A High Performance Java GC with Thread Private Heaps

Pat_Caudill@Instantiations.com

Allen_Wirfs-Brock@Instantiations.com

www.instantiations.com

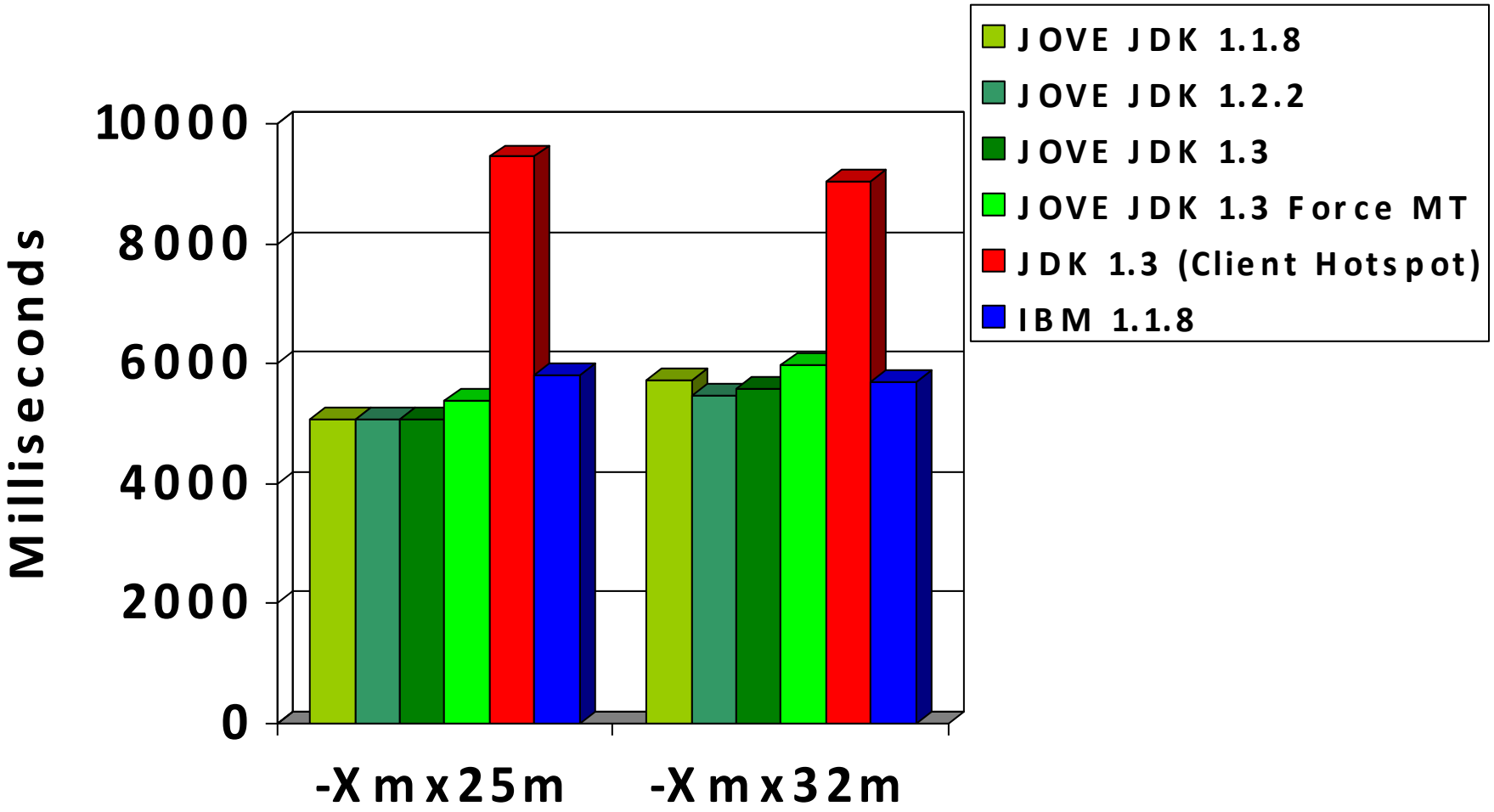
JOVE

- Commercial static native compiler for Java.
  - V1.0 7/99, V1.5 1/00, V2.0 10/00
- Translates class files to Wintel native exe's
- Veryaggressive global optimization
  - but no escape analysis (yet)
- Closed world, whole program assumptions
- JDK-level independent
- Suitable for real-world programs
  - (yes, it can handle SwingSet, Java2D Demo, Java 3D, ImageJ, Jbuilder 4.0, etc.)

JOVE Garbage Collector

- Direct descendent of Tektronix Smalltalk Collector described at OOPSLA'86:
  - Copying, multi-generational collector
  - Generation number/size dynamically adaptable
  - Non-movable large object space
  - Remembered sets track inter-generation references
  - Dual stacks segregate object references from primitive data - no stack parsing
  + Compiler generated, type-specific, object scanners

GCBench Results



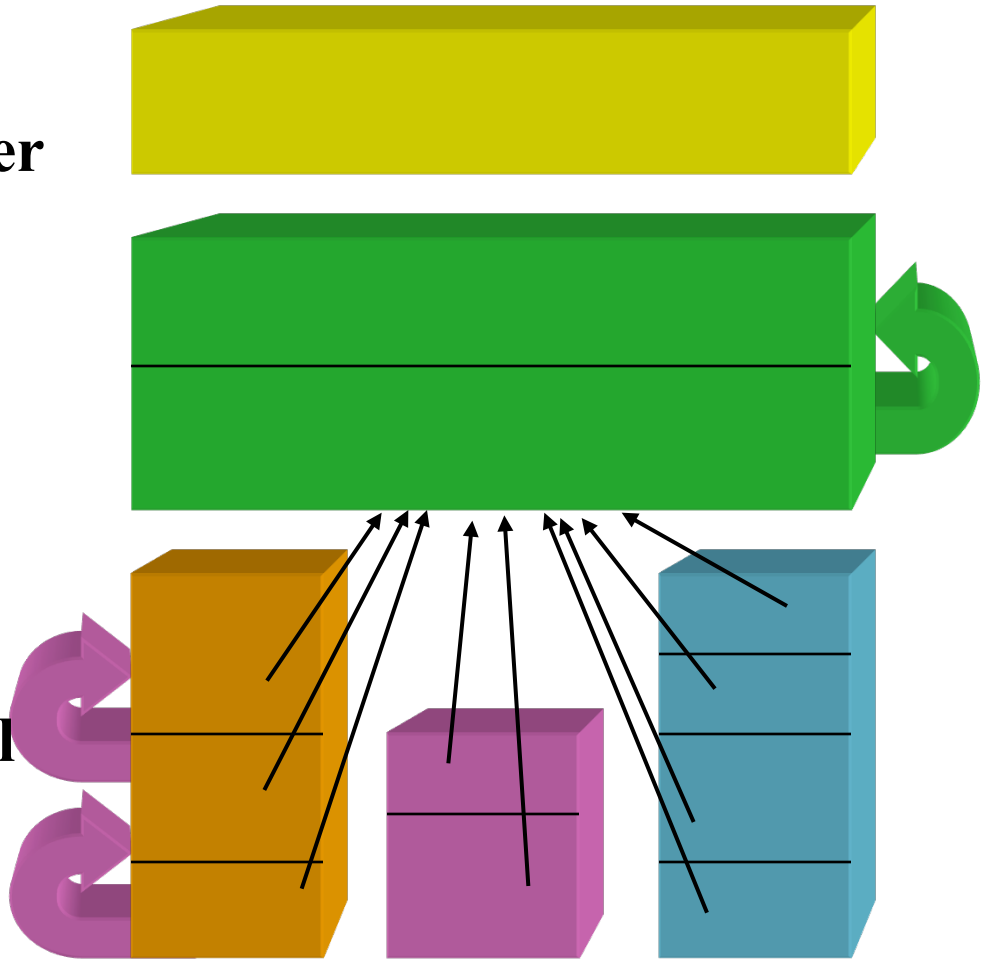
(300 MHz Pentium II, 128MB RAM, NT 4.0)

Multi-threaded GC

Static Objects
Created by Optimizer

Dynamic Objects
Shared by Multiple
Threads

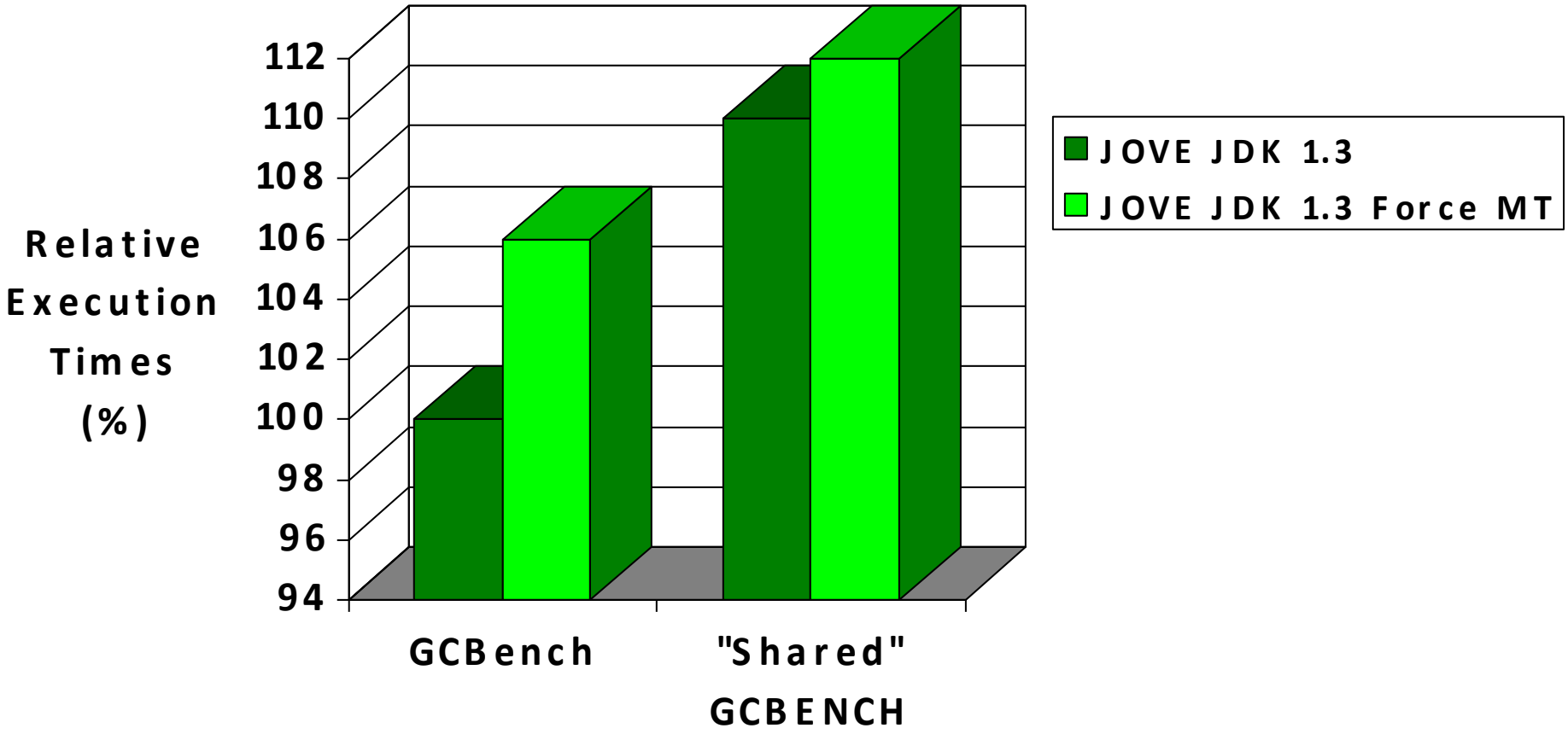
Dynamic Objects
Private to Individual
Threads



Basic Strategy

- Assumes that "most" objects are only accessed by the thread that creates them:
  - Each thread has a private multi-generational heap
  - All new objects allocated in a thread private heap
  - Private objects may reference same private heap or shared heap
  - Shared heap may only reference shared heap
  - Write barrier detects attempt to store private heap reference into shared object, this triggers...
  - Promotion of private object to youngest shared generation (transitive closure) --- no copying required
- Thread private heaps are locally collected- no synchronization
- Shared heap collection - stops the world

“Shared” GCBench Results

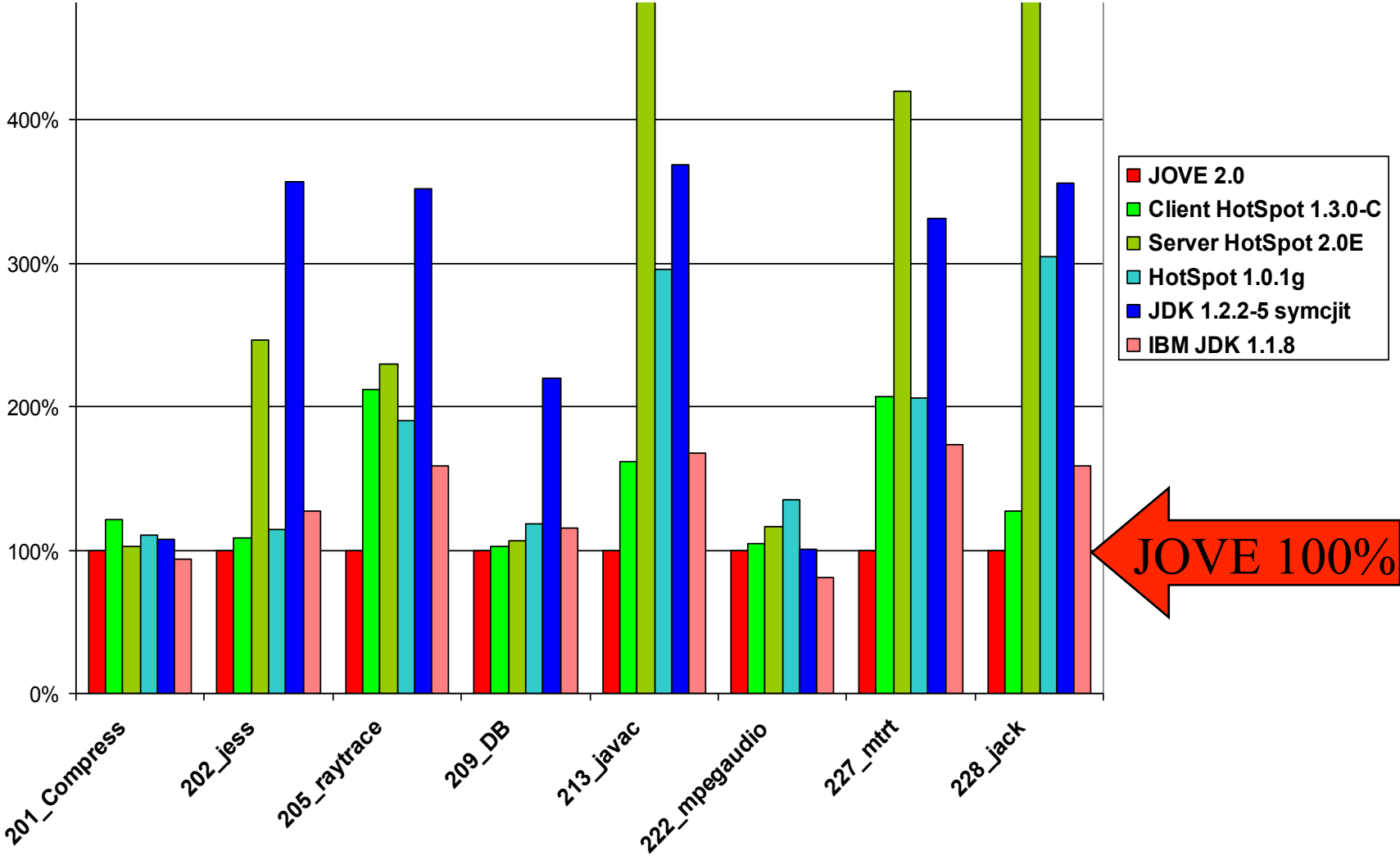


(300 MHz Pentium II, 128MB RAM, NT 4.0)

Future

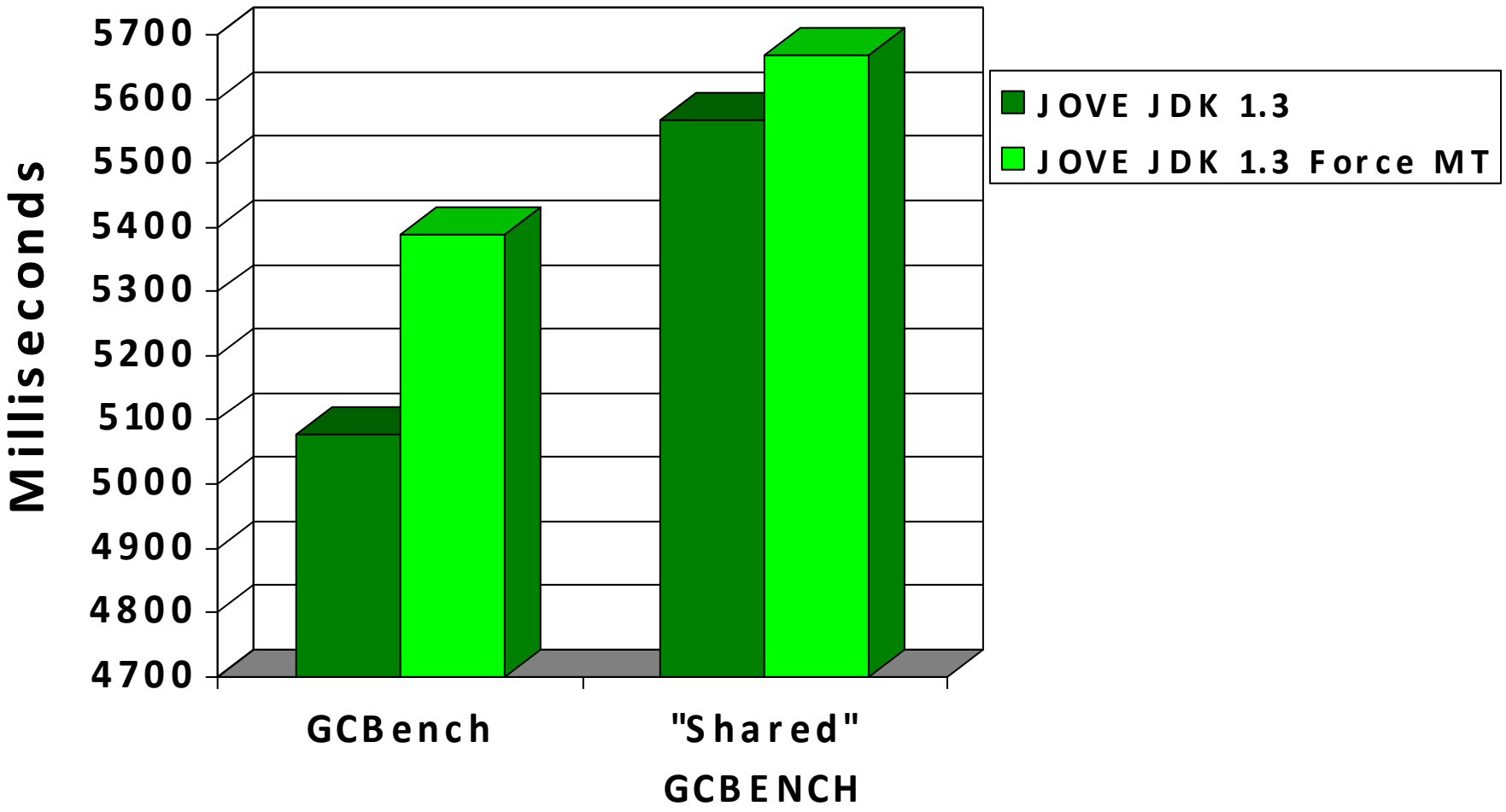
- How well does assumptions about thread independence match real programs?
- Apply same basic idea to “single-threaded” phase structured programs

Relative Execution times for SPECjvm98 Benchmarks on various JVMs Normalized to JOVE 2.0



Smaller is better

“Shared” GCBench Results



(300 MHz Pentium II, 128MB RAM, NT 4.0)