



Benchmarking Java Grande Applications

Mark Bull, Lorna Smith, Martin Westhead, David Henty and Robert Davey

EPCC, The University of Edinburgh
epcc-javagrande@epcc.ed.ac.uk

<http://www.epcc.ed.ac.uk/javagrande>

- ▶ Background
 - ▶ Related work
 - ▶ Description of the Suite
 - ▶ Performance metrics
 - ▶ Results
 - ▶ Conclusions
 - ▶ Acknowledgements
-

- ▶ Java Grande Forum
 - community initiative to promote the use of Java for “Grande” applications
- ▶ Benchmarking initiative
 - to develop a suite of benchmarks to measure different execution environments of Java
- ▶ This work
 - results and performance analysis
- ▶ Funding



partly funded by
UKHEC

- ▶ Small applets
 - testing JVMs in browsers
 - ▶ Performance based operations
 - e.g. arithmetic operations, method calls, object creation...
 - ParkBench, UCSD Benchmarks, Mathew, Coddington et al...
 - ▶ Computational kernels
 - SciMark, SPEC JVM98,
 - ▶ Full-scale applications
 - monte carlo, molecular dynamics, ray tracer....
-

The aim of the suite is to:

- ▶ Demonstrate the use of Java for Grande applications
 - ▶ Provide metrics for comparing Java execution environments
 - ▶ Expose features of the execution environment critical to Grande applications
-

▶ Section I

- low level operations
- Arith, Assign, Cast, Create, Exception, Loop, Serial, Math, Method

▶ Section II

- simple kernels
- Series, LUFact, HeapSort, SOR, Crypt, FFT, Sparse

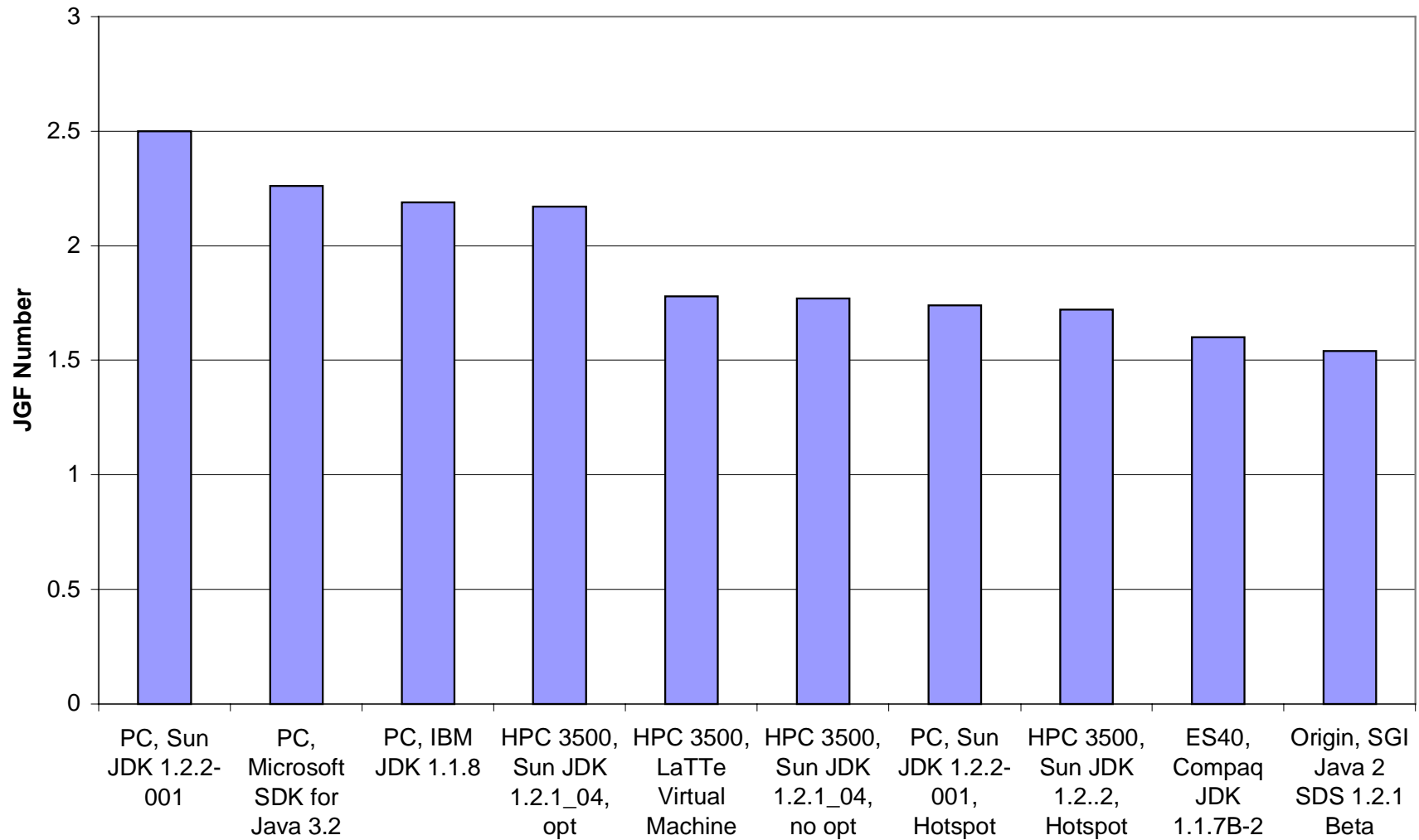
▶ Section III

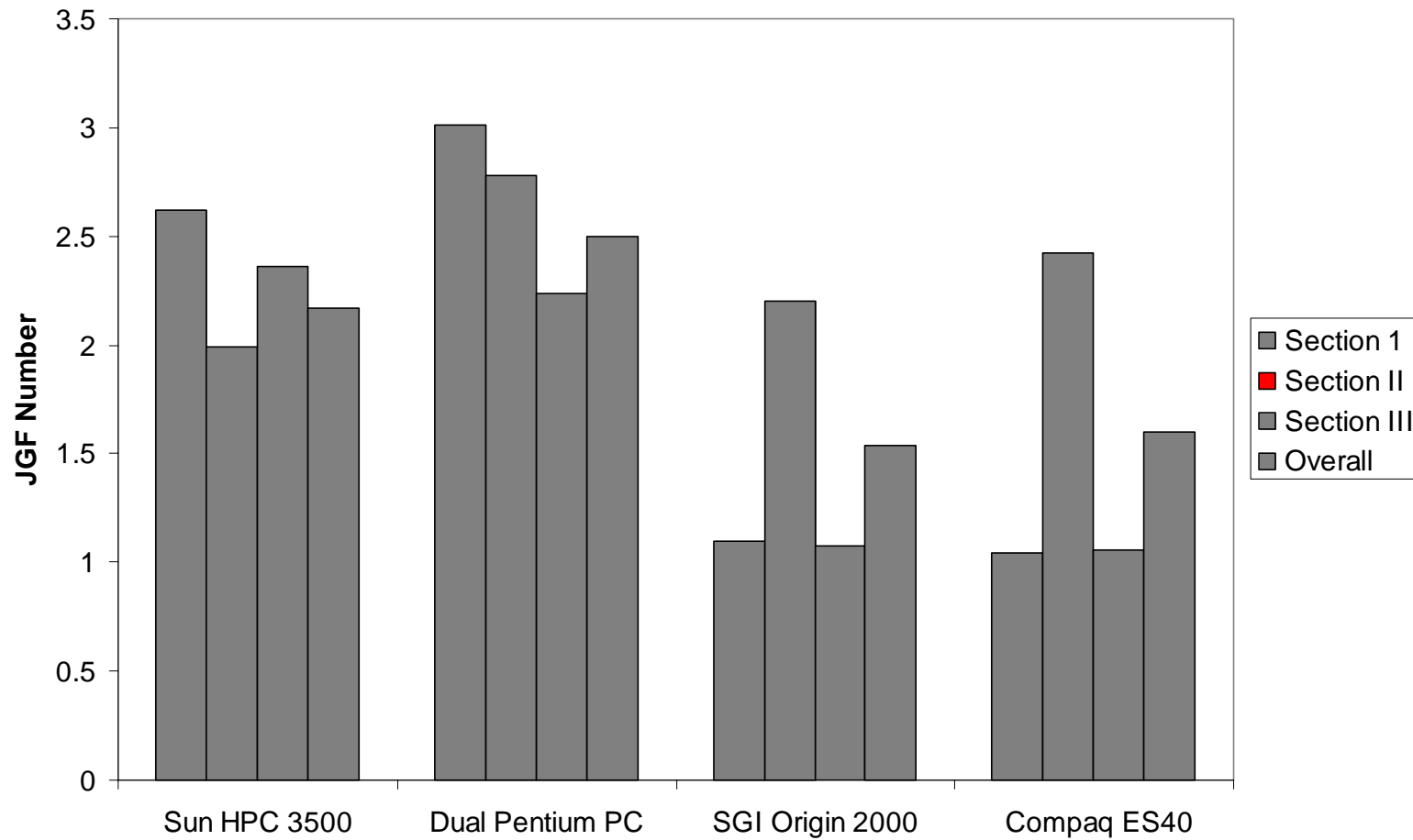
- application codes
 - Euler, MonteCarlo, MolDyn, Search, RayTracer
-

- ▶ Execution time
 - wall clock time of “interesting” computations
 - ▶ Temporal performance
 - operations per second
 - ▶ Relative performance
 - ratio of temporal performance to a reference system
 - Sun JDK Version 1.2.1_02 (production version) on a Sun Enterprise 3000 250Mhz UltraSPARC processor.
 - ▶ Average performance metrics
 - “overall” JGF Number: the average of Section II, Size B results and Section III, Size A results.
-

<p>Sun HPC 3500 Sun JDK 1.2.1_04, no opt Sun JDK 1.2.1_04, -Xoptimize Sun JDK 1.2.2, HotSpot VM 1.0.1 LaTTe Virtual Machine 0.9.0</p>	<p>Dual Pentium PC Sun JDK 1.2.2-001 Sun JDK 1.2.2-001, HotSpot VM 1.0.1 IBM JDK 1.1.8 Microsoft SDK for Java 3.2</p>
<p>SGI Origin 2000 SGI Java 2 SDK 1.2.1, Beta</p>	<p>Compaq ES40 Compaq JDK 1.1.7B-2</p>

- Sun HPC 3500: 400 Mhz UltraSPARC II, 8Gb RAM, solaris 7
 - Dual Pentium PC: 500 Mhz Pentium Pro, 512 Mb RAM, Windows NT
 - SGI Origin 2000: 300 MHZ R12000, 4Gb RAM, IRIX 6.5
 - Compaq ES40: 500 MHz EV6, 4Gb RAM, Digital UNIX V4.0F
-



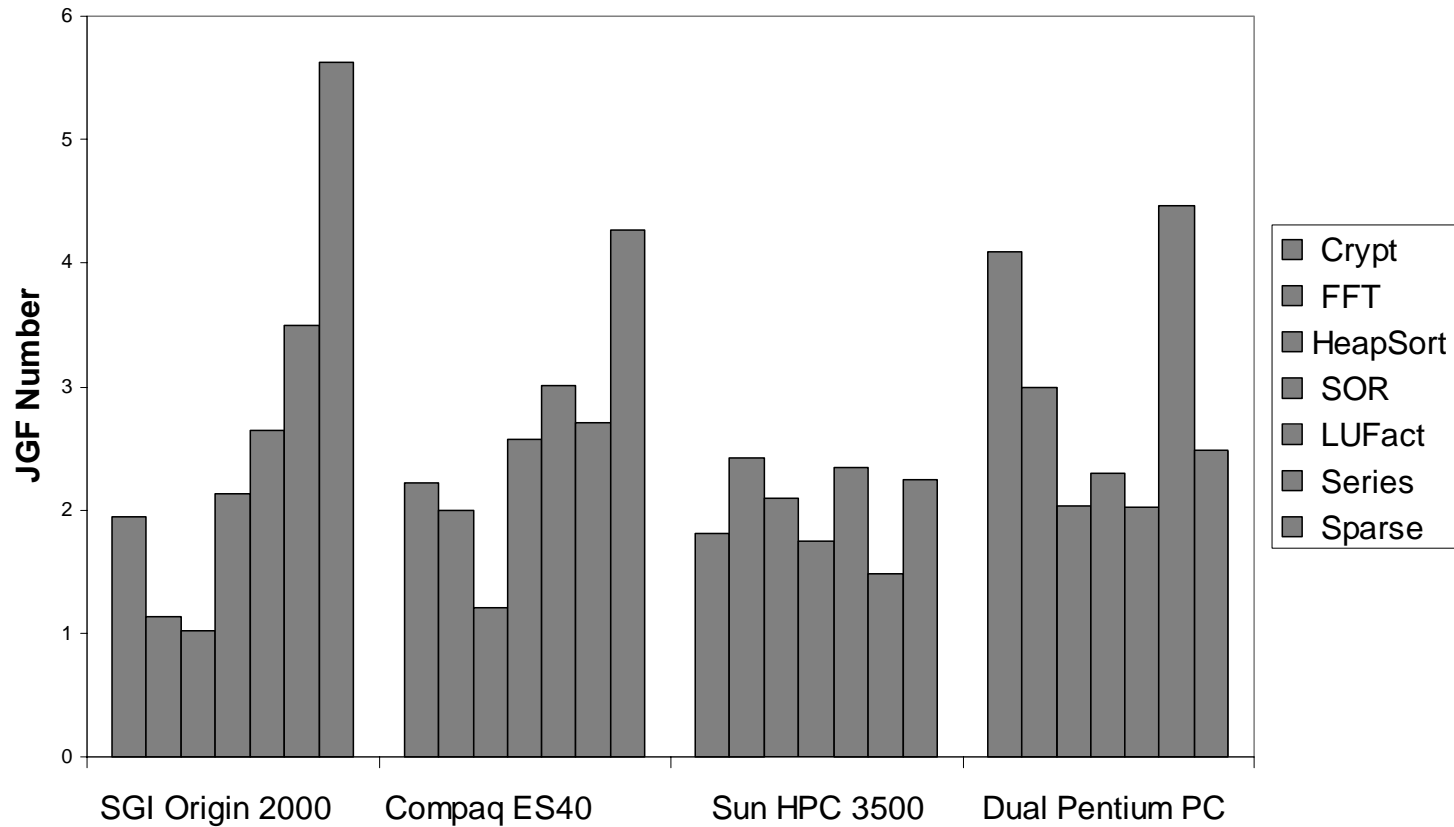


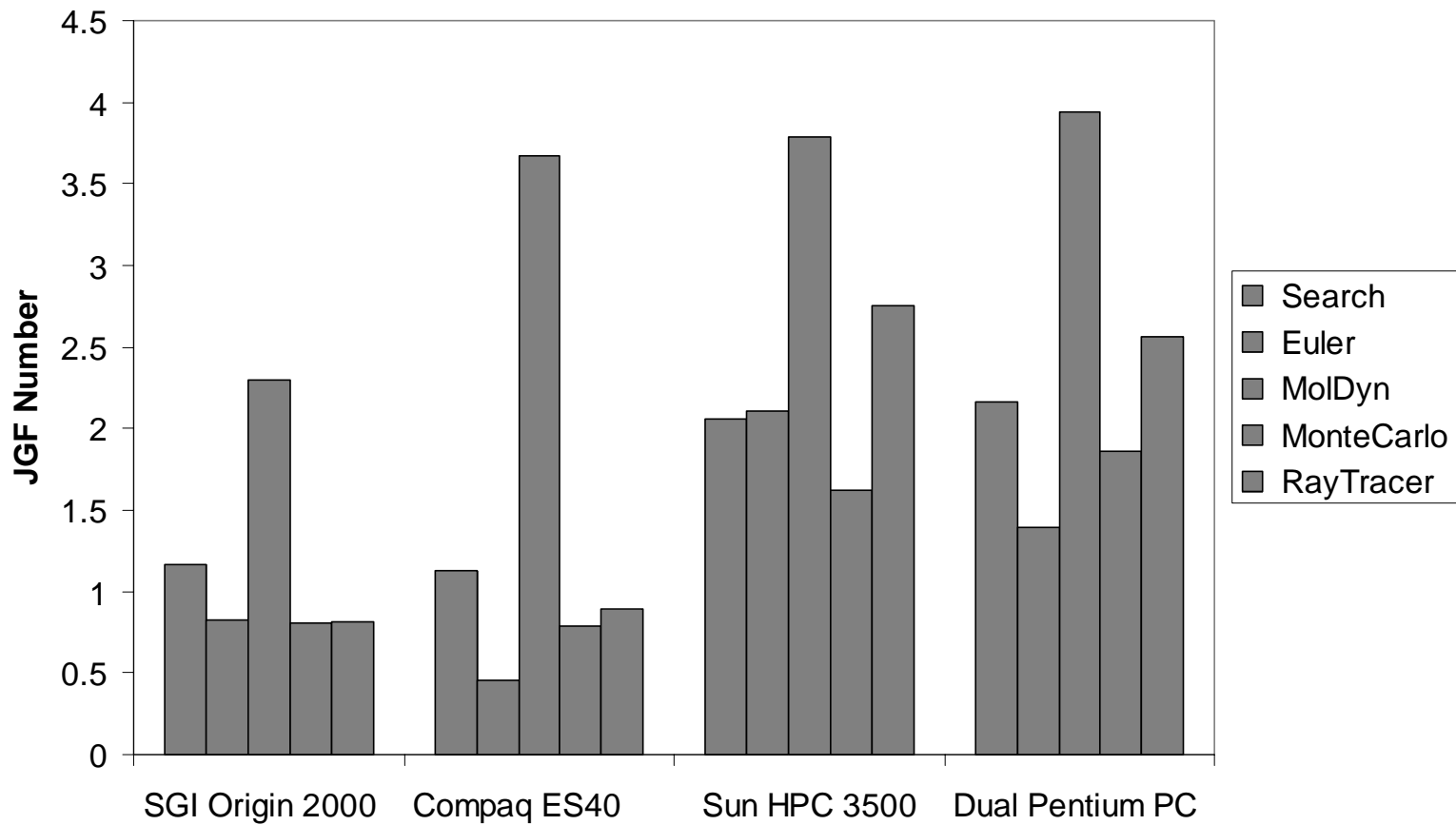
▶ Overall JGF Number

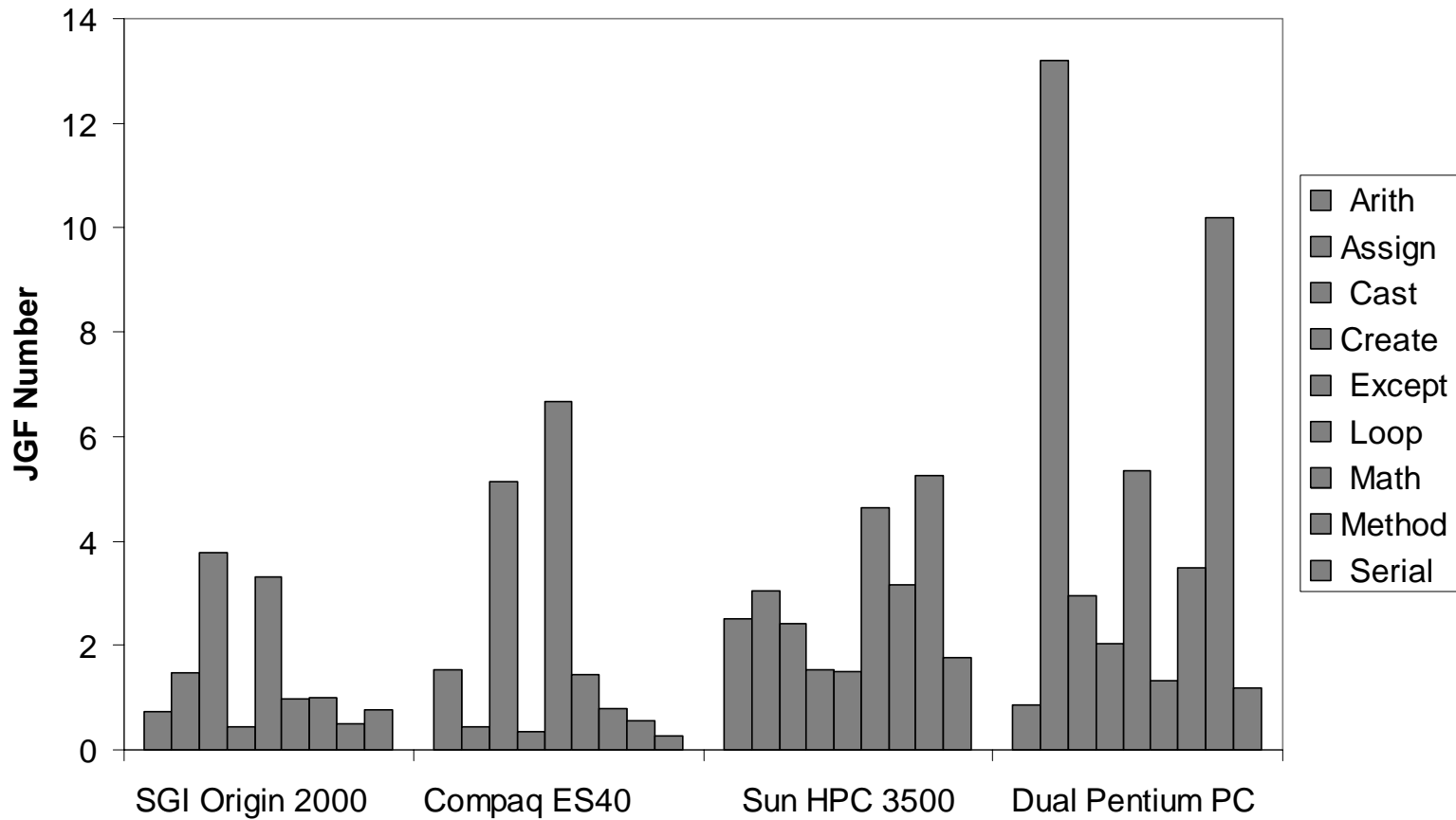
- Origin 2000 + Compaq ES40
 - good performance on Section II, relative to other sections
- Dual Pentium PC + Sun HPC 3500
 - good performance on both Section II + III

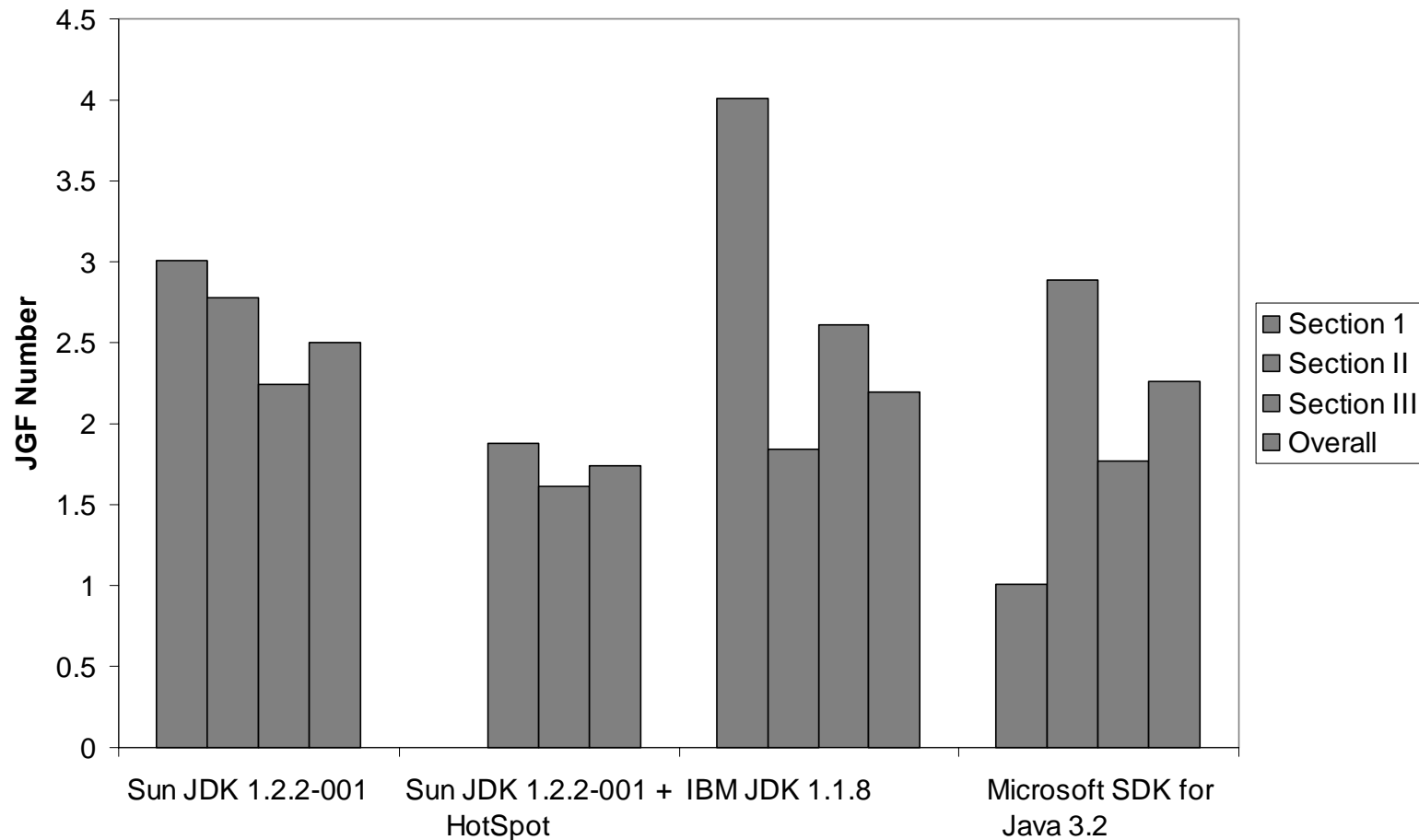
▶ Section 1

- Origin 2000, Compaq ES40
 - poor performance on Section I, relative to section II
 - Dual Pentium PC + Sun HPC 3500
 - relatively high Section I JGF Numbers
-

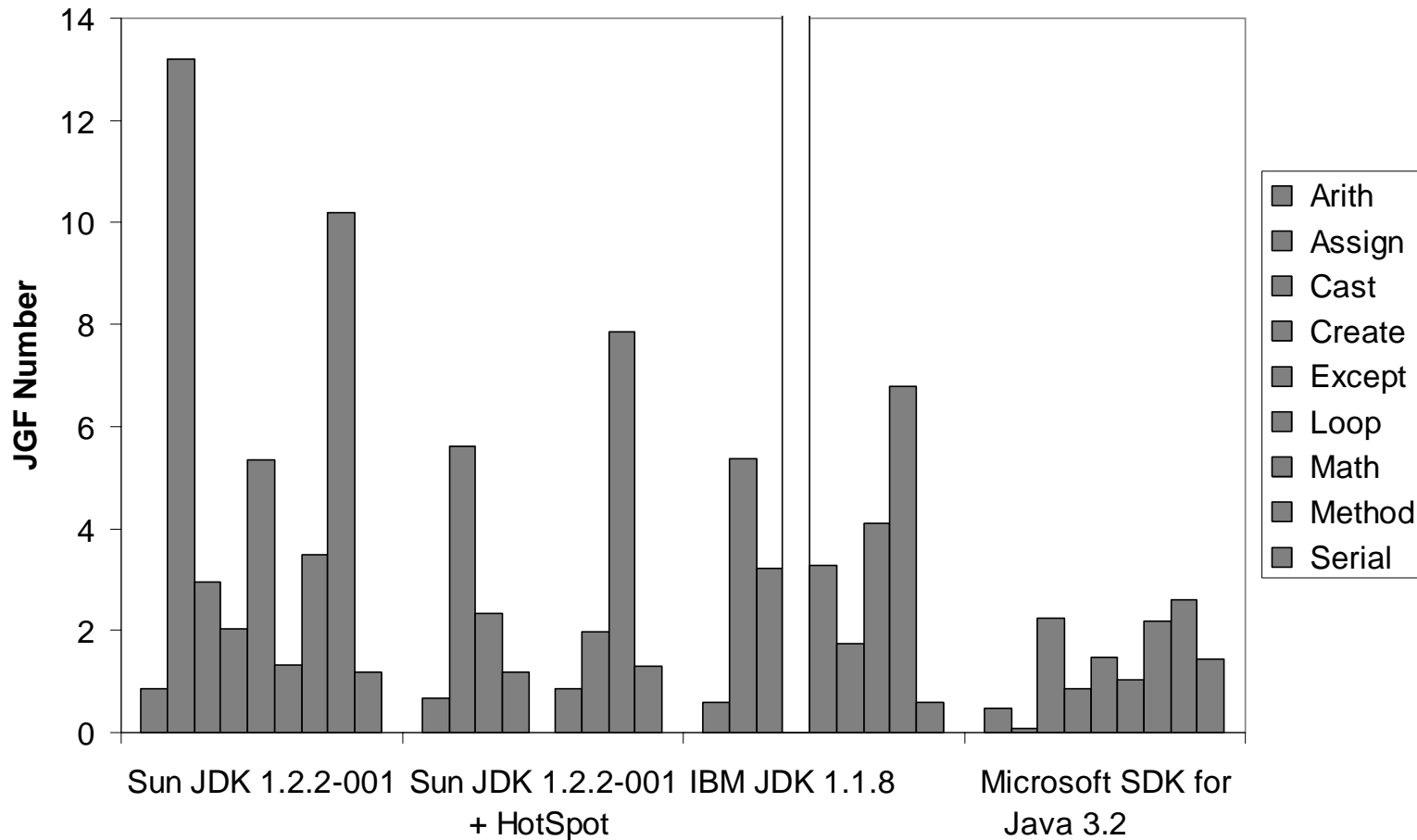


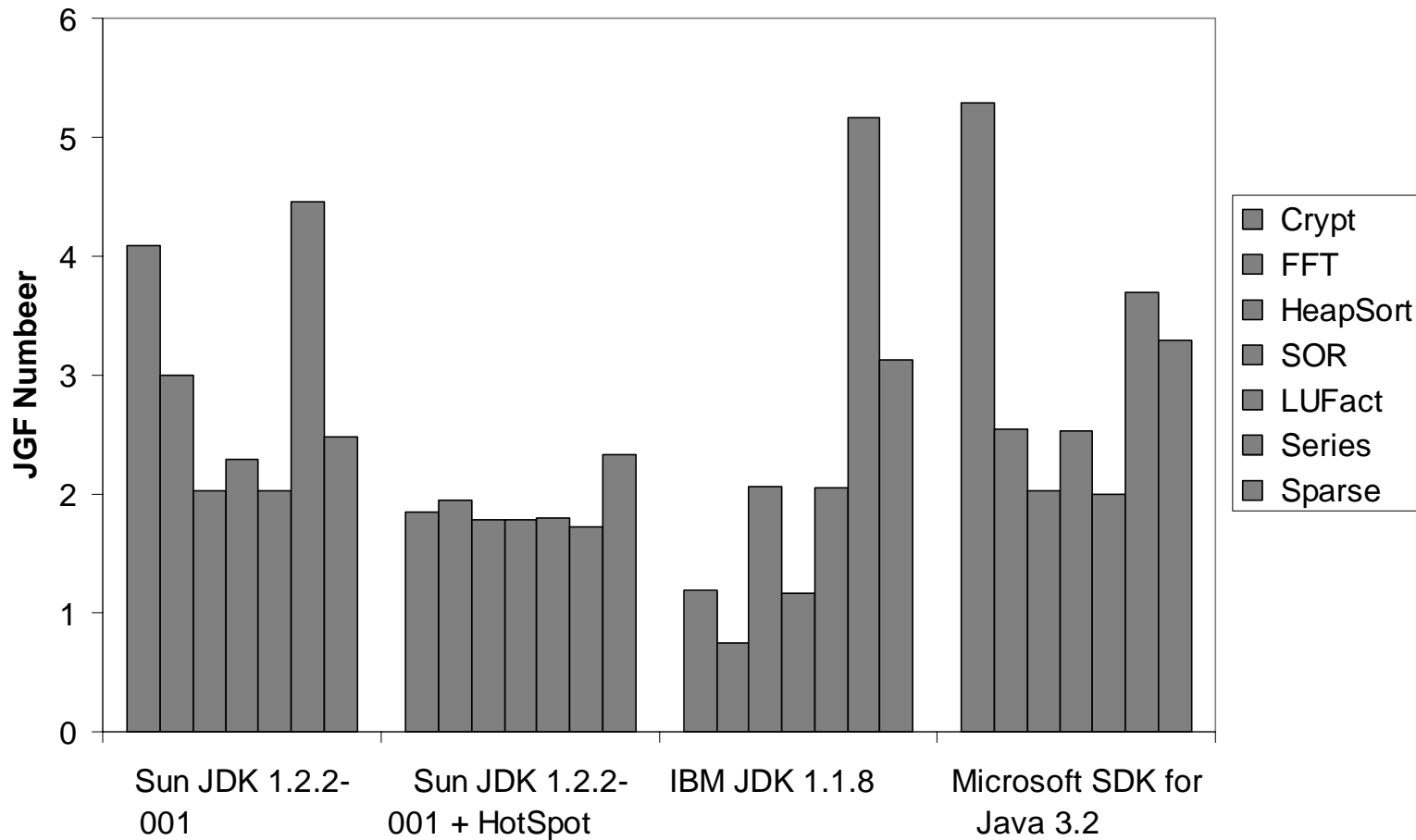


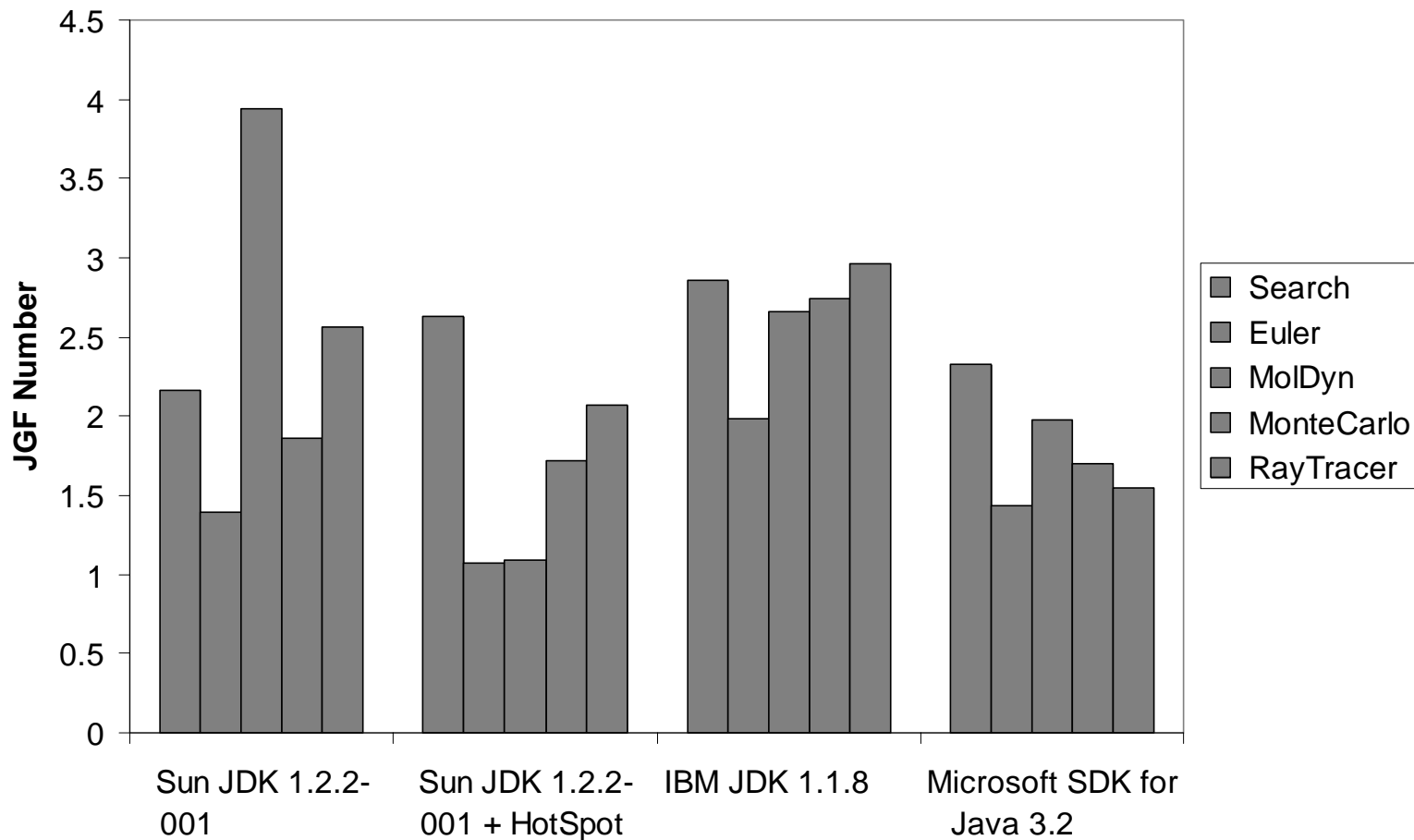


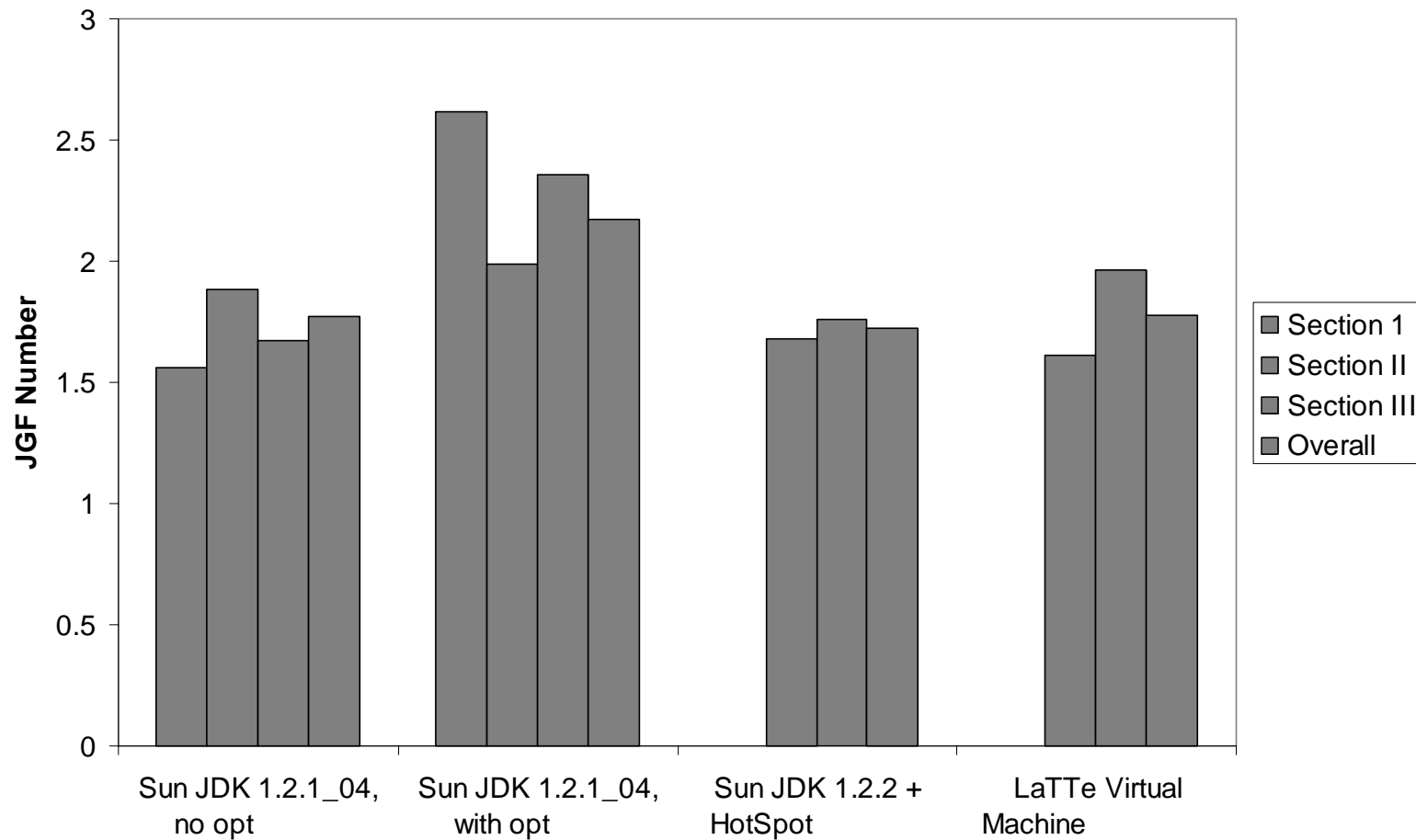


- ▶ Microsoft SDK for Java 3.2
 - outperforms Sun JDK on Section II
 - poor performance on Section I
 - ▶ IBM JDK 1.1.8
 - outperforms SUN JDK on Section III
 - very good performance on Section I
 - ▶ Sun JDK 1.2.2-001
 - gives best overall performance
 - more consistent performance on both sections II + III
 - consistent performance on Section I, relative to other sections
 - ▶ Sun JDK 1.2.2-001 + HotSpot
 - poor performance in comparison to Sun JDK 1.2.2-001
-







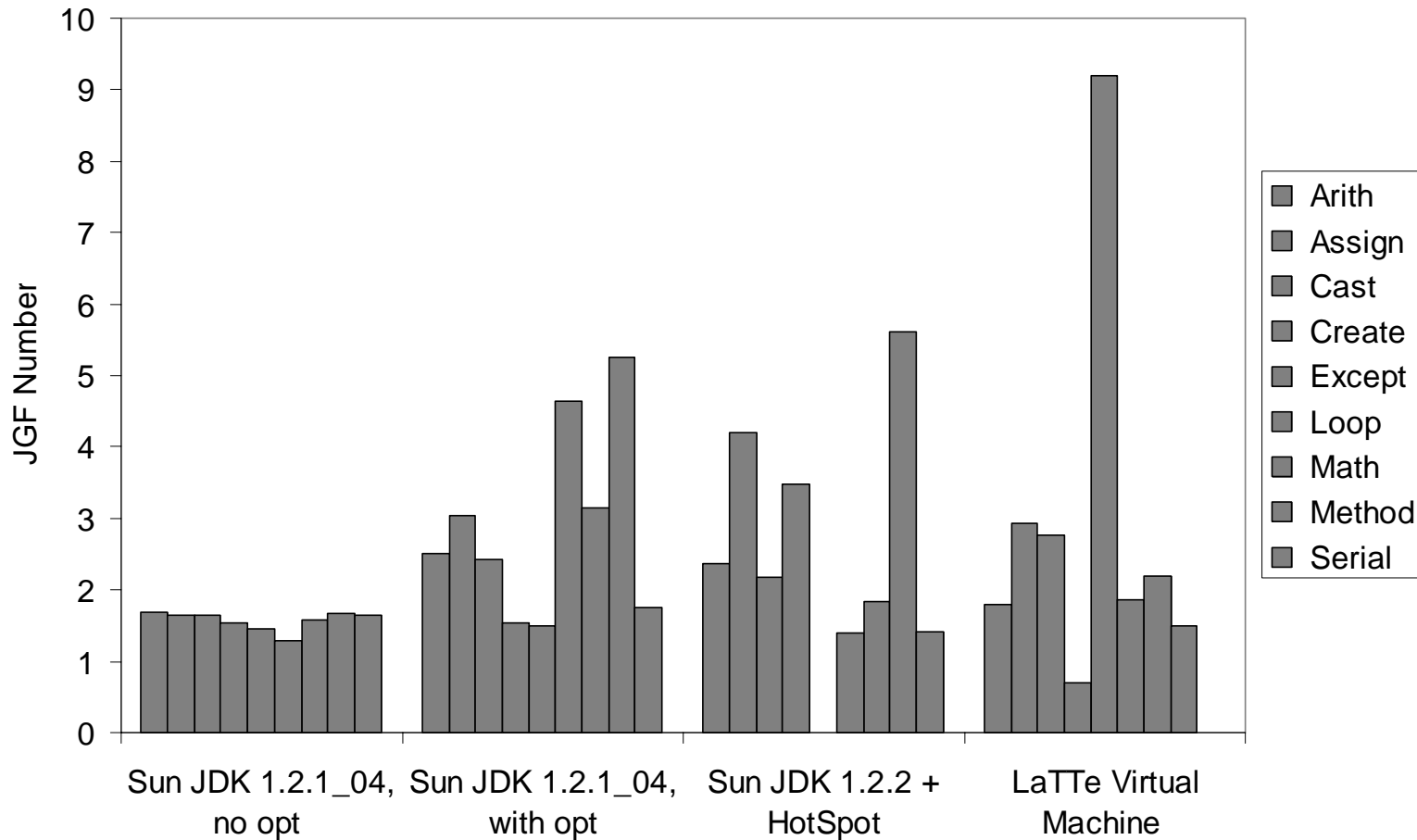


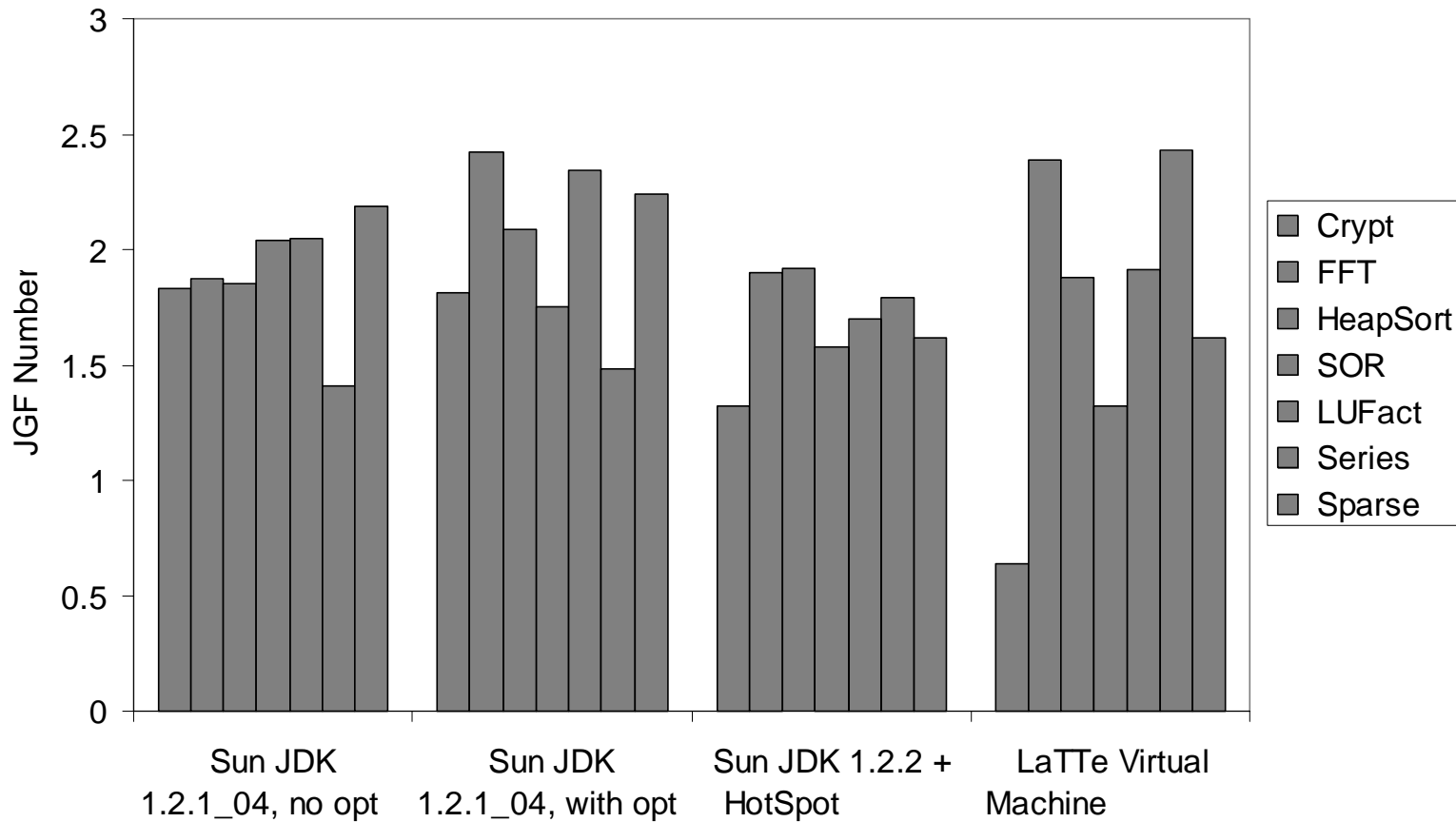
▶ Overall JGF Number:

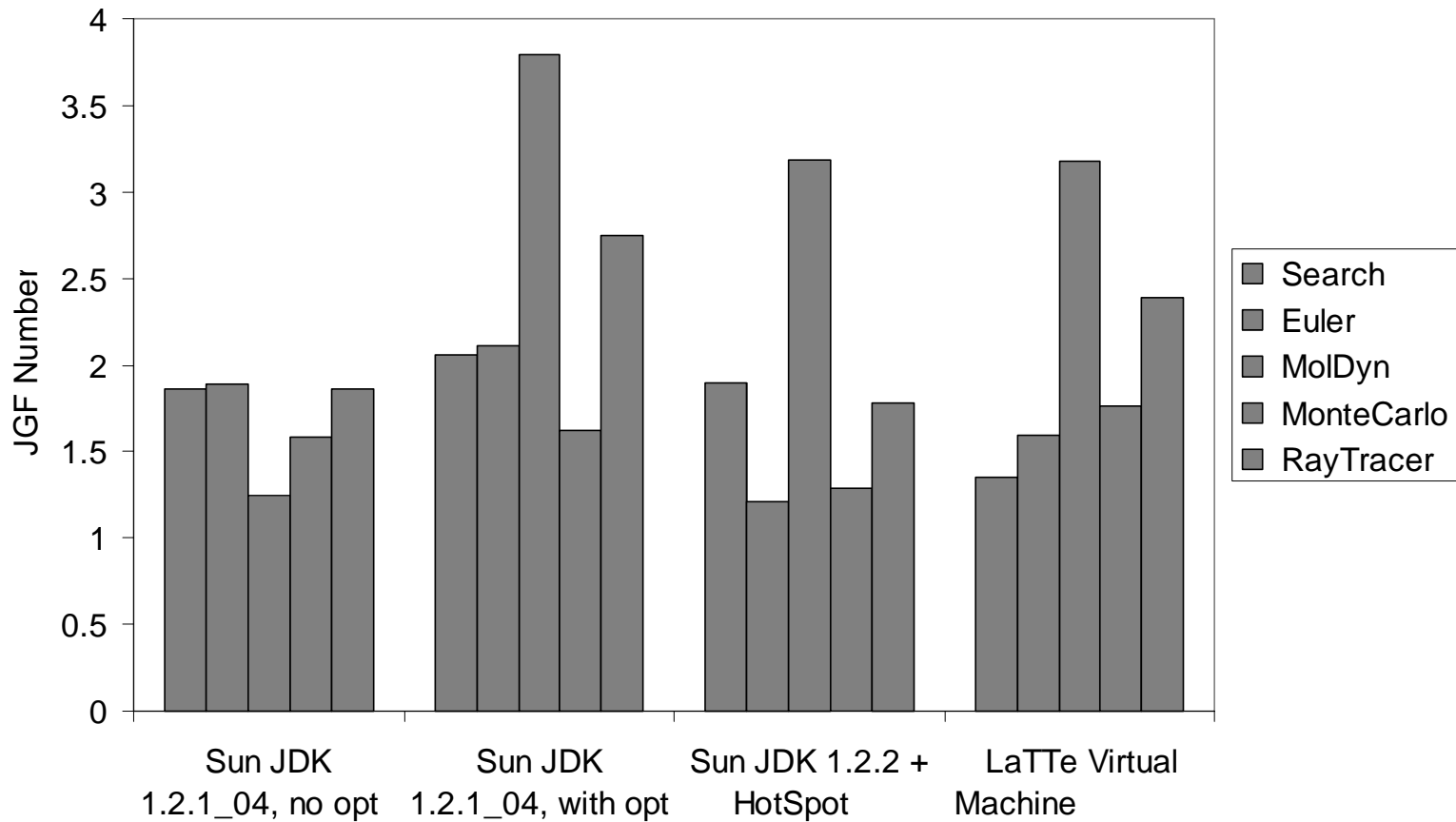
- Sun JDK 1.2.1_04 with run-time optimisation
 - increased JDK performance on all sections
 - particularly benefited Section III results
- Sun JDK 1.2.2 + HotSpot and LaTTe Virtual Machine
 - broadly comparable results to unoptimised Sun JDK 1.2.1_04

▶ Section I:

- Sun JDK 1.2.1_04 with run-time optimisation
 - increased JDK performance
 - Sun JDK 1.2.2 + HotSpot
 - broadly comparable results to unoptimised Sun JDK 1.2.1_04
-







- ▶ System comparisons
 - Origin 2000 + Compaq ES40
 - good performance on Section II, relative to other sections
 - Dual Pentium PC + Sun HPC 3500
 - good performance on all Sections
 - ▶ Dual Pentium PC
 - Sun JDK 1.2.2-001:
 - gives best overall performance, more consistent performance on all sections
 - ▶ Sun HPC 3500
 - Sun JDK 1.2.1_04 with run-time optimisation
 - increased JDK performance on all sections
 - Sun JDK 1.2.2 + HotSpot and LaTTe Virtual Machine
 - broadly comparable results to unoptimised Sun JDK 1.2.1_04
-

- ▶ We wish to thank the following who have contributed benchmarks to the suite:
 - Jesudas Mathew and Paul Coddington of the University of Adelaide
 - Roldan Pozo of NIST
 - Florian Doyan, Wilfried Klauser and Denis Caromel of INRIA
 - Hon Yau, formerly of EPCC
 - Gabriel Zachmann of the Fraunhofer Institute for Computer Graphics
 - Reed Wade of the University of Tennessee at Knoxville
 - John Tromp of CWI, Amsterdam
 - David Oh, formerly of MIT's Computational Aerospace Sciences Laboratory

 - ▶ The authors gratefully acknowledge the assistance of:
 - Philip Snowdon, Cray
 - Harvey Richardson, Sun
-

- ▶ Java Grande Benchmarking
 - <http://www.epcc.ed.ac.uk/javagrande>
 - ▶ Java Grande Forum
 - <http://www.javagrande.org>
 - ▶ UKHEC
 - <http://www.ukhec.ac.uk/>
-