

# Performance modeling at Ghent University

---

Lieven Eeckhout  
Ghent University, Belgium



# Research topics

---

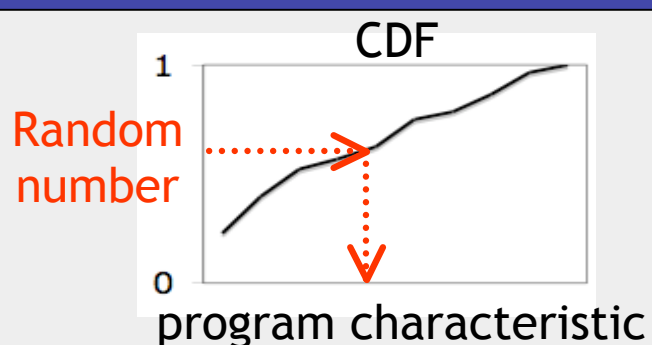
- Statistical simulation
  - Superscalar processor
  - Multi-core processor
- Benchmark generation
- Analytical superscalar processor modeling
- Workload characterization

# Statistical simulation in a nutshell

## Profiler

Functional simulation &  
simulation of locality events

## Generator



## Statistical simulator

simulation of  
synthetic trace

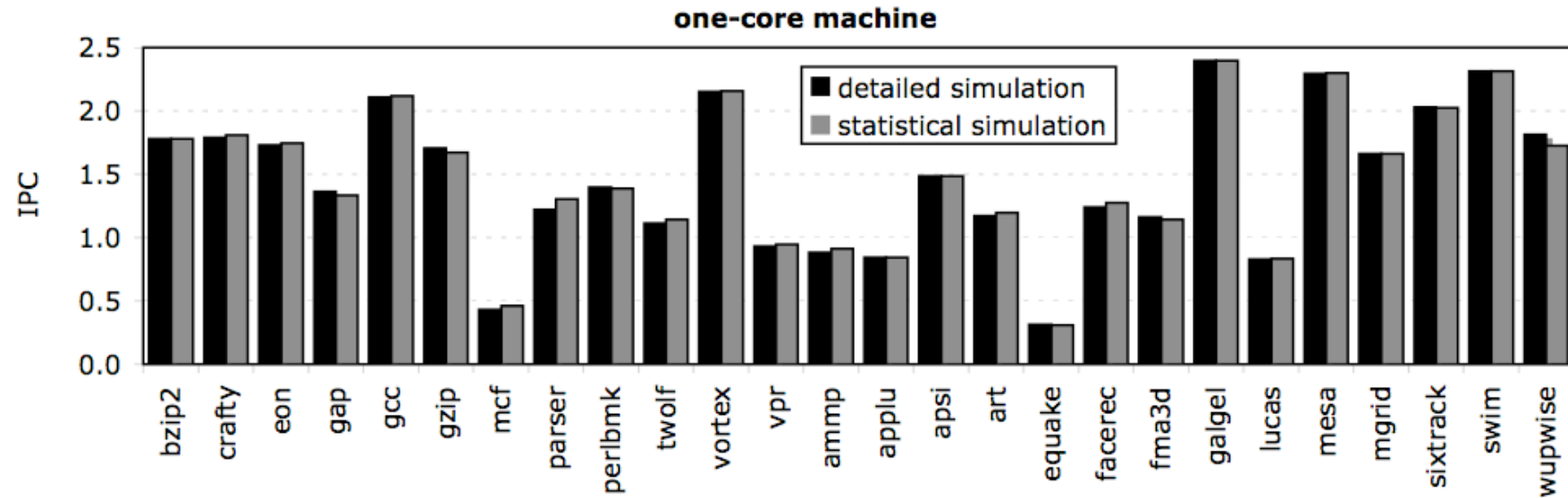
- Collect distribution of program characteristics
  - instruction mix
  - statistical control flow graph
  - inter-instruction dependence distributions
  - miss rates: branch predictor + caches
- Generate a synthetic trace from statistical profile
  - linear sequence of synthetic instructions
  - 1M synthetic instructions is sufficient
- Yields performance metrics such as IPC and EPI
- No simulation of branch predictor and caches needed (info obtained from synthetic instructions)

# Why care about statistical simulation?

---

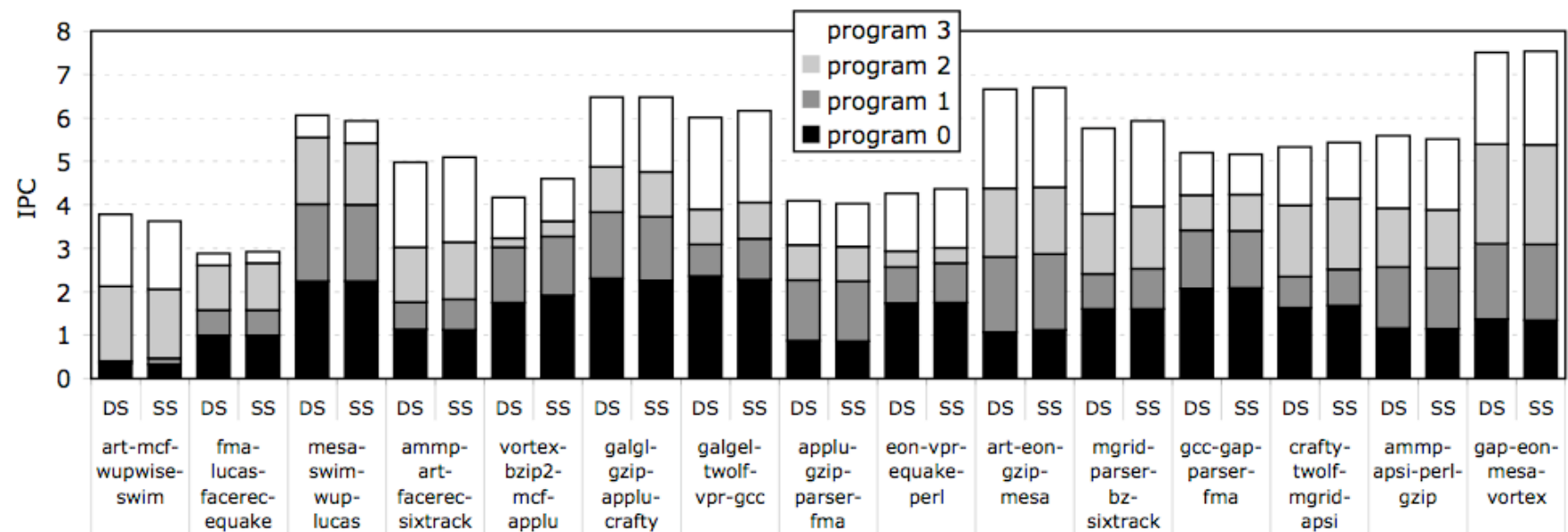
- Extremely fast
  - ~1M synthetic instructions
- Early design space exploration
  - Short development time
  - Short evaluation time
  - Make early design trade-offs

# Uni-core processor



[ISCA'04, ICS'06, DATE'06]

# Multi-core processor w/ L2 shared cache



# Benchmark generation

---

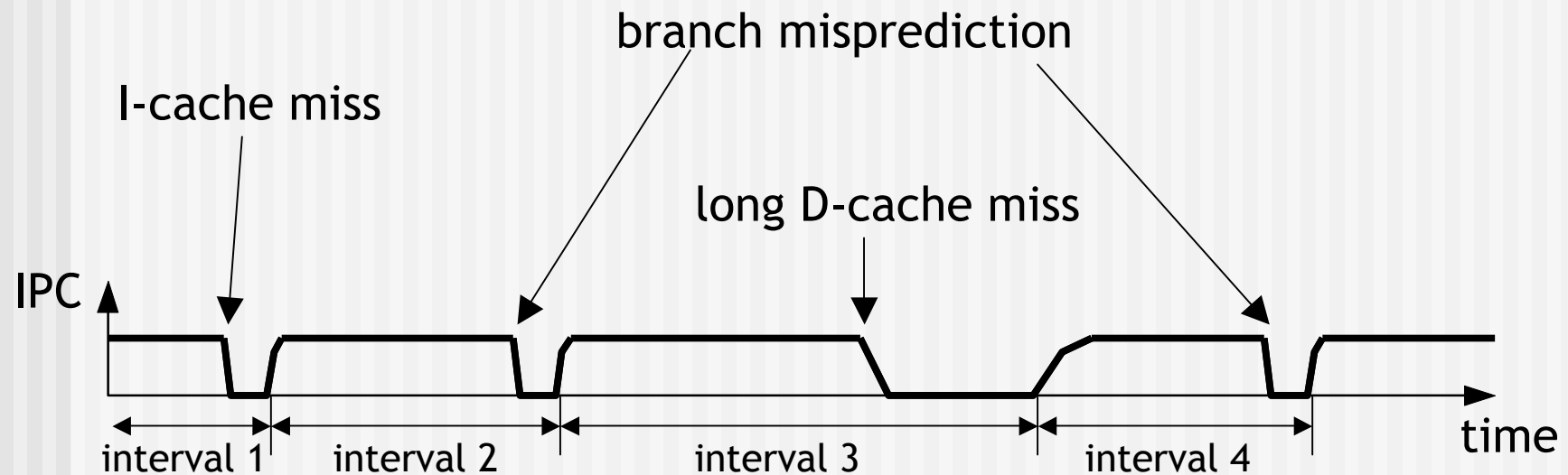
- Hide proprietary application information while preserving performance characteristics
- Enables disseminating proprietary applications as benchmarks
- Two approaches
  - Bottom-up: performance cloning *[ISWC'06, ACM TACO'07, HPCA'08]*
    - Use statistical simulation theory
    - Generate synthetic benchmark instead of a synthetic trace
  - Top-down: code mutation *[ASPLOS'08]*
    - Binary rewrite proprietary application

# Research topics

---

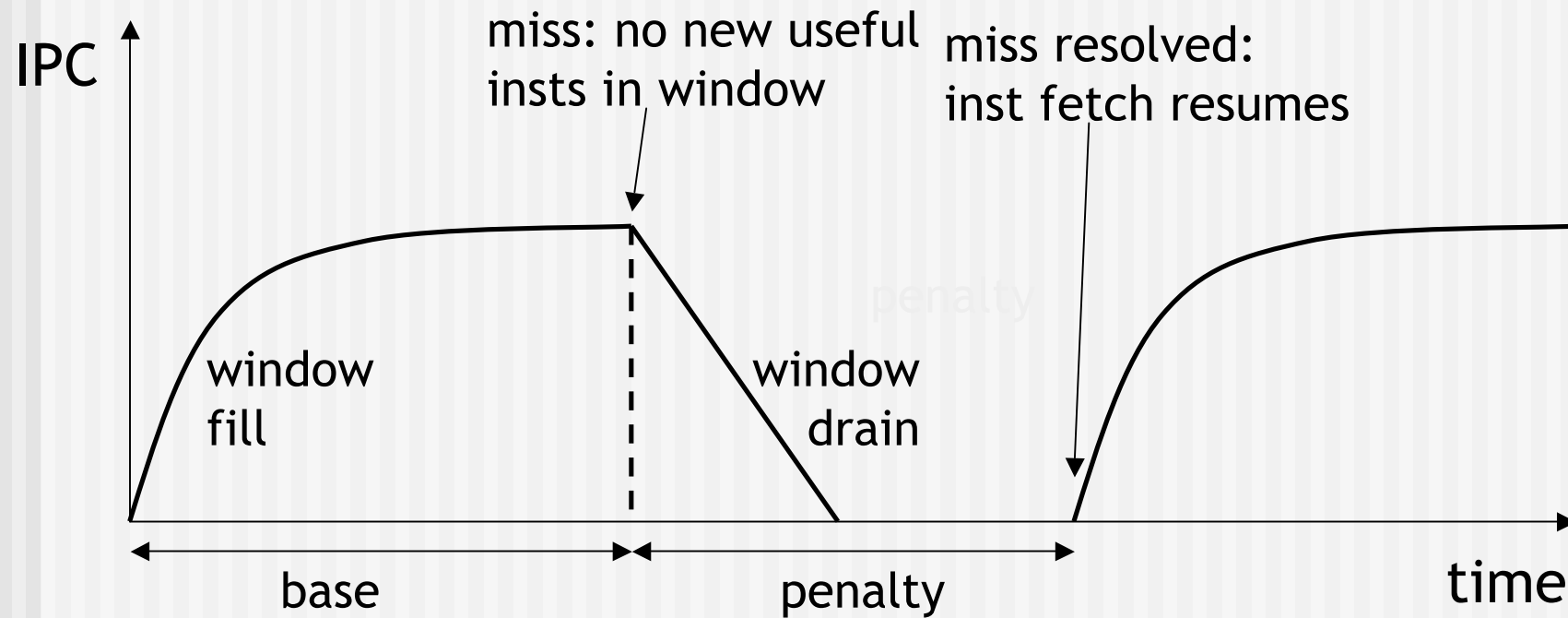
- Statistical simulation
- Benchmark generation
- Analytical modeling
- Workload characterization
  - Benchmark similarity
  - Benchmark suite composition
  - Performance prediction

# Miss events split program execution into isolated intervals



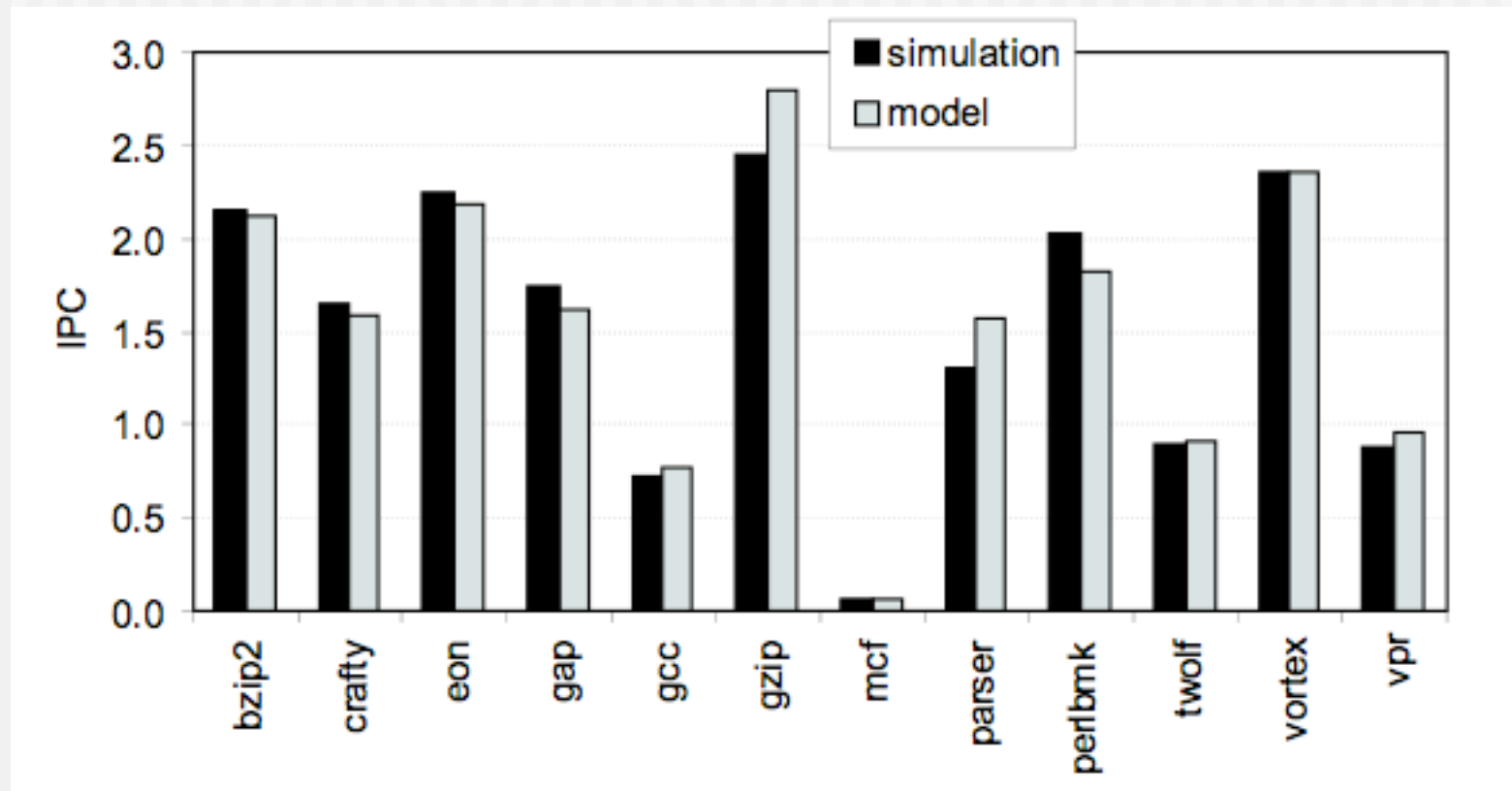
Interval: time between the resolution of the previous miss and the resolution of the current miss

# Interval analysis: penalty is miss event dependent

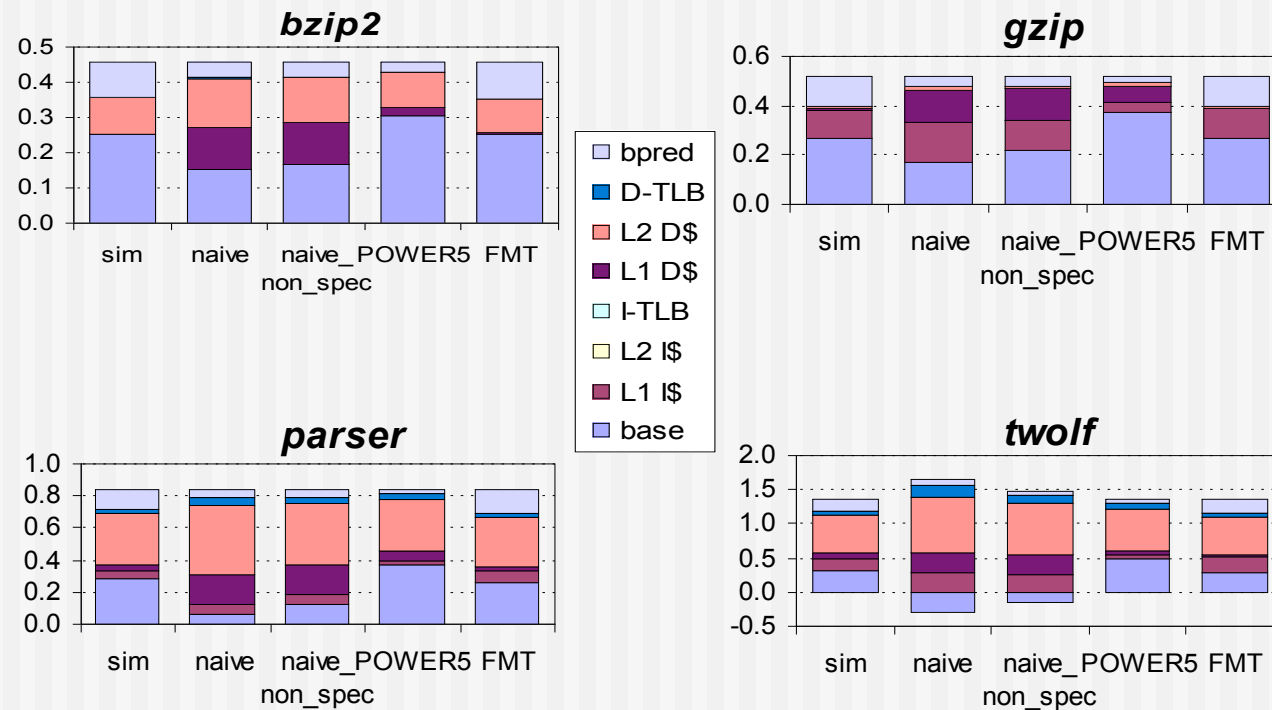


# Mechanistic processor model

*based on first principles*  
*takes as input: miss event rates, instruction mix, ILP*

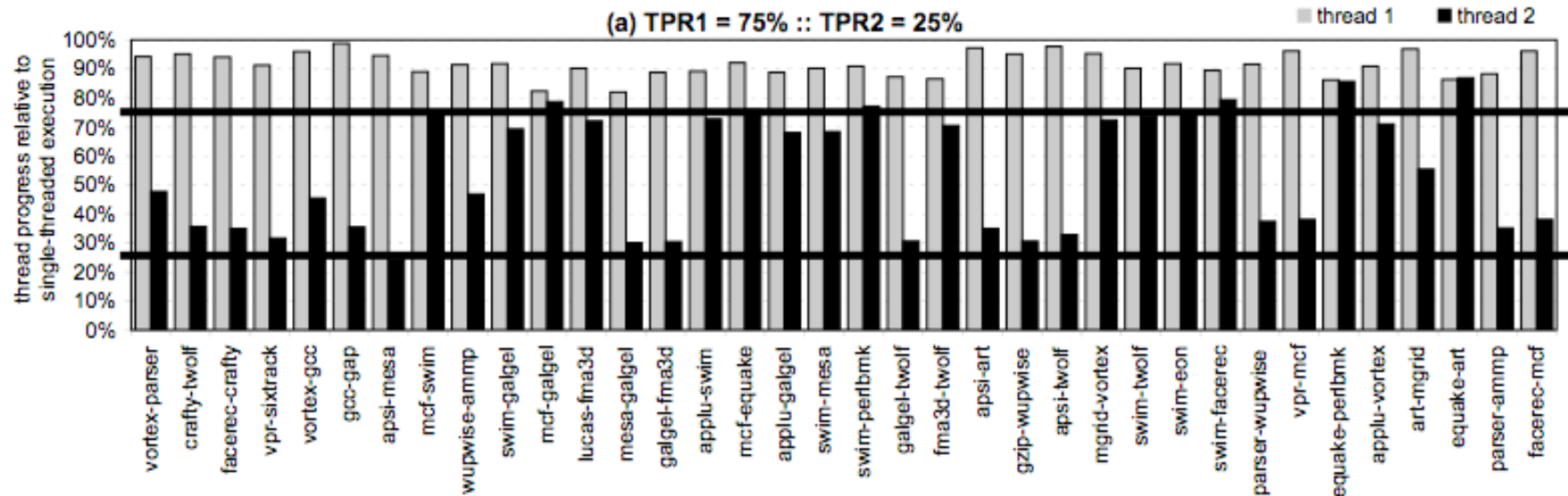


# Counter architecture for computing CPI components



# Towards performance isolation in SMT processors

- Per-thread cycle accounting in SMT processor
  - Track per-thread progress
  - Impose per-thread progress



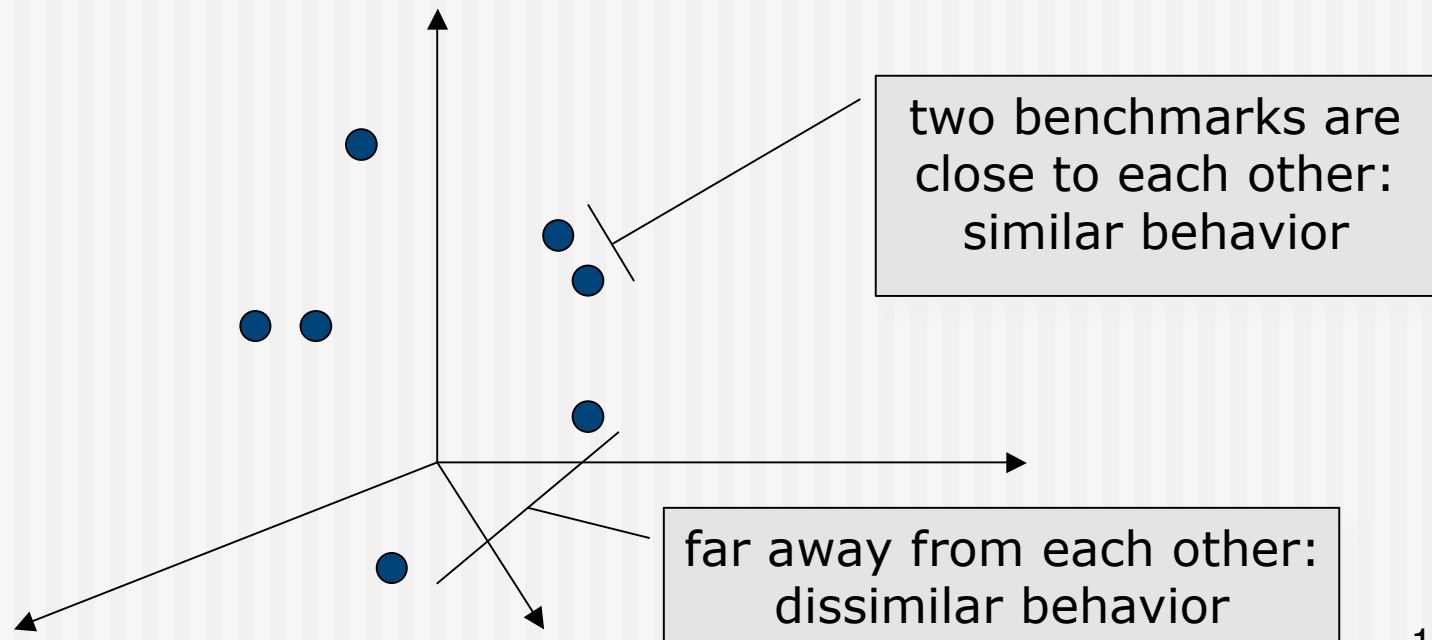
# Research topics

---

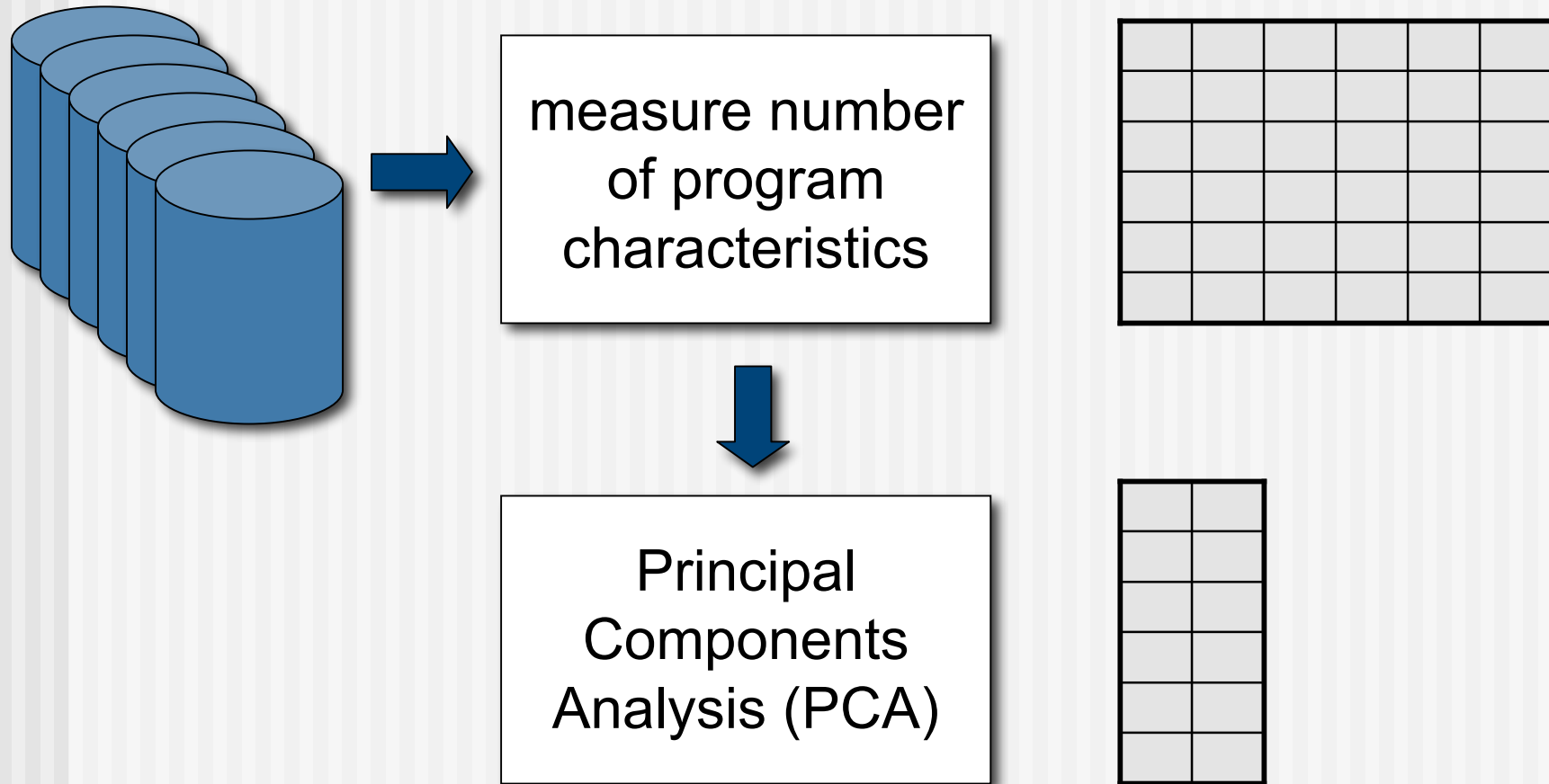
- Statistical simulation
- Benchmark generation
- Analytical modeling
- Workload characterization
  - Benchmark similarity
  - Benchmark suite composition
  - Performance prediction

# Measuring similarity

- Workload space is p-dimensional space
  - $p = \#$  program characteristics
  - related to branch behavior, cache behavior, parallelism, instruction mix, etc.

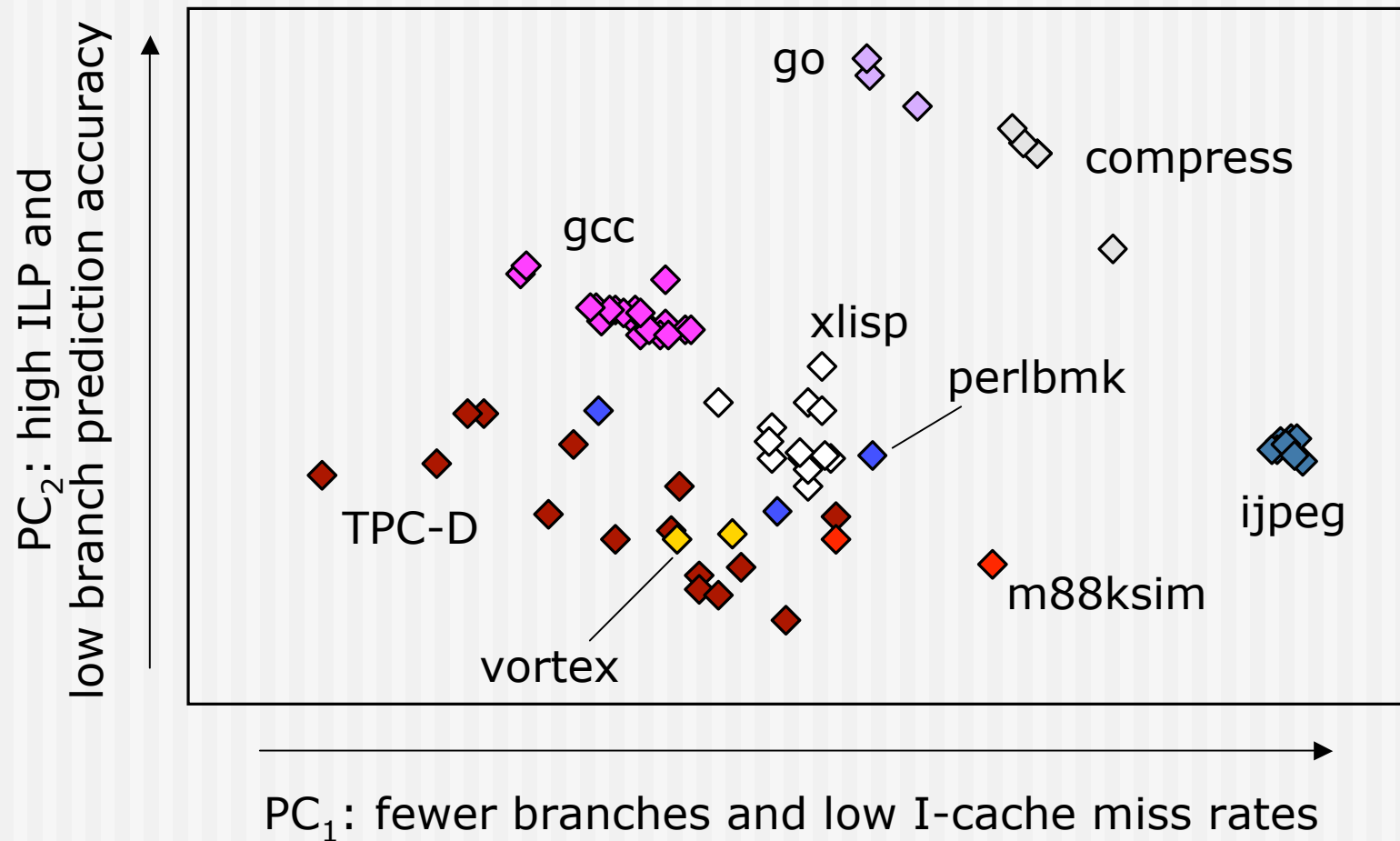


# Methodology



# Workload space after PCA

PC<sub>1</sub> vs. PC<sub>2</sub>



# Performance prediction

