Techniques for Real-System Characterization of Java Virtual Machine Energy and Power Behavior

Gilberto Contreras

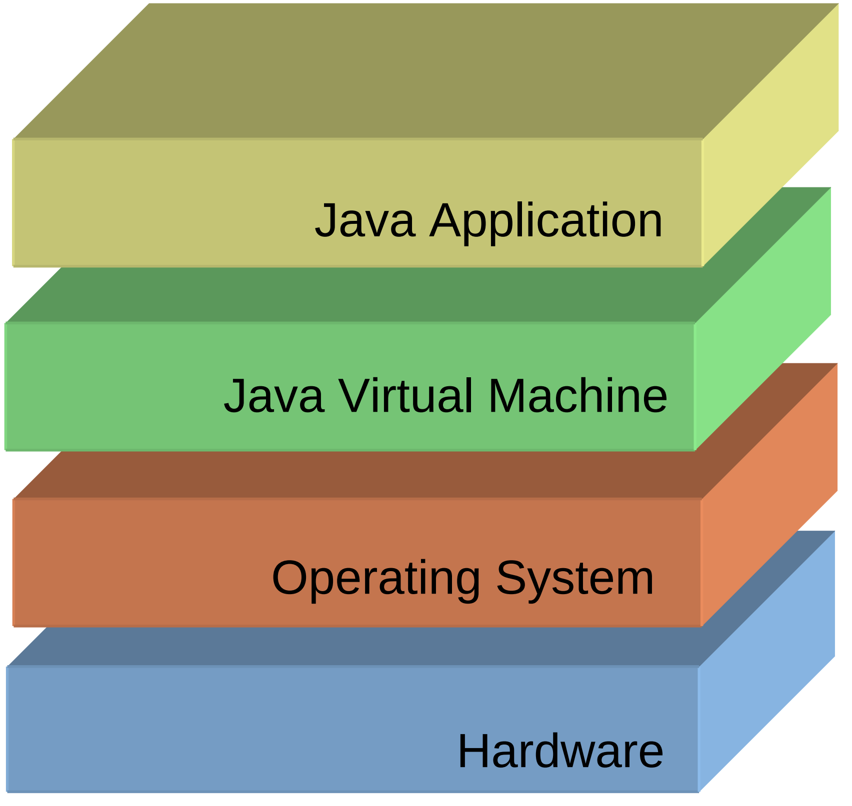
Margaret Martonosi

Department of Electrical Engineering
Princeton University

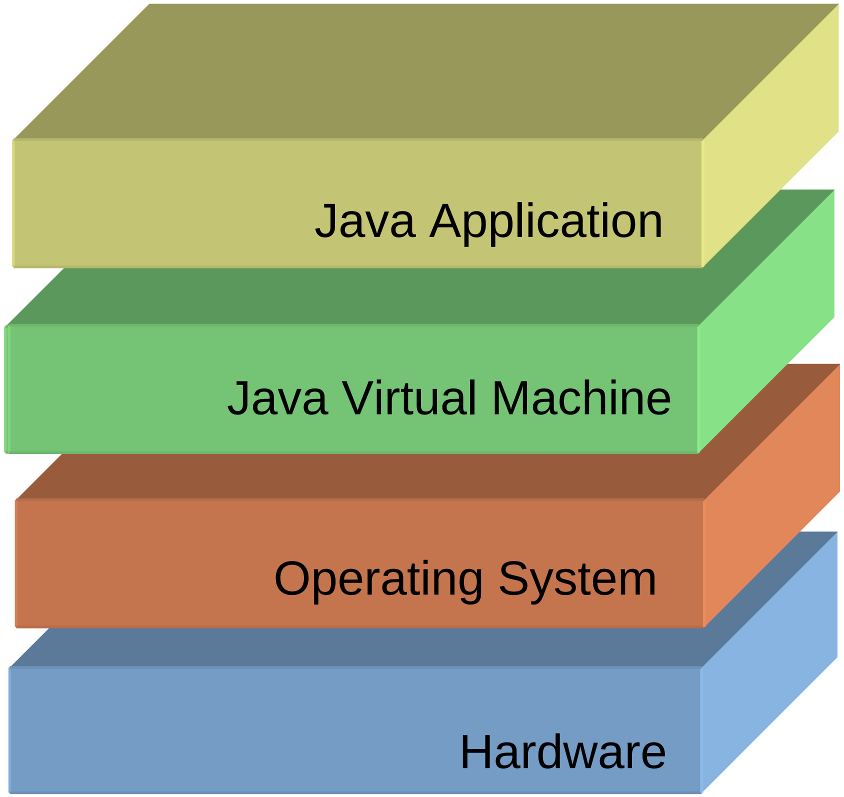
Why Study Power in Java Systems?

- The Java platform has been adopted in a wide variety of devices
- Java servers demand performance, embedded devices require low-power
- Performance is important, power/energy/thermal issues are equally important
- How do we study and characterize these requirements in a multi-layer platform?

Power/Performance Design Issues



Power/Performance Design Issues



- **How do the various software layers affect power/performance characteristics of hardware?**
- **Where should time be invested when designing power and/or thermally aware Java virtual Machines?**

Outline

- Approaches for Energy/Performance Characterization of Java virtual machines
- Methodology
  - Breaking the JVM into sub-components
  - Hardware-based power/performance characterization of JVM sub-components
- Results
  - Jikes & Kaffe on Pentium M
  - Kaffe on Intel XScale
- Conclusions

Power & Performance Analysis of Java

- Simulation Approach

√ Flexible: easy to model non-existent hardware

x Simulators may lack comprehensiveness and accuracy

x Thermal studies require tens of seconds granularity

  - Accurate simulators are too slow

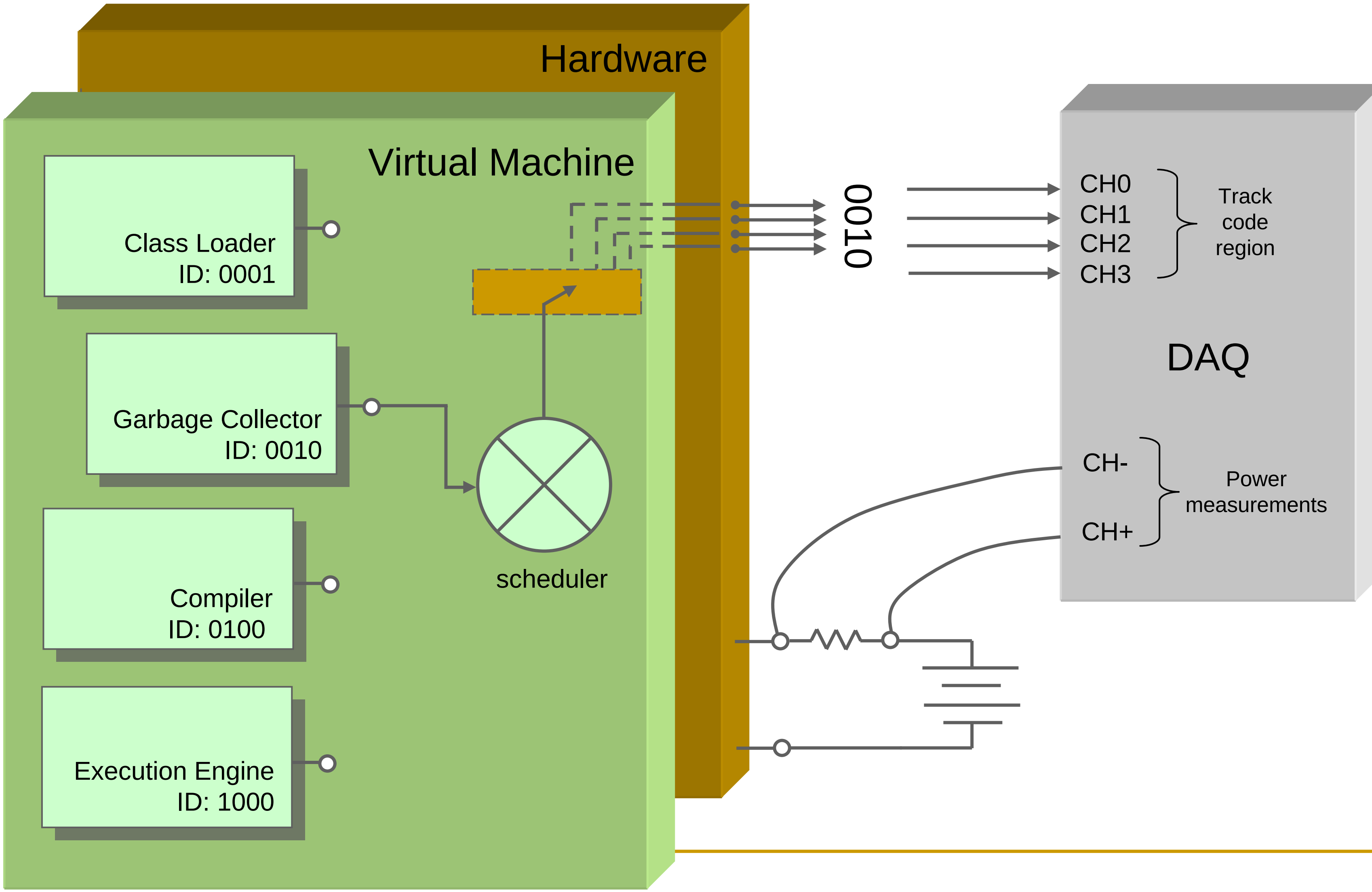
- Hardware Approach

√ Able to capture full-system characteristics and effects

√ Data gathering is comparable to hardware speeds

x Only applicable to existent hardware

Hardware-based Characterization



Two Virtual Machines

	Jikes RVM	Kaffe JVM
Design goal	High performance	Flexibility and portability
Architecture support	High-end processors	High-end to embedded
Garbage collection	Multiple collectors	Mark-and-sweep
Runtime optimizations	Runtime compiler with different optimization levels	Just-in-time compiler

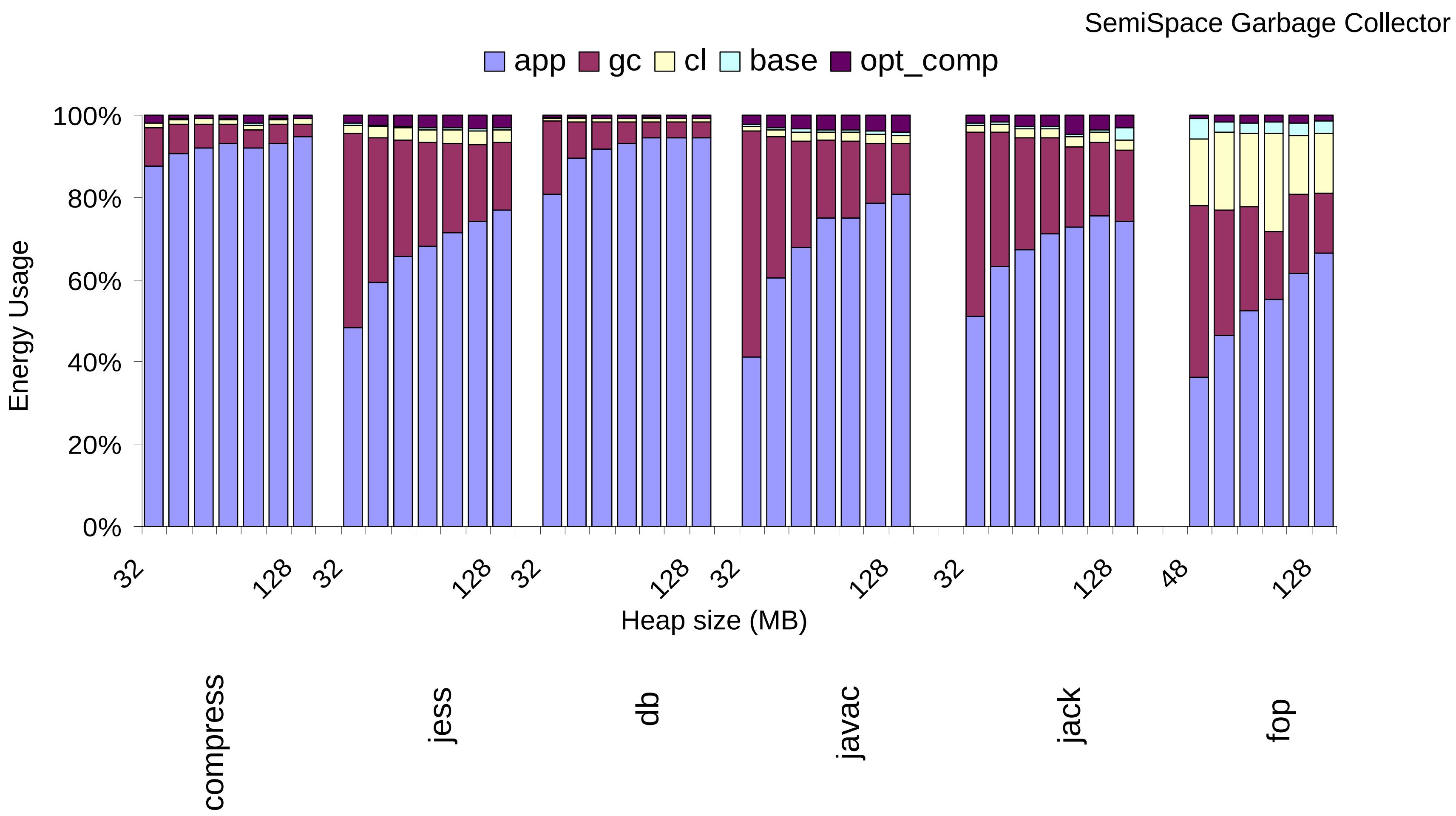
Two Platforms

	Pentium M (P6)	Intel XScale
Platform Type	High-performance mobile computers	High-end handheld devices
Configuration	1.6Ghz, 512MB RAM	400Mhz, 32MB RAM
Theoretical Max Power	31W	1.4W
JVM Used	Jikes RVM Kaffe	Kaffe

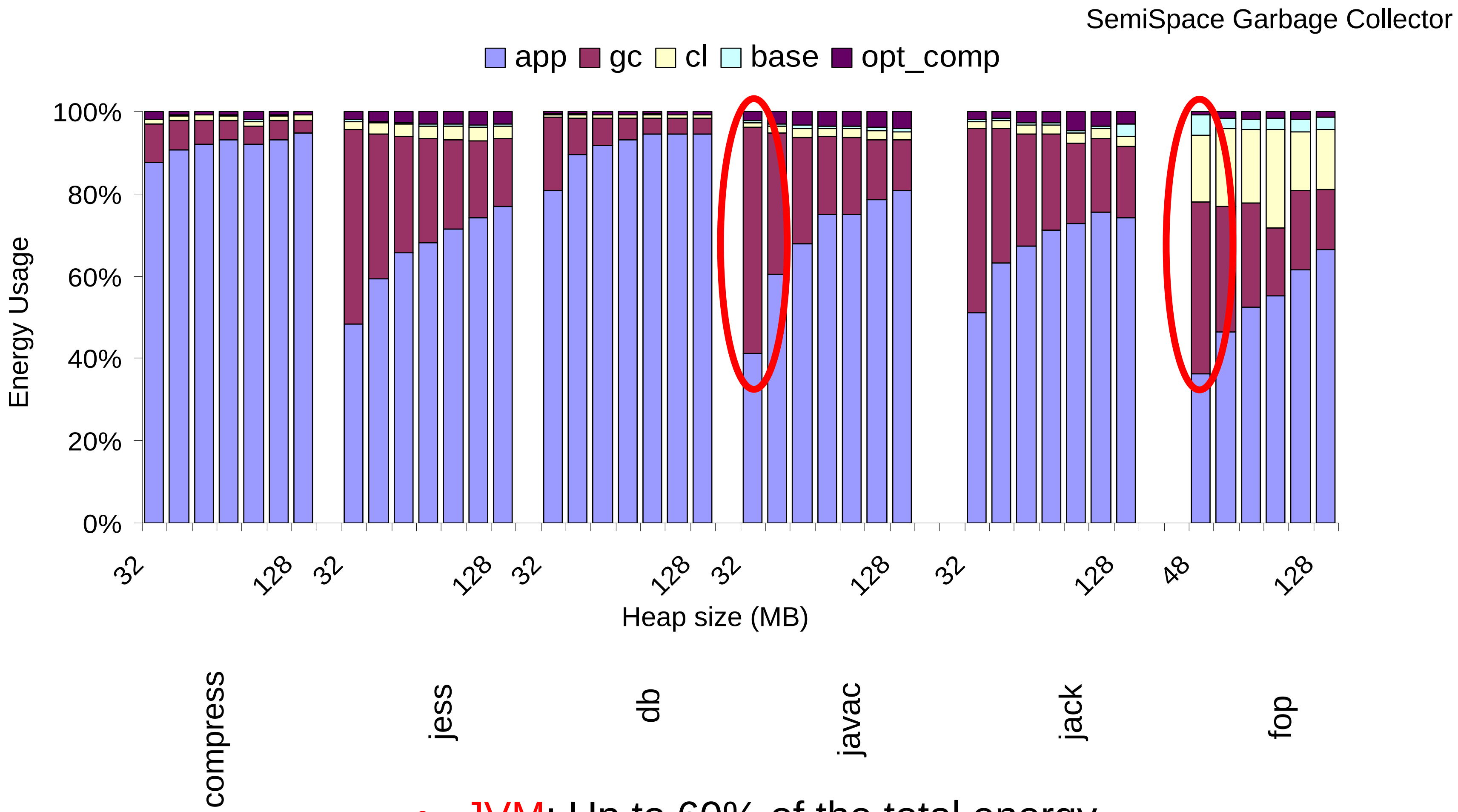
Outline

- Approaches for Energy/Performance Characterization of Java virtual machines
- Methodology
  - Breaking the JVM into sub-components
  - Power/performance hardware-based characterization of JVM sub-components
- Results
  - Jikes & Kaffe on Pentium M
  - Kaffe on Intel XScale
- Conclusions

Jikes Energy Distribution on P6

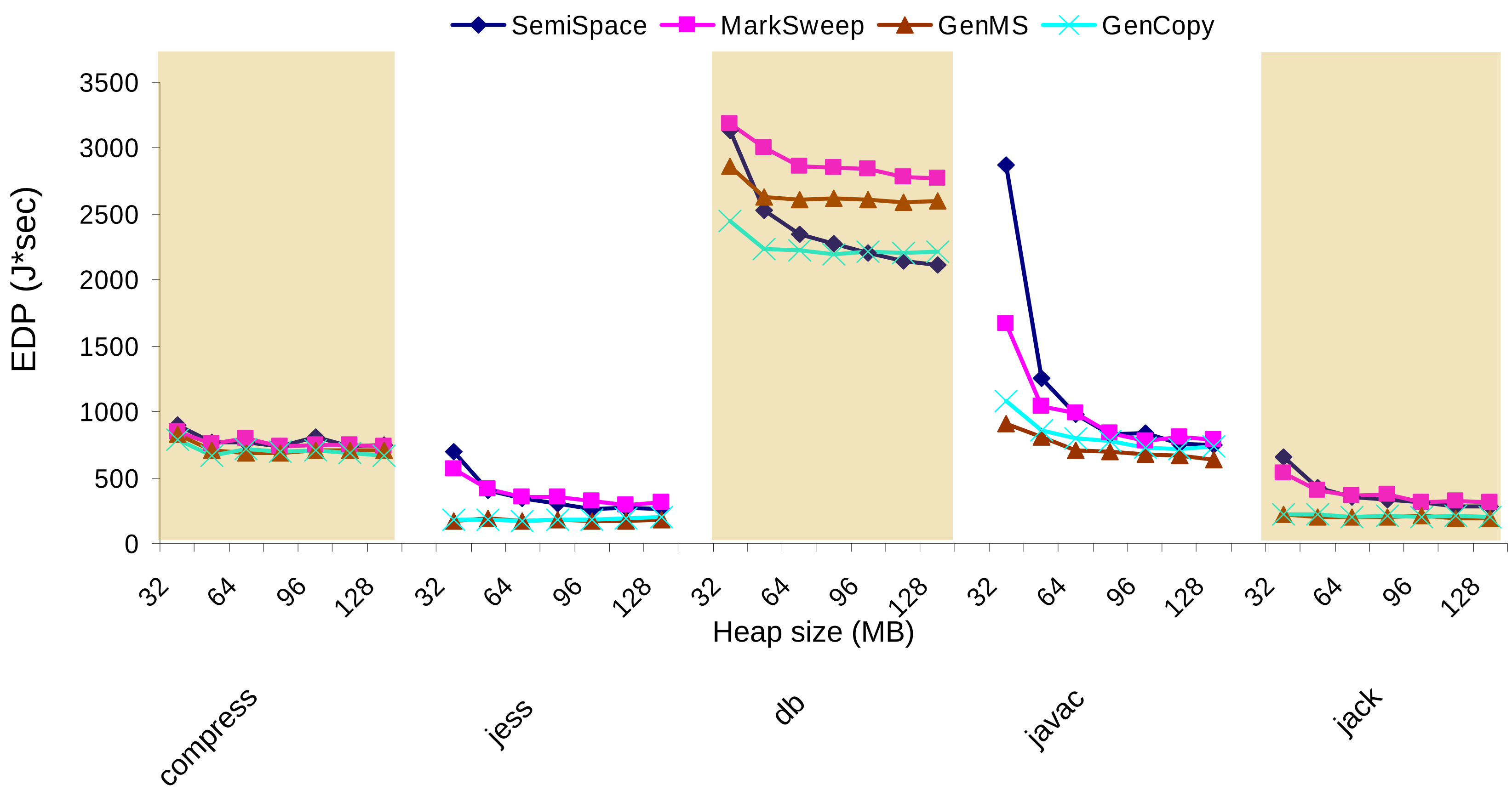


Jikes Energy Distribution on P6



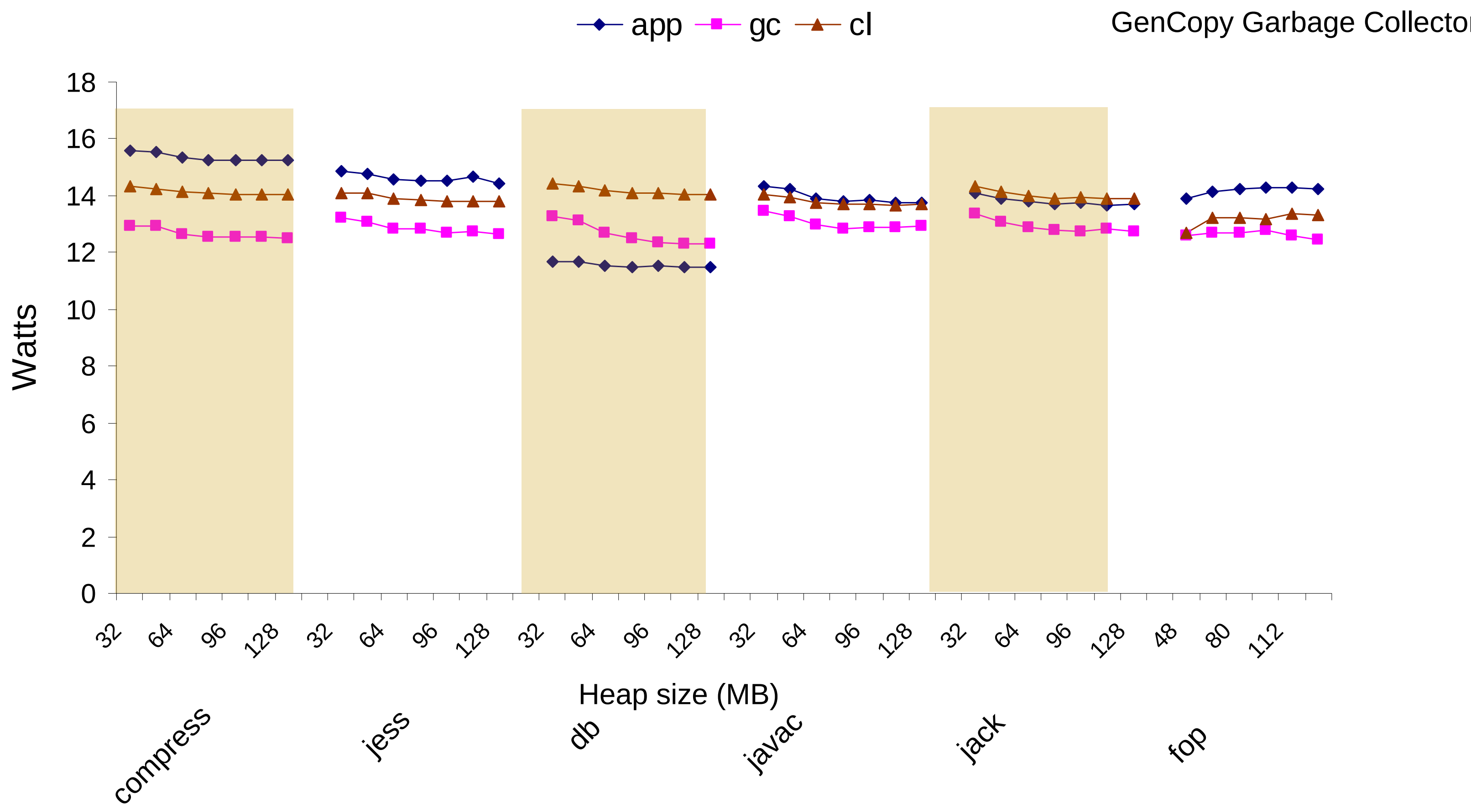
- JVM: Up to 60% of the total energy
- GC: Average 37% of the total energy of SpecJVM98

Jikes Energy-Delay Product on P6



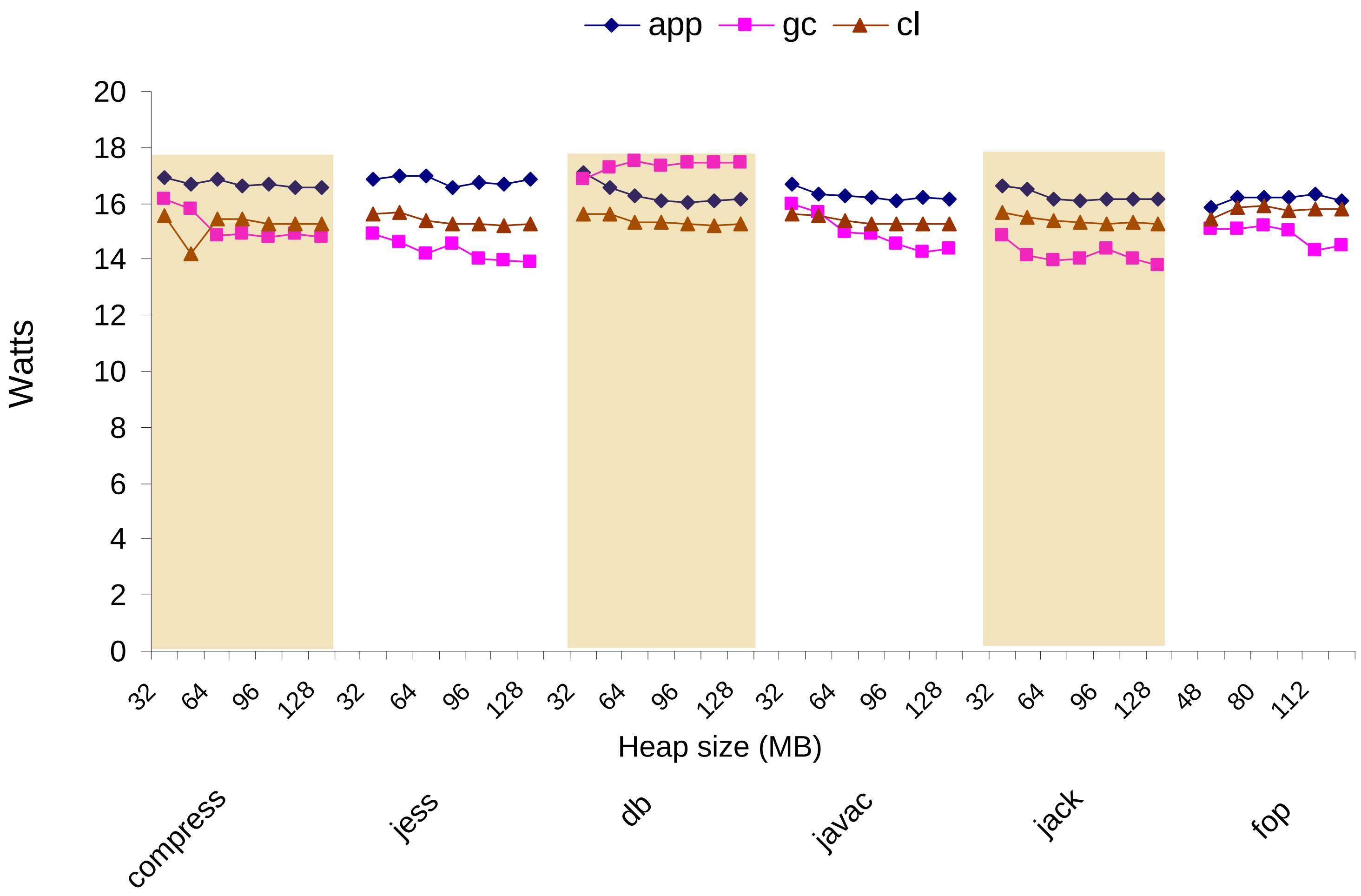
- Jikes: heap size has a significant impact on energy efficiency
- EDP decrease across heap sizes due to a decrease in application execution time

Jikes Power Consumption on P6



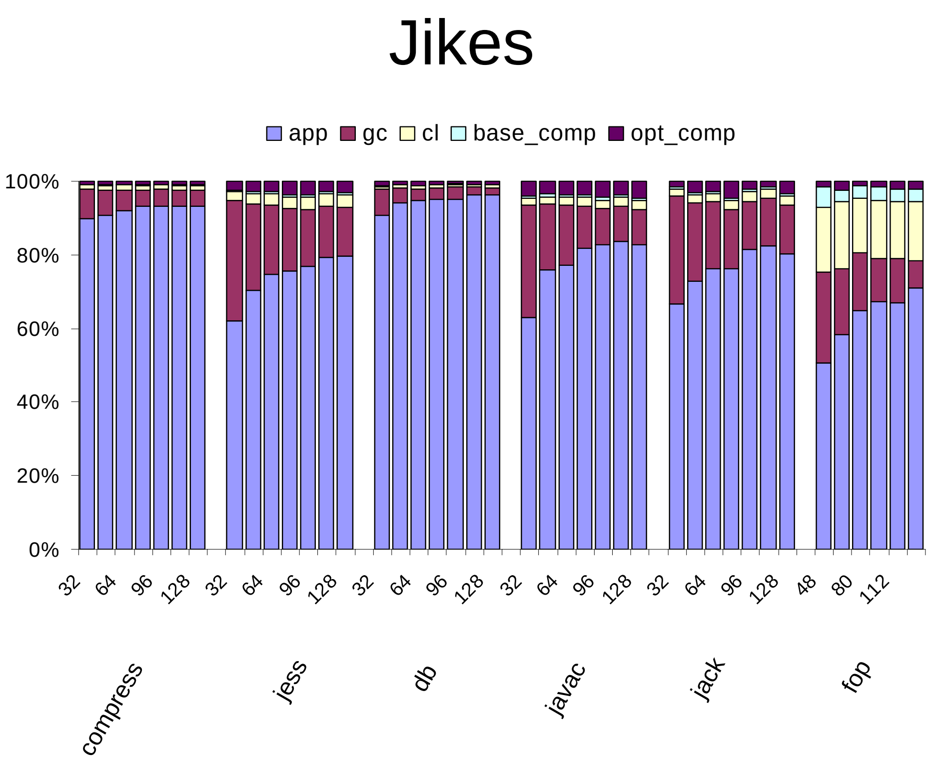
- Average power for JVM varies little across heap-sizes
- Garbage collector is high energy consumer, but low power

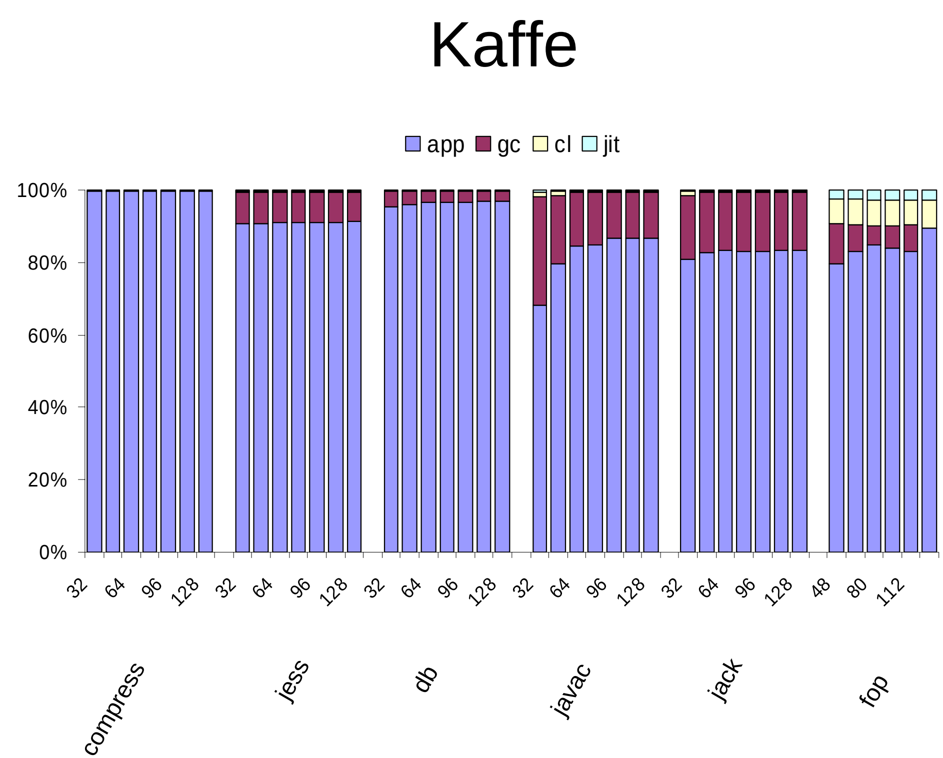
Jikes Peak Power on P6



- Execution engine has the highest peak-power

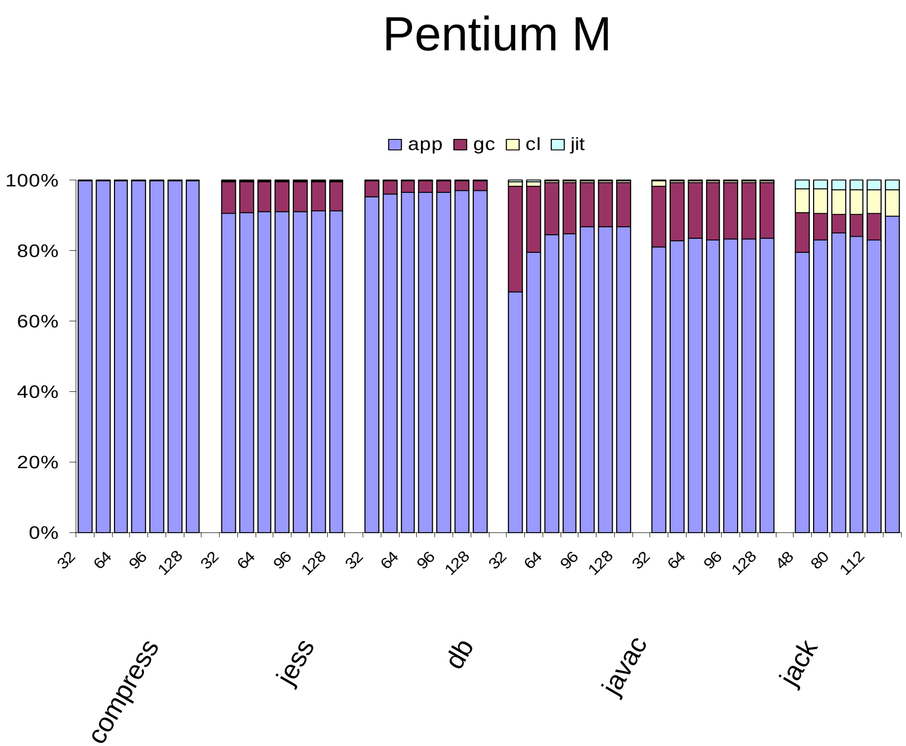
Jikes versus Kaffe: Energy Distribution on P6

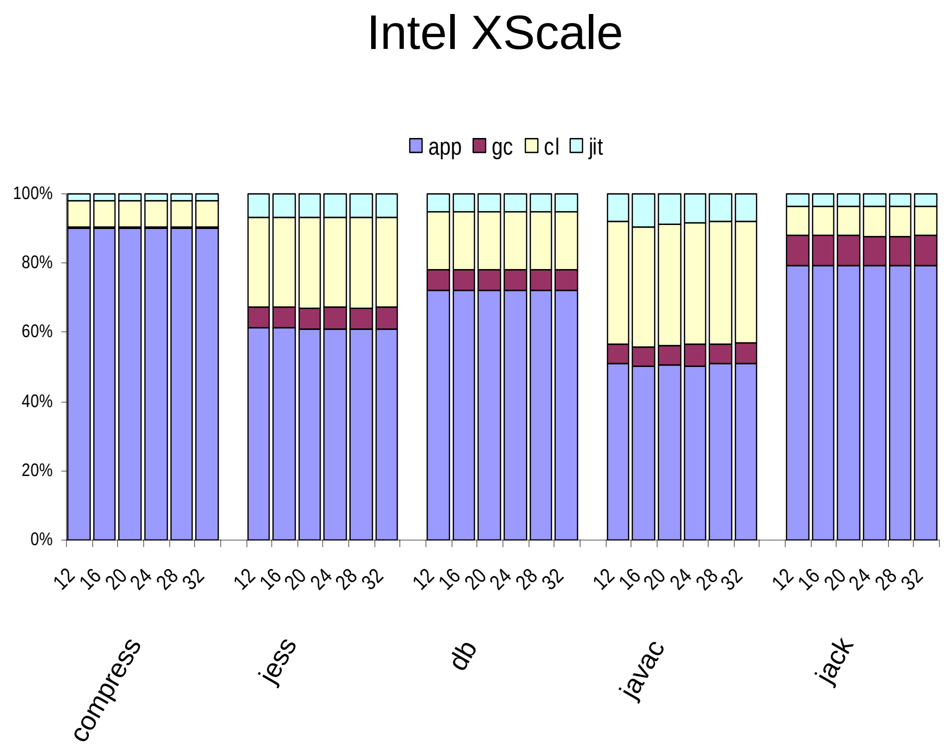




- Kaffe: high application energy caused by long execution times
- Kaffe: 8% of total average energy goes to virtual machine

Kaffe Across Platforms





- XScale: no classes are included in the binary
- XScale: GC only represents 6% of the total energy consumed

Conclusions

- Methodology
  - The complexity of the Java virtual machine calls for a more in-depth power/energy analysis
  - Hardware-based characterization of the virtual machine’s sub-components allow long execution times and trustworthy measurements
- Lessons learned
  - In both platforms, JVM energy overhead is considerable
  - Jikes: the GC is low power but high energy consumer (up to 37% on average)
  - For Kaffe on XScale, the class loaded becomes high-energy consumer (18% for measured benchmarks)

Thank you!