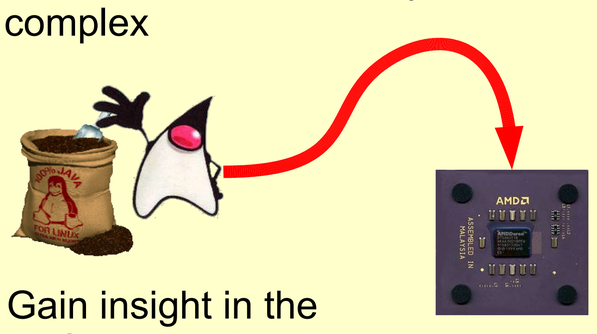
Spotting Java Performance Bottlenecks

A. Georges, D. Buytaert, L. Eeckhout, K. De Bosschere
Dept. ELIS – Ghent University – St.-Pietersnieuwstraat 41, 9000 Gent
{ageorges, dbuytaer, leeckhou, kdb}@elis.ugent.be

Problem

- Java workloads are very complex



- Gain insight in the performance behaviour using hardware performance monitors (HPM)
- Detect bottlenecks in the the execution profile

Annotated execution to detect phases at method level

Execution phases

- A phase is a subtree in the dynamic call-tree of the program

⇒ Phase = root method of the subtree

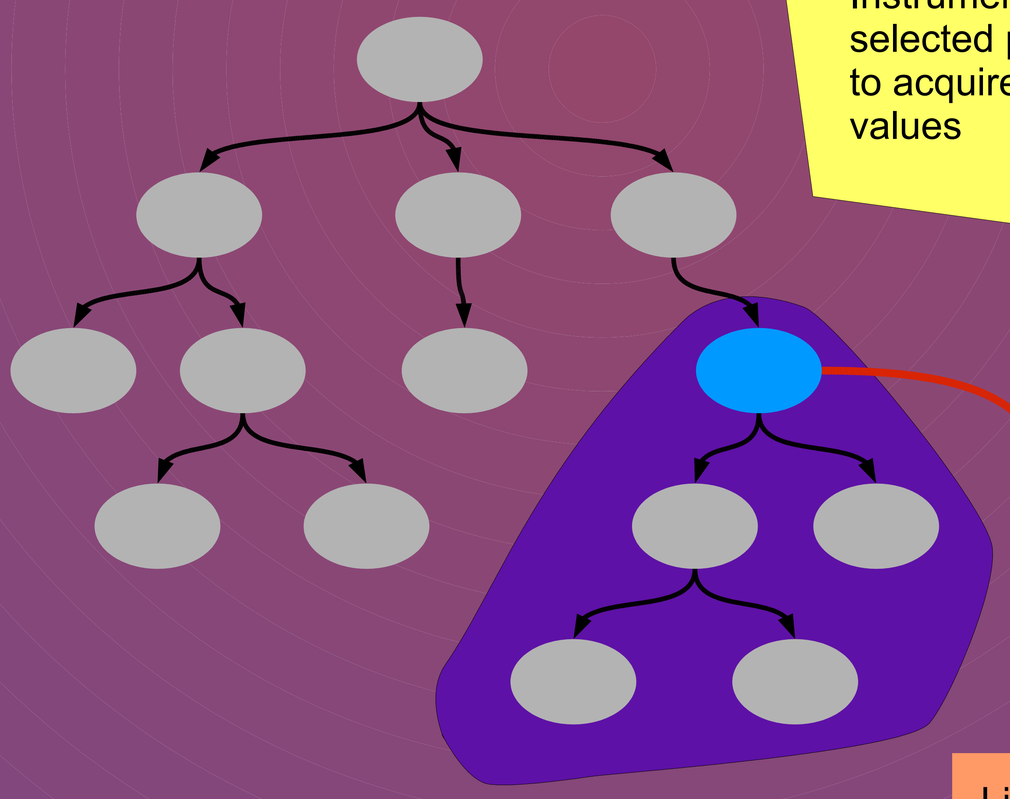
- Phase selection is based on two parameters such that

average execution time of the method > average threshold
total execution time of the method > total threshold

- Phases should exhibit the following properties

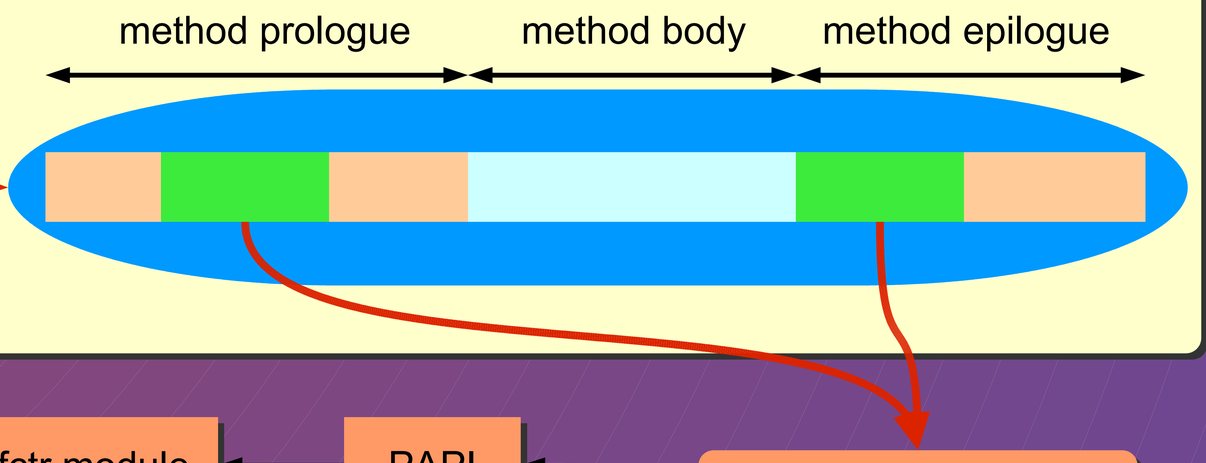
instances of single phase ⇒ little behavioural variation
different phases ⇒ significant behavioural variation

Instrument selected phases to acquire HPM values



Instrumentation

- Jikes RVM compilers instrument methods at compile time by adding a call to the instrumentation code in the prologue and before the epilogue



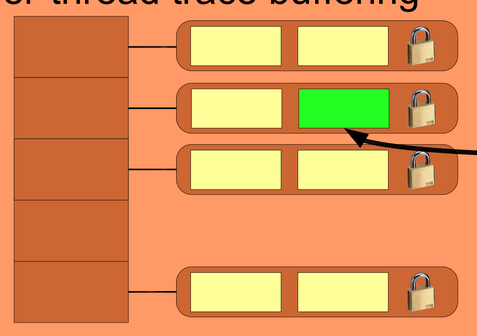
Linux kernel perfctr module

PAPI

Jikes' HPM Interface

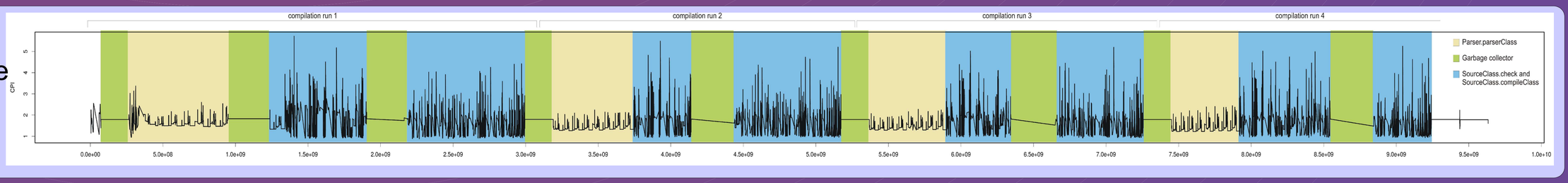
- Stop counting
- Read values
- Store values
- Resume counting

Per-thread trace buffering



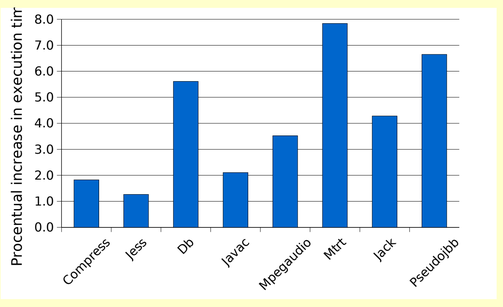
Analyse trace to determine potential bottlenecks

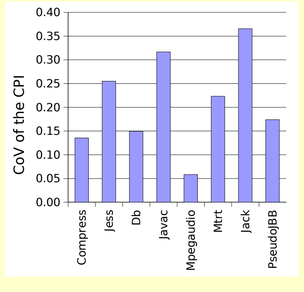
CPI time line for Javac



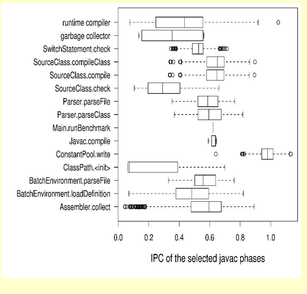
Evaluation

- Benchmarks from SPECjvm98 and SPECjbb2000





Intra- (left), and inter-phase (right) variability for the CPI (cycles per instruction) performance metric



Bottlenecks

Method/phase	Time	CPI	L1 D-cache	L1 I-cache	L2 D-cache	L2 I-cache	Branch misprediction
SourceClass.check	7.06%	2.25	8.06	13.13	1.74	1.75	23.20
SourceClass.compileClass	26.04%	1.74	5.14	6.53	0.98	0.85	16.26
Garbage collector	28.99%	1.81	4.48	0.02	2.55	0.01	4.76
Parser.parseClass	22.68%	1.48	2.93	5.52	0.54	0.43	18.26
Benchmark average		1.67	4.28	4.48	1.32	0.66	13.26

- Potential bottlenecks in Javac (misses are given per 1,000 instructions)

Dries Buytaert is sponsored by a grant from the Flemish Institute for the Promotion of Innovation by Science and Technology (IWT-Flanders), Andy Georges is sponsored by the IWT CoDAMoS project, Lieven Eeckhout is a Postdoctoral Fellow of the Fund for Scientific Research Flanders (F.W.O. Flanders). This research is also funded by Ghent University.