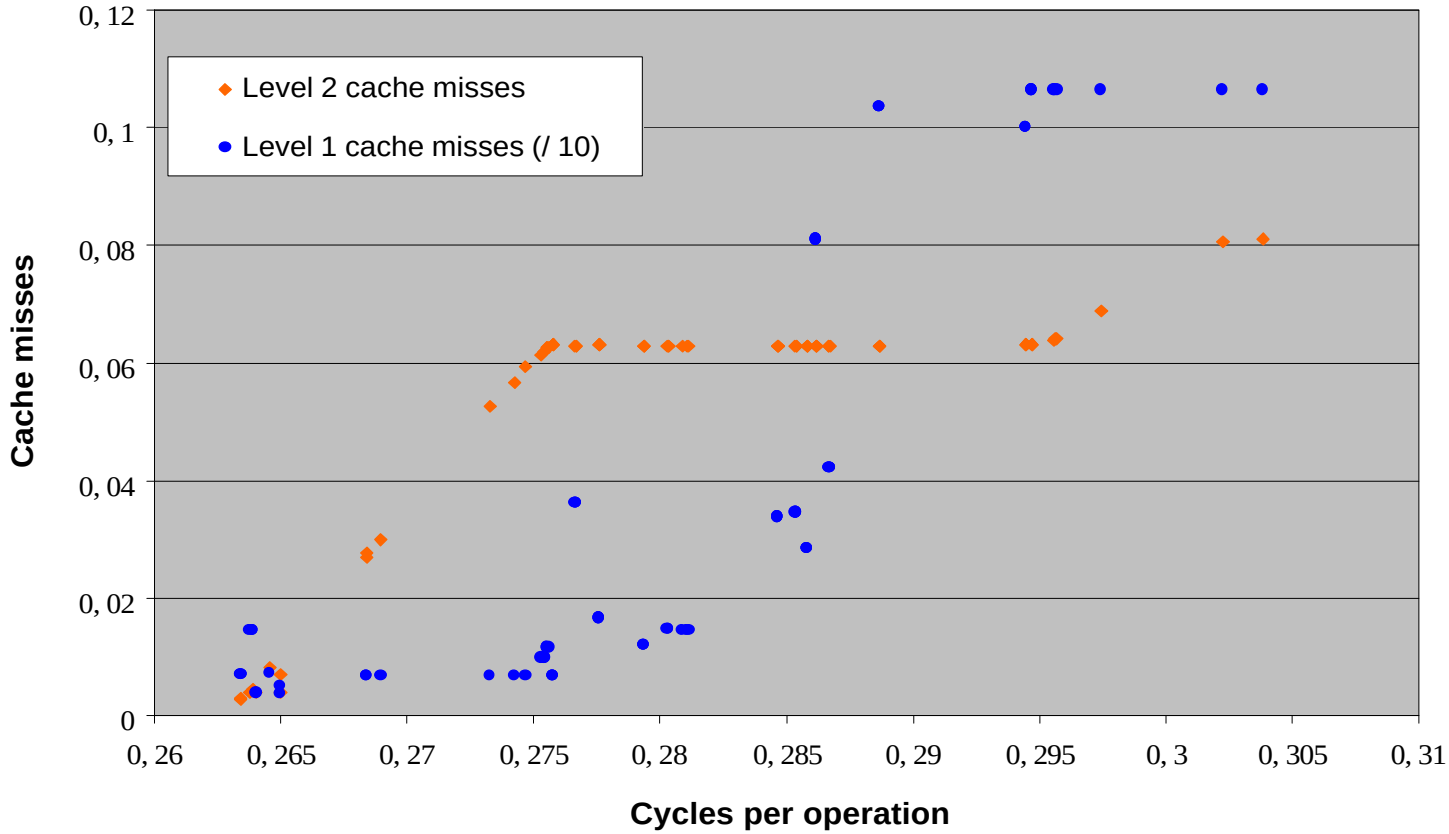
Low-level information



$Cycles_{1op}$~ level 2 cache misses



What is cause?



=> ***statistical analysis to find causal relations***

Parallel Systems lab

Causal Models?

- Performance Analysis is about causes:
  - Why good/bad performance?
  - Reason of long phase times (eg. idling)?
  - Performance dependency of environment variables
- Flexibility
  - Refine model with extra information
  - When information is incomplete, estimate performance with the statistical expectancy
  - Reuse submodels

Parallel Systems lab

Conclusions

- Methodology is non-trivial
- Refinement strategy to tackle complexity
- Causal Models?