

c^ϵ ,

Thomas VanDrunen^K, Daniel von Dincklage^ζ, Ben Wiedermann^δ

^αIntel, ^βAustralian National University, ^γUniversity of Massachusetts at Amherst, ^δUniversity of Texas at Austin,

University of New Mexico, ^ζUniversity of Colorado, ^ηTufts, ^θIBM TJ Watson Research Cent/F940(.)]TJ/F92, 6.974 Tf 125.022 3.254 Td[(i)]TJ/F91 8.966 Tf 2.793 -3.254

The DaCapo Benchmarks: Java Benchmarking Development and Analysis (*Extended Version*)

Stephen M Blackburn^{α β}, Robin Garner^β, Chris Hoffmann^γ, Asjad M Khan^γ, Kathryn S McKinley^δ,

Rotem Bentzur^ε, Amer Diwan^ζ, Daniel Feinberg^ε, Daniel Franklin^θ, Samuel Z Guyer^η, Martin Hirzel^θ,

Antony Hosking^ι, Maria Jump^δ, Han Lee^α, J Eliot B Moss^γ, Aashish Phansalkar^δ, Darko Stefanovi

the DaCapo benchmarks are not definitive, and they may or may not be representative of workloads that vendors and clients care about most. Regardless, we believe this paper is a step towards a wider community discussion and eventual consensus on how to select, measure, and evaluate benchmarks, VMs, compilers, runtimes, and hardware for Java and other managed languages.

2. Related Work

We build on prior methodologies and metrics, and go further to recommend how to use them to select benchmarks and for best practices in performance evaluation.

2.1 Java Benchmark Suites

In addition to SPEC (discussed in Section 3), prior Java benchmarks suites include Java Grande [26], Jolden [11, 34], and Ashes [17]. The Java Grande Benchmarks include programs with large demands for memory, bandwidth, or processing power [26]. They focus on array intensive programs that solve scientific computing problems. The programs are sequential, parallel, and distributed. They also include microbenchmark tests for language and

variable workloads make performance hard to analyze and reason about. For example, the level and number of classes optimized and

.....
1 -

(a) SPEC



How well these large objects are handled will thus in large part determine the performance of the collector on *xalan*.

	Rank
--	------

are publicly available, evolving, and have and will remain open to public feedback [14].

Acknowledgments

We thank Andrew Appel, Randy Chow, Frans Kaashoek, and Bill

Benchmark Description and Origin

Short Description	A parser generator and translator generator.
Long Description	ANTLR parses one or more grammar files and generate a parser and lexical analyzer for each.
Threads	Single threaded

Benchmark Description and Origin

Short Description A graph plotting toolkit and pdf renderer

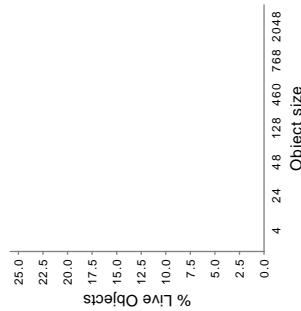
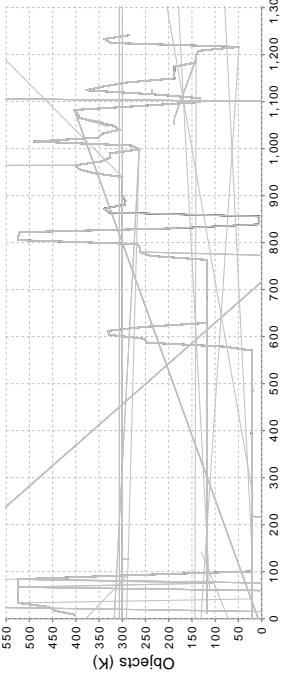
Benchmark Description and Origin

Short Description	An integrated development environment
Long Description	Run a series of eclipse jdt (non-gui) performance tests
Threads	Workload is single threaded, but Eclipse uses multiple threads internally
Repeats	Single iteration, performs multiple distinct Eclipse tasks
Version	3.1.2
Copyright	Eclipse Foundation

Long Description fop takes an XSL-FO file, parses it and formats it, generating an encrypted pdf

Benchmark Description and Origin

Short Description An output-independent print formatter



<i>Benchmark Description and Origin</i>	
Short Description	A python interpreter written in Java
Long Description	jython executes (interprets) the pybench benchmark or a small python program
Threads	Single threaded
Repeats	Single iteration runs a single iteration of the pybench python benchmark
Version	2.1
Copyright	Copyright (c) Python Software Foundation, Jim Hugunin and Barry Warsaw
Author	Jim Hugunin and Barry Warsaw
License	Jython Software License.

<i>Benchmark Characteristics</i>	
Total Allocation (MB)	1,183.4
(Obj)	25,940,819
Maximum Live (MB)	0.1
(Obj)	2,788
Pointer Mutations (M)	82.96
Classes_Loaded	251

<i>Benchmark Description and Origin</i>	
Short Description	A source code analyzer for Java

Benchmark Description and Origin

Short Description	An XSLT processor for transforming XML documents
Long Description	Xalan transforms an XML document (either a test case or the works of Shakespeare) and transforms the document into html
Threads	Single threaded
Repeats	8 iterations, each transforms a single large XML document (the works of Shakespeare)
Version	2.4.1

<i>Benchmark Description and Origin</i>	
Short Description	
Long Description	
Threads	
Repeats	
Version	
Copyright	
Author	
License	

<i>Benchmark Characteristics</i>	
Total Allocation (MB)	262.0
(Obj)	7,955,141
Maximum Live (MB)	1.2
(Obj)	22,150
Classes Loaded	184

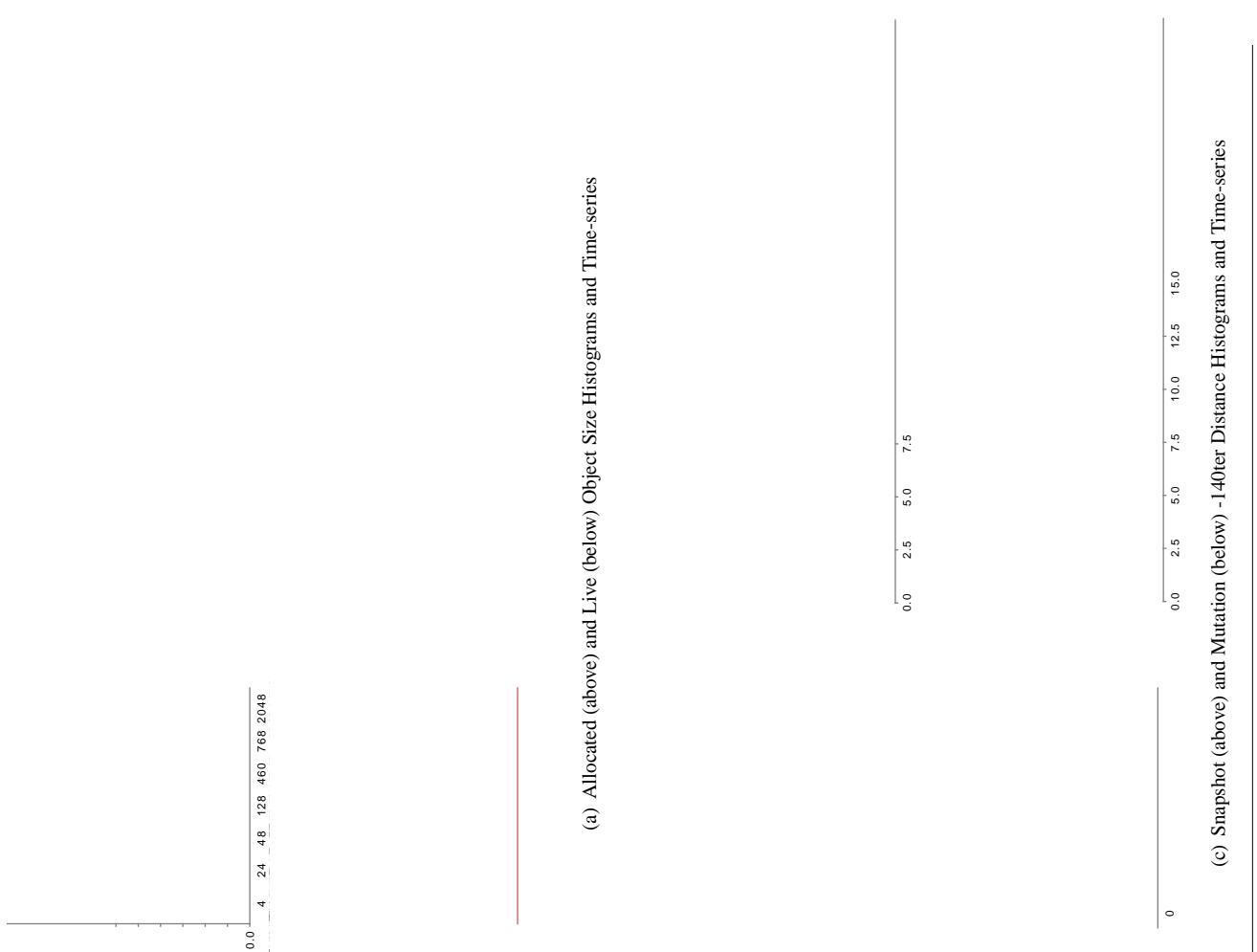


Figure 16.

Benchmark Description and Origin

Short Description	
Long Description	
Threads	
Repeats	
Version	
Copyright	
Author	
License	

Benchmark Characteristics

Total Allocation (MB)	74.6
(Obj)	3,218,642
Maximum Live (MB)	8.5
(Obj)	291,681
Pointer Mutations (N)	33,07
Classes Loaded	85

Benchmark Description and Origin

Short Description

Benchmark Description and Origin

Short Description
Long Description
Threads
Repeats
Version
Copyright
Author
License

Benchmark Characteristics

Total Allocation (MB)	0.7
(Obj)	3,022
Maximum Live50((MB))	1334(0.7) ⁶

Benchmark Description and Origin

Short Description
Long Description
Threads
Repeats
Version

Benchmark Description and Origin

Short Description	
Long Description	
Threads	
Repeats	
Version	
Author	

Benchmark Characteristics

Total Allocation (MB)	270.7
(Obj)	9,393,097
Pointer Mutations (M d O J 0.398 w25)	11,949
Classes Loaded J 0.39817	

Benchmark Description and Origin