Benchmarking Computer Systems

Tomas Kalibera

PURDUE
UNIVERSITY

My benchmarking background

Systems

CORBA, Mono, RT Java (Ovm, RTS, WebSphere), Java (OpenJDK, Ovm, JikesRVM), R (FastR, GNU-R)

Benchmarks

SciMark, CORBA micro, CSiBE, FFT, SPEC JVM, SPEC JBB, SPEC CPU, DaCapo, Shootout, AT&T, Collision Det.

Methodology

Planning multi-level experiments with non-determinism, impact of code-layout, ratio estimation, concurrency and memory sharing metrics

Teaching, PhD

Most PL/systems papers include execution time measurements

Papers from 2011

	Papers	**Measured time**
ISMM	13	12
PLDI	55	42
ASPLOS	32	25
TOPLAS	13	5
TACO	9	6
Total	**122**	**90**

Kalibera, Jones: Quantifying performance changes with effect size confidence intervals. http://www.cs.kent.ac.uk/pubs/2012/3233/

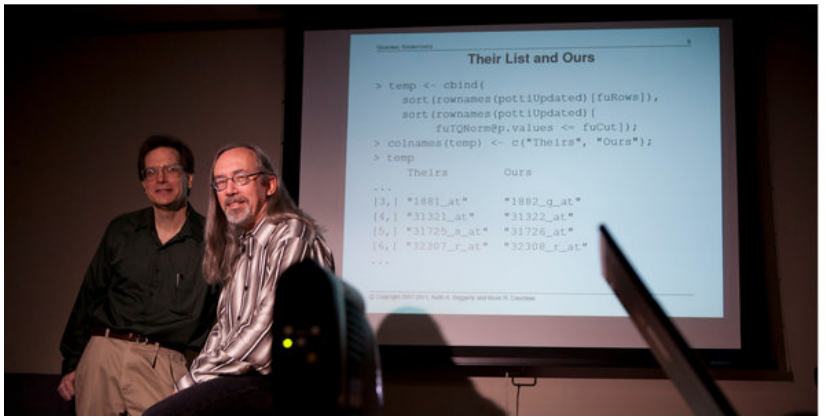
Practices (not?) to follow

In 2006, Potti and Nevins at Duke reported they could predict lung cancer using expression arrays, and started a company

By 2010, 3 major papers were retracted, the company was gone, Potti resigned, and a major investigation was underway

Due to a combination of bad science ranging from fraud, unsound methods, to off-by-one errors in Excel spreadsheets

Uncovered by a repetition study conducted by statisticians Baggerly and Coombes with access to raw data and 2,000 hours of effort

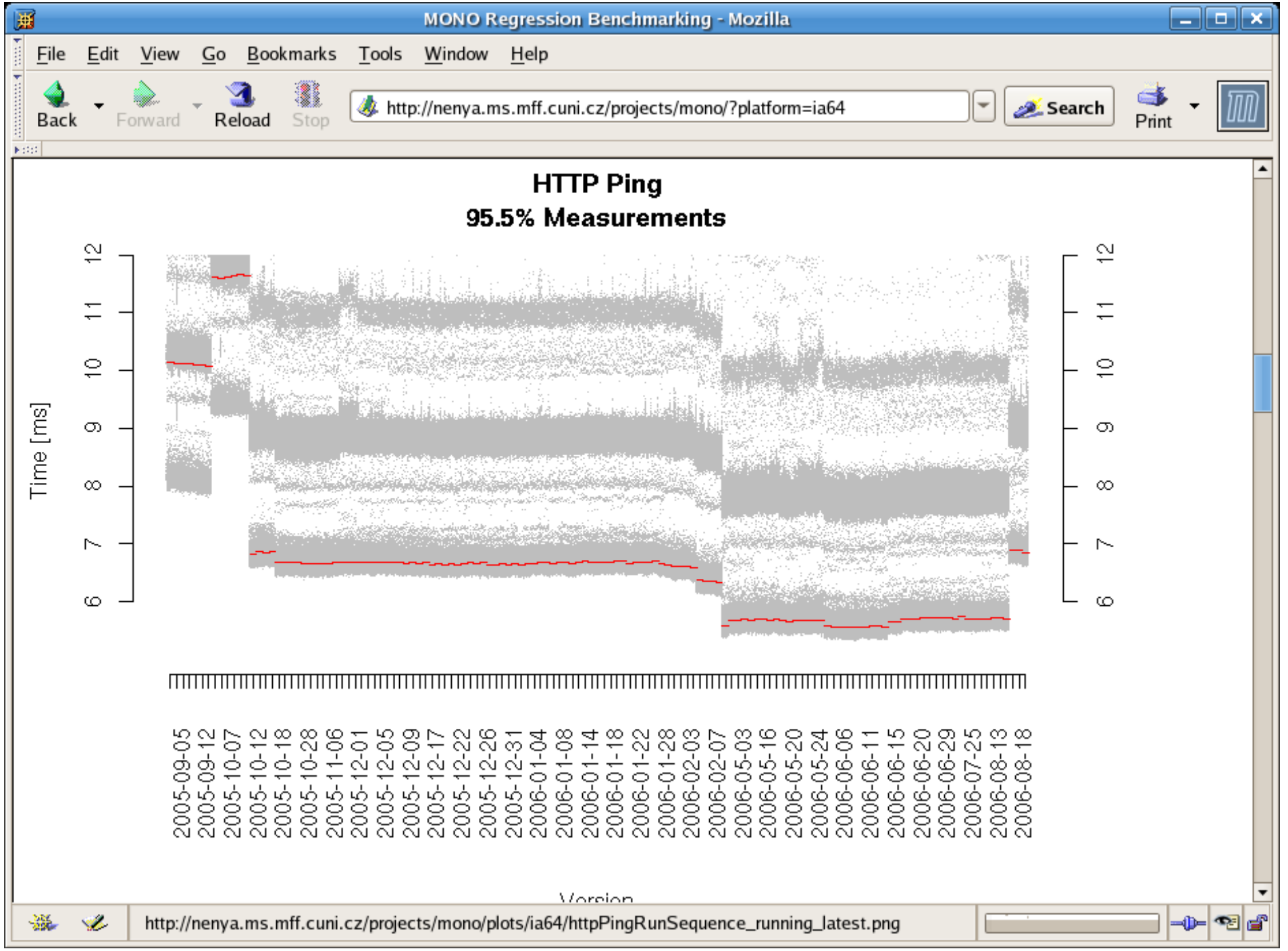


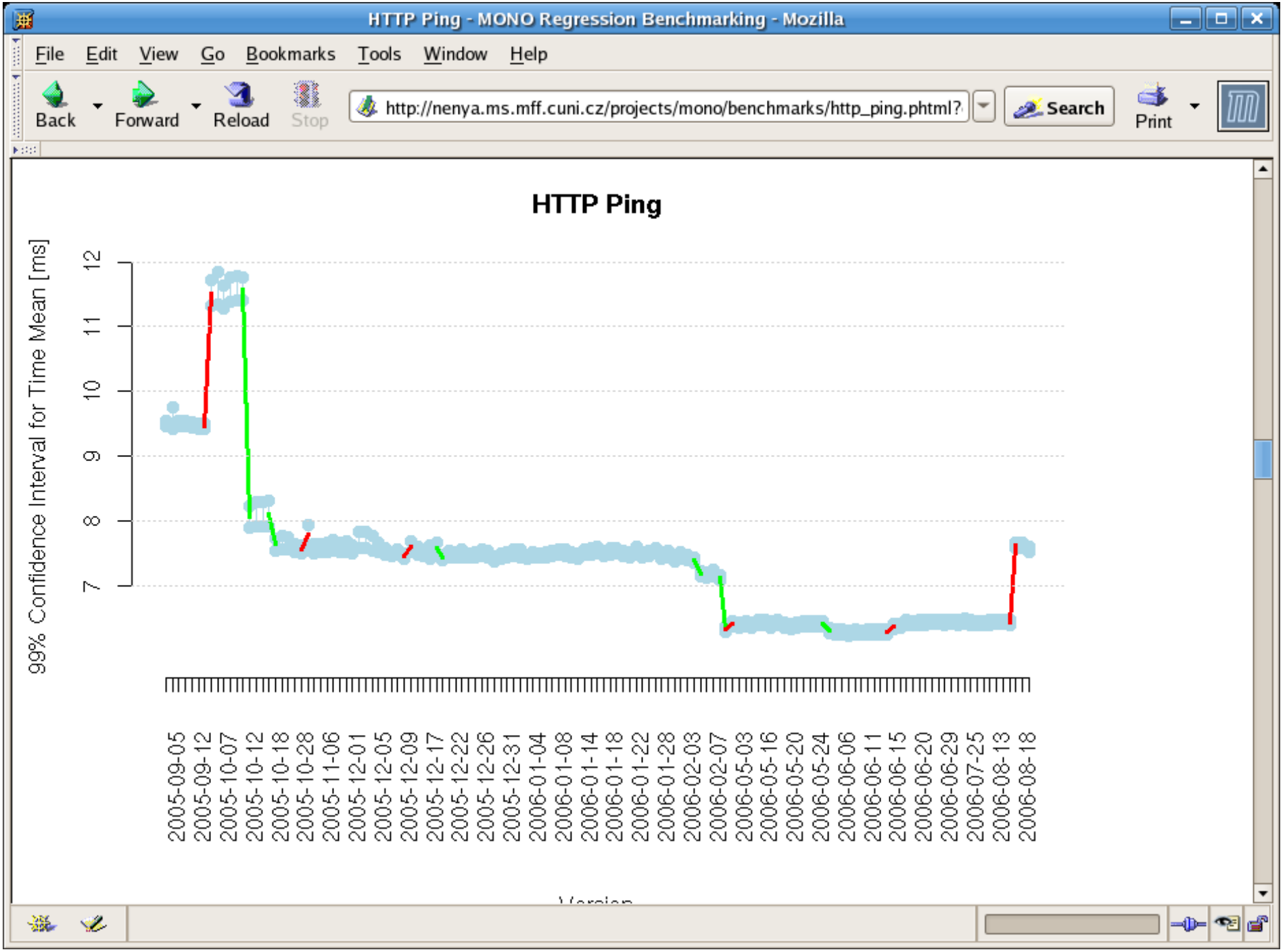
Experimental results in PL/systems papers ignore elementary statistics

Papers from 2011

	Papers	Measured time	No error estimate (ignored uncertainty)	
ISMM	13	12	5	**42%**
PLDI	55	42	39	**93%**
ASPLOS	32	25	18	**72%**
TOPLAS	13	5	4	**80%**
TACO	9	6	5	**83%**
Total	**122**	**90**	**71**	**79%**

Kalibera, Jones: Quantifying performance changes with effect size confidence intervals. http://www.cs.kent.ac.uk/pubs/2012/3233/

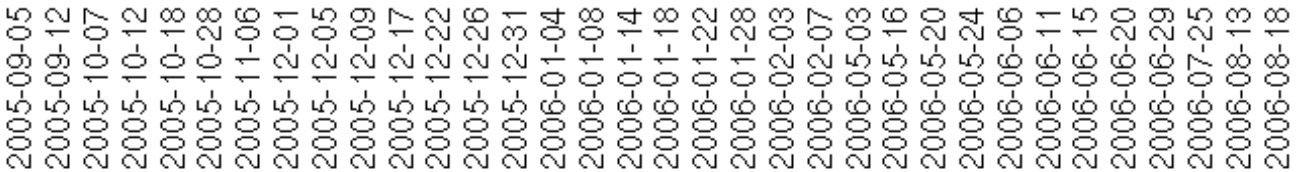

MONO Regression Benchmarking - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://nenya.ms.mff.cuni.cz/projects/mono/?platform=ia64
Search
Print
HTTP Ping
95.5% Measurements
Time [ms]
6 7 8 9 10 11 12
2005-09-05 2005-09-12 2005-10-07 2005-10-12 2005-10-18 2005-10-28 2005-11-06 2005-12-01 2005-12-05 2005-12-09 2005-12-17 2005-12-22 2005-12-26 2005-12-31 2006-01-04 2006-01-08 2006-01-14 2006-01-18 2006-01-22 2006-01-28 2006-02-03 2006-02-07 2006-05-03 2006-05-16 2006-05-20 2006-05-24 2006-06-06 2006-06-11 2006-06-15 2006-06-20 2006-06-29 2006-07-25 2006-08-13 2006-08-18
Version
http://nenya.ms.mff.cuni.cz/projects/mono/plots/ia64/httpPingRunSequence_running_latest.png


HTTP Ping - MONO Regression Benchmarking - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://nenya.ms.mff.cuni.cz/projects/mono/benchmarks/http_ping.phtml?
Search
Print
HTTP Ping
99% Confidence Interval for Time Mean [ms]
7 8 9 10 11 12
2005-09-05 2005-09-12 2005-10-07 2005-10-12 2005-10-18 2005-10-28 2005-11-06 2005-12-01 2005-12-05 2005-12-09 2005-12-17 2005-12-22 2005-12-26 2005-12-31 2006-01-04 2006-01-08 2006-01-14 2006-01-18 2006-01-22 2006-01-28 2006-02-03 2006-02-07 2006-05-03 2006-05-16 2006-05-20 2006-05-24 2006-06-06 2006-06-11 2006-06-15 2006-06-20 2006-06-29 2006-07-25 2006-08-13 2006-08-18

HTTP Ping - MONO Regression Benchmarking - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop http://nenya.ms.mff.cuni.cz/projects/mono/benchmarks/http_ping.phtml? Search Print



Version

Newer Version	Older Version	Change Impact[%]	
2005-10-04	2005-09-12	21.76 %	diff used-class-libraries-diff
2005-10-11	2005-10-10	-30.28 %	diff used-class-libraries-diff
2005-10-17	2005-10-14	-5.82 %	diff used-class-libraries-diff
2005-10-28	2005-10-25	3.01 %	diff used-class-libraries-diff
2005-12-09	2005-12-08	1.93 %	diff used-class-libraries-diff
2005-12-19	2005-12-17	-1.8 %	diff used-class-libraries-diff
2006-02-04	2006-02-03	-2.68 %	diff used-class-libraries-diff
2006-04-27	2006-02-07	-11.12 %	diff used-class-libraries-diff
2006-04-30	2006-04-27	1.3 %	diff used-class-libraries-diff
2006-06-03	2006-05-24	-1.58 %	diff used-class-libraries-diff
2006-06-14	2006-06-13	1.43 %	diff used-class-libraries-diff
2006-08-16	2006-08-15	18.39 %	diff used-class-libraries-diff

http://nenya.ms.mff.cuni.cz/projects/mono/diffs...?old=2005-10-14&new=2005-10-17&bench=http_ping

Experimental evaluation ingredients

Know the goal

Know the system

Know the real applications, benchmarks

Run initial experiments, explore data

Choose appropriate statistics

Plan, dimension experiments

Automate experiments execution

Validate findings

Know the goal

"Our speed-up is 5%"

$$1 - \frac{\text{time on new system}}{\text{time on old system}}$$

Papers from ISMM, PLDI, ASPLOS, TOPLAS, TACO (2011)

	Papers	Measured time	Reported speedup
Total	122	90	65

Over 70% of papers that measured time reported speed-up

Kalibera, Jones: Quantifying performance changes with effect size confidence intervals. http://www.cs.kent.ac.uk/pubs/2012/3233/

Know the goal

"Our speed-up is 5% ± 1.5% with 95% confidence"

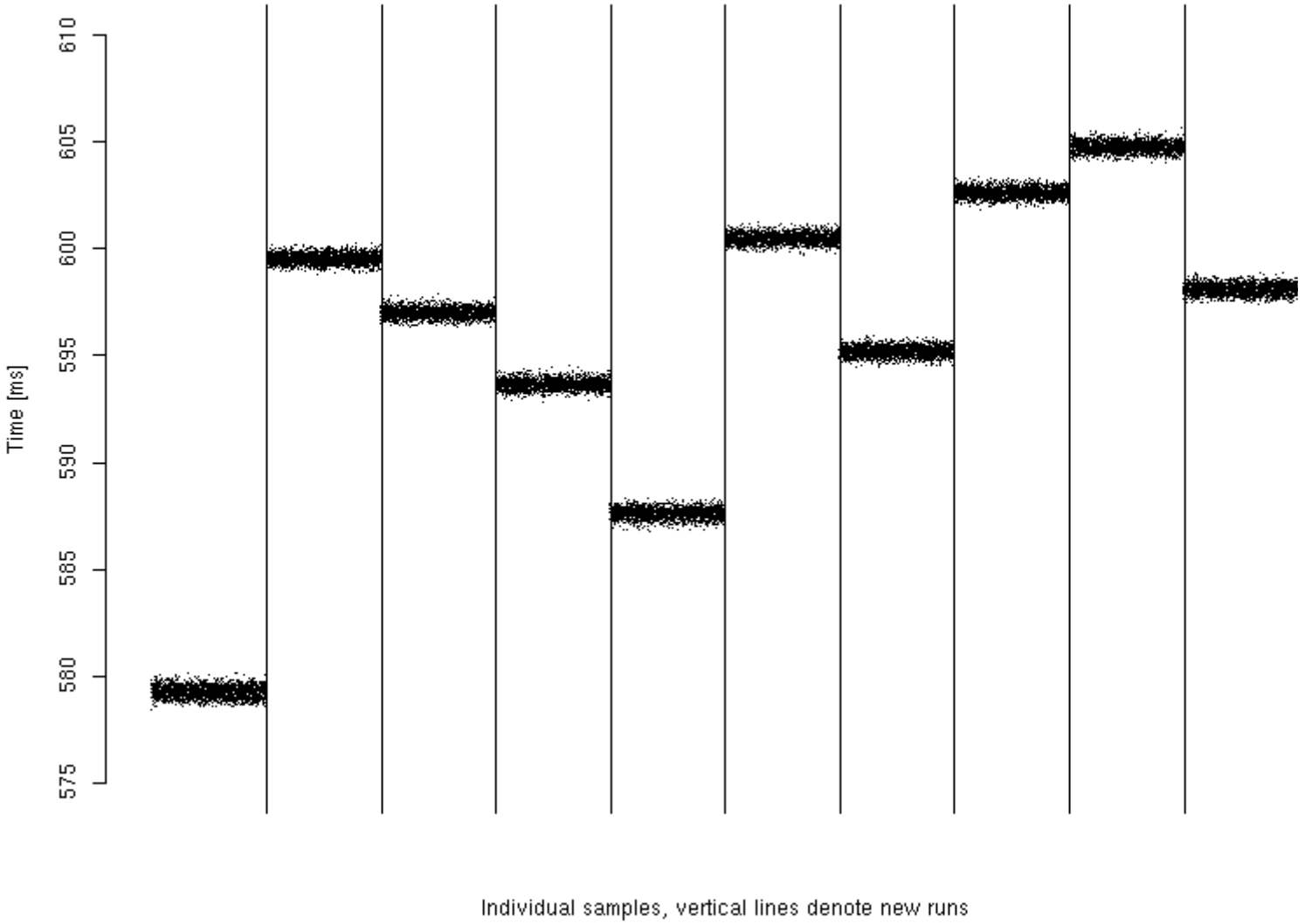
$$1 - \frac{\text{time on new system}}{\text{time on old system}}$$

- Typically we want to compare two systems, estimating relative speed-up
- Typically we focus on steady-state, mean performance
- May have to re-visit as we progress

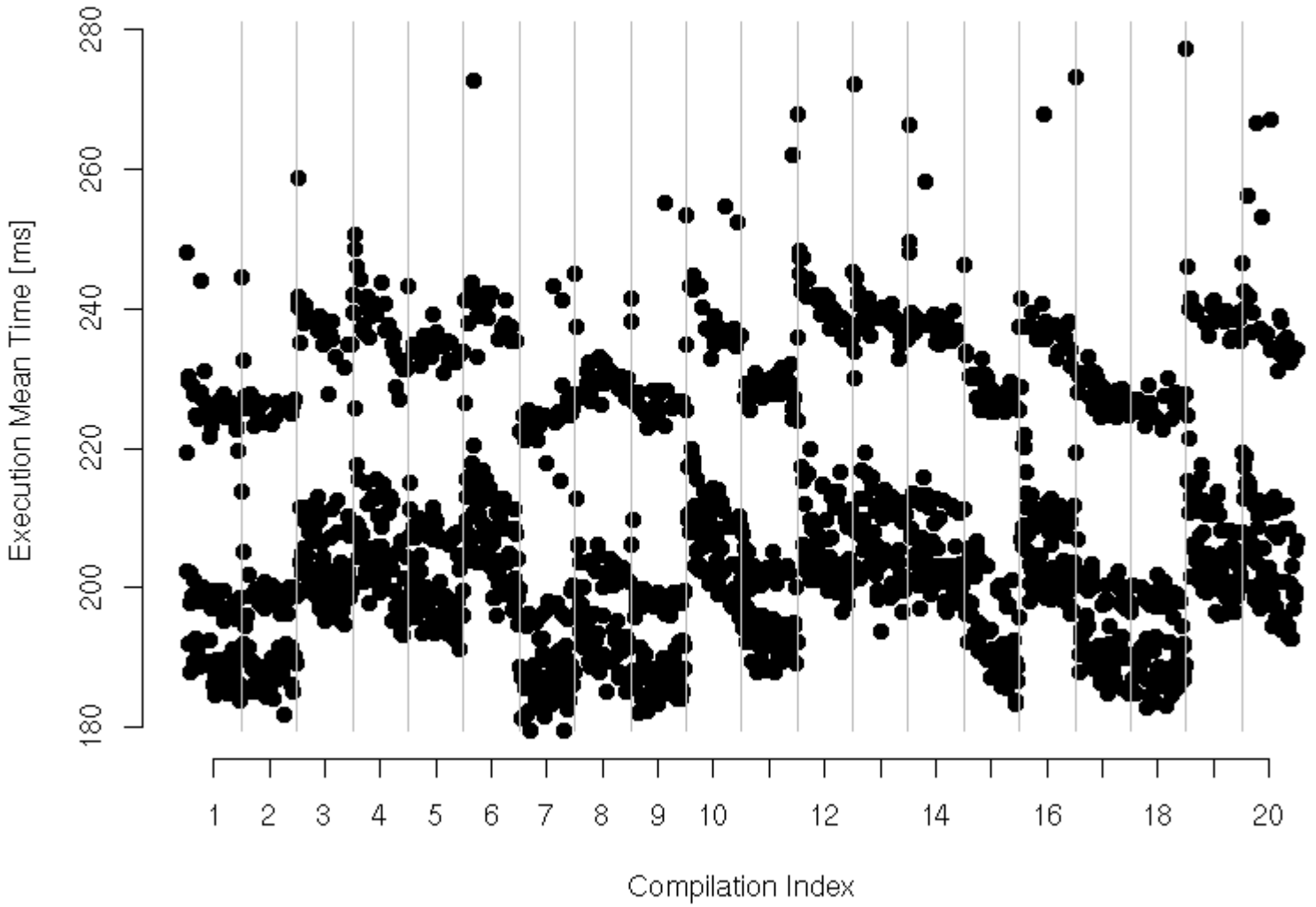
Know the system

- Hardware
  - Frequency scaling, cores, caches, memory, performance counters
- Operating system and system software
  - Scheduler, swapper, compiler, linker, page allocator
- Virtual machine
  - Just-in-time compiler, garbage collector

Know the system (FFT runs)



Know the system (Mono builds)

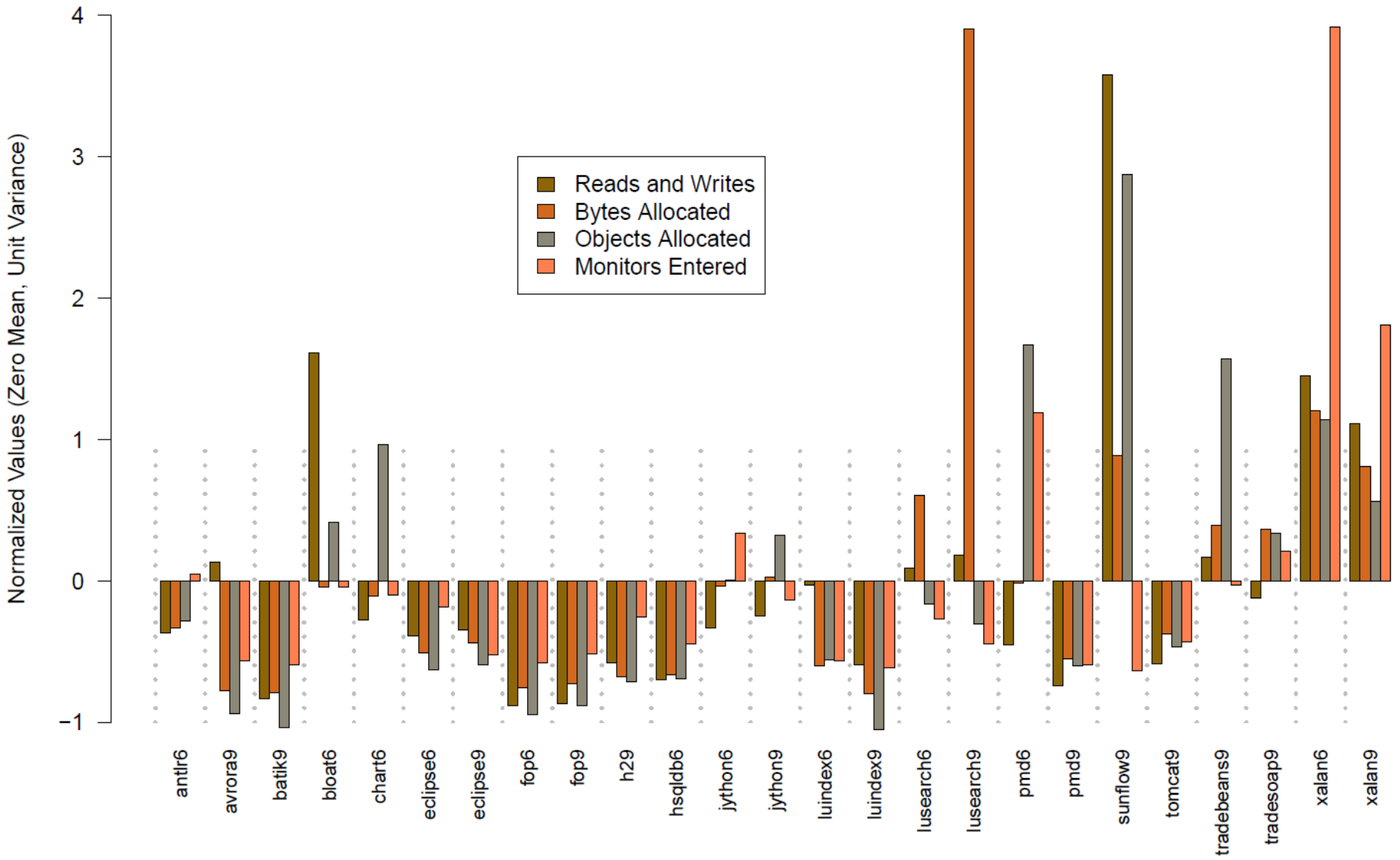


Know real applications, benchmarks

Thread density

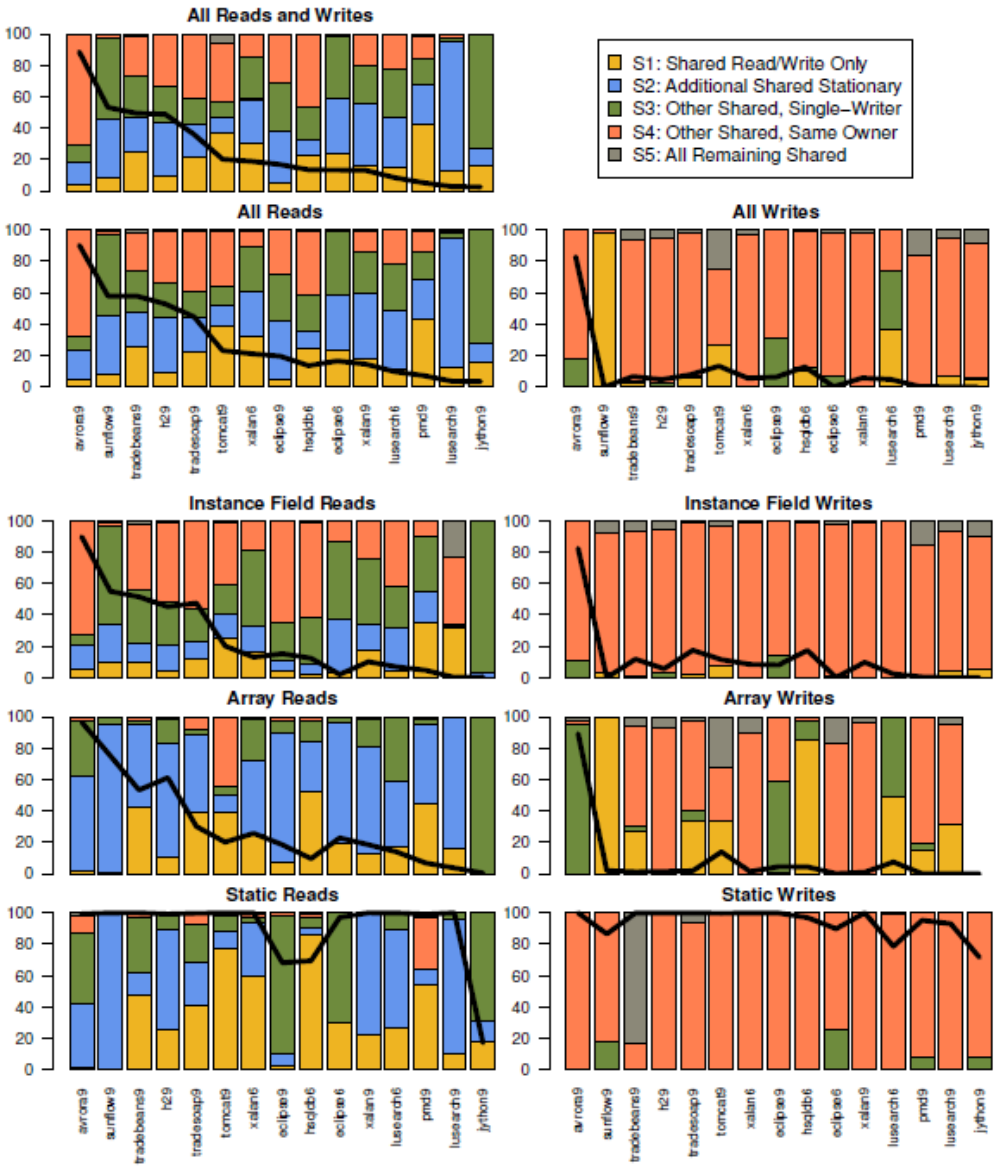
	Any Ops	Alt. Modifs	Spot-shared	Shared	Any Ops	Alt. Modifs	Spot-shared	Shared
	Density				Periodic density			
avrora9	25	22	25	25	22	22	22	22
xalan6	8	8	8	8	8	8	8	8
tomcat9	9	15	12	14	9	8	7	8
tradesoap9	13	44	32	28	7	13	12	11
h29	5	4	4	5	4	4	4	4
tradebeans9	4	4	4	4	4	4	4	4
xalan9	4	4	4	4	4	4	4	4
pmd9	5	5	5	5	4	4	3	3
hsqldb6	201	385	382	393	1	34	15	15
lusearch6	63	59	66	64	44	0	57	59
lusearch9	4	5	5	5	4	0	4	4
sunflow9	9	9	9	9	4	0	4	4
antlr6	1	1	1	2	1	0	0	1
batik9	3	3	5	6	1	0	0	1
bloat6	1	1	1	2	1	0	0	0
chart6	1	2	1	3	1	0	0	0
eclipse6	2	6	5	5	1	0	0	1
eclipse9	28	197	191	72	1	1	0	1
fop6	1	1	1	2	1	0	0	0
fop9	1	1	1	2	1	0	0	1
jython6	1	1	1	2	1	0	0	0
jython9	1	1	2	2	1	0	0	1
luindex6	1	2	2	2	1	0	0	1
luindex9	2	3	3	3	1	0	0	1
pmd6	1	1	1	2	1	0	0	1

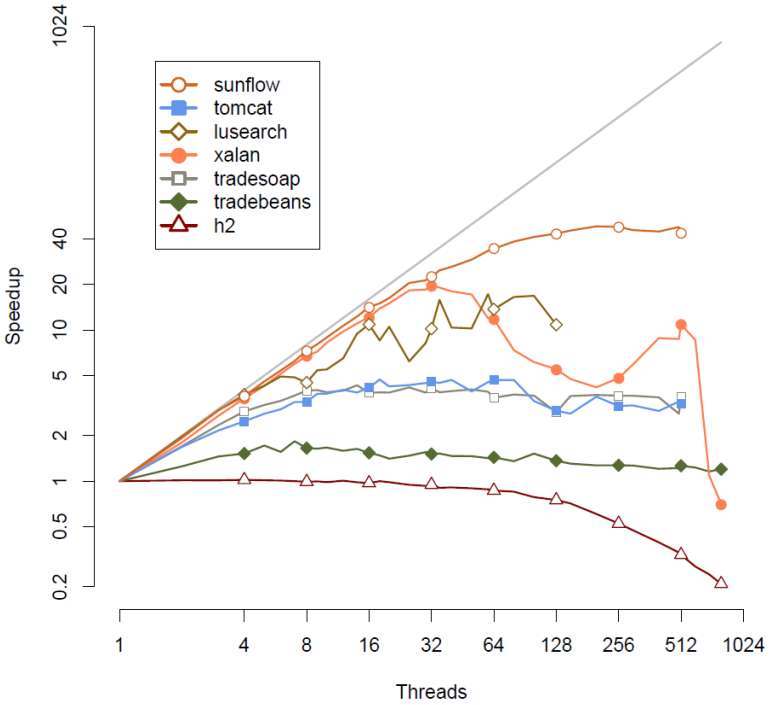
Know real applications, benchmarks



Kalibera, Mole, Jones, Vitek: A black-box approach to understanding concurrency in DaCapo. OOPSLA'12

Know real applications, benchmarks

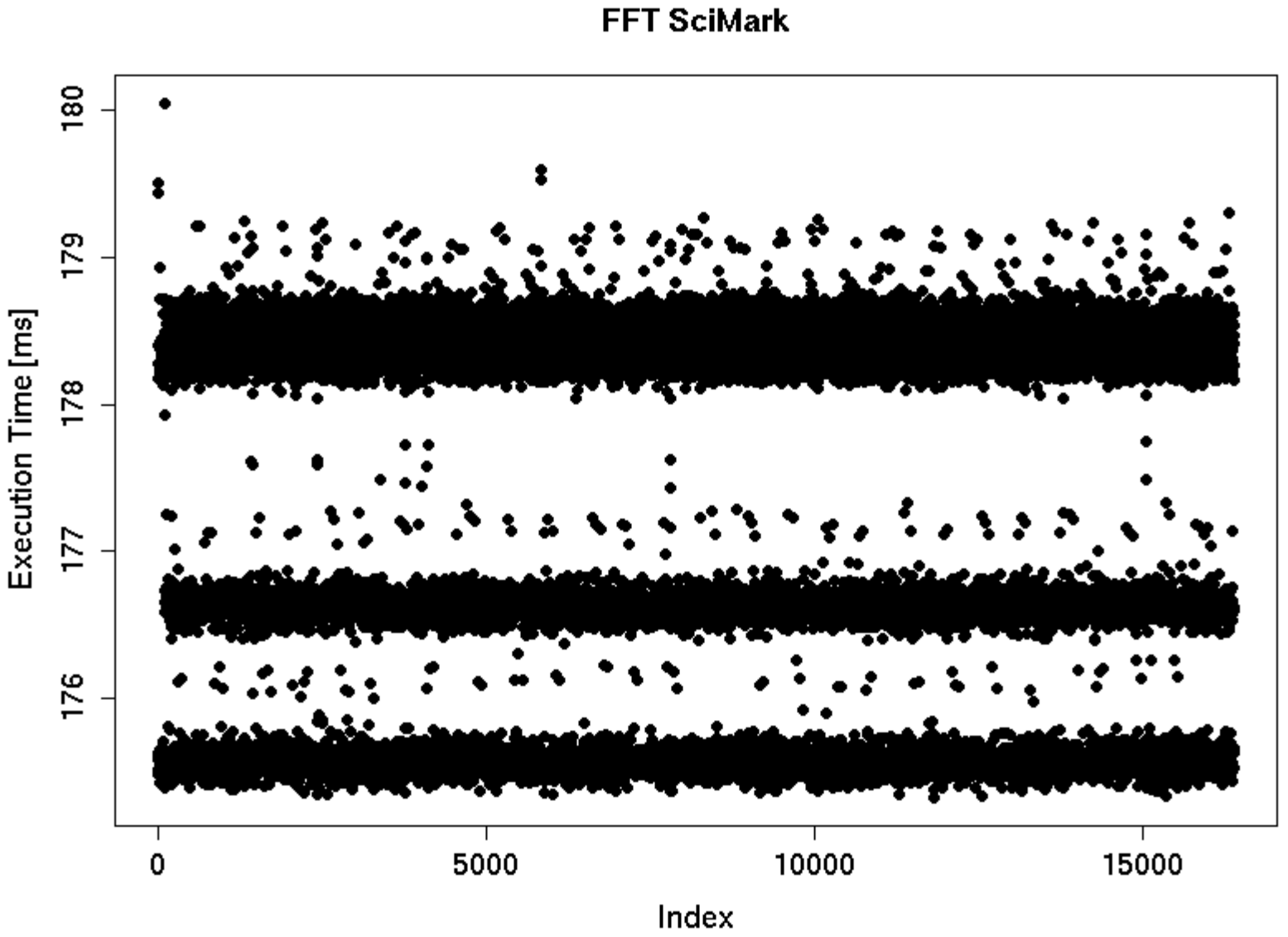




Kalibera, Mole, Jones, Vitek: A black-box approach to understanding concurrency in DaCapo. OOPSLA'12

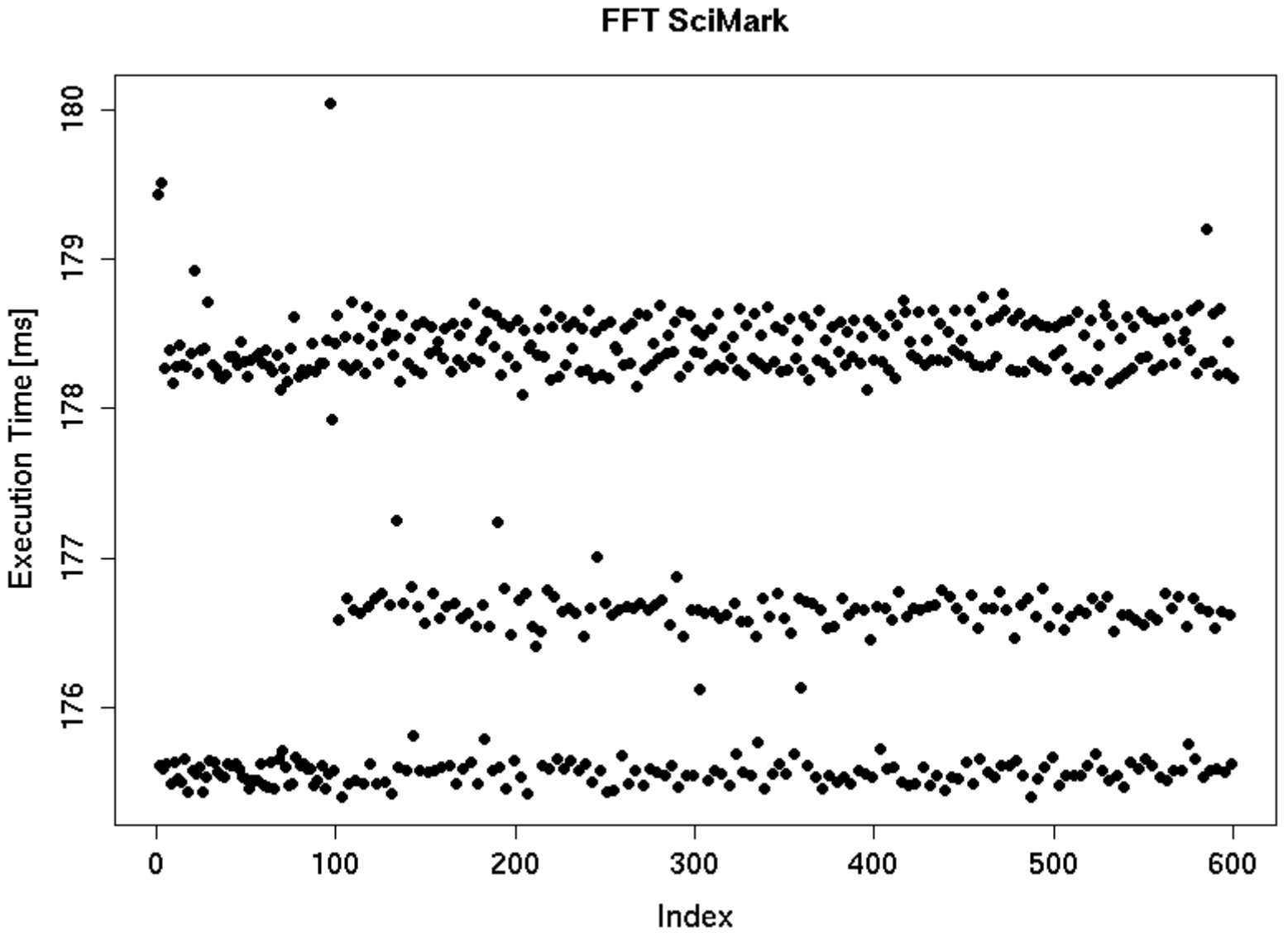
Initial experiments – explore results

FFT SciMark (Mono kernel benchmark)



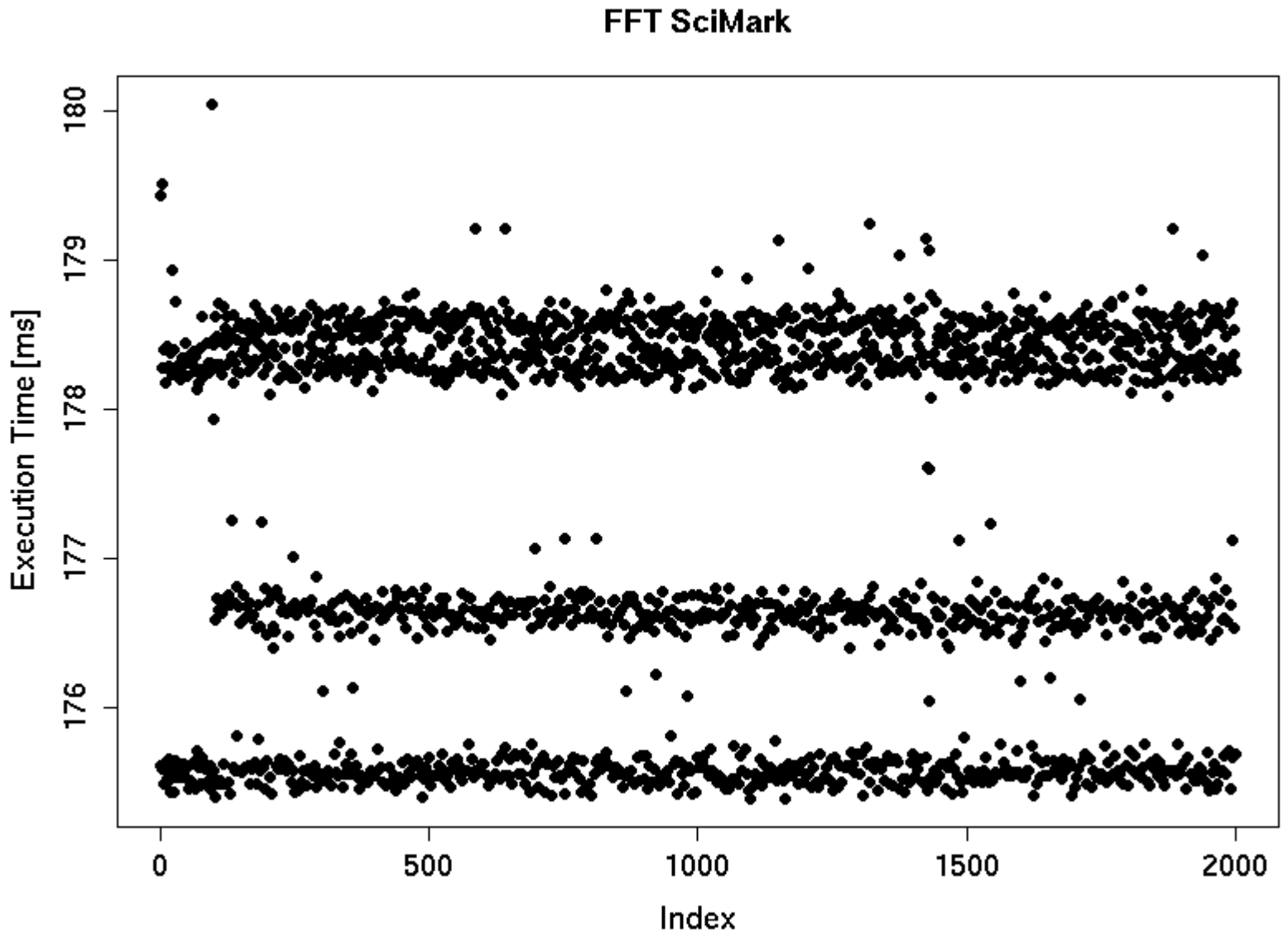
Initial experiments – explore results

FFT SciMark – run sequence plot



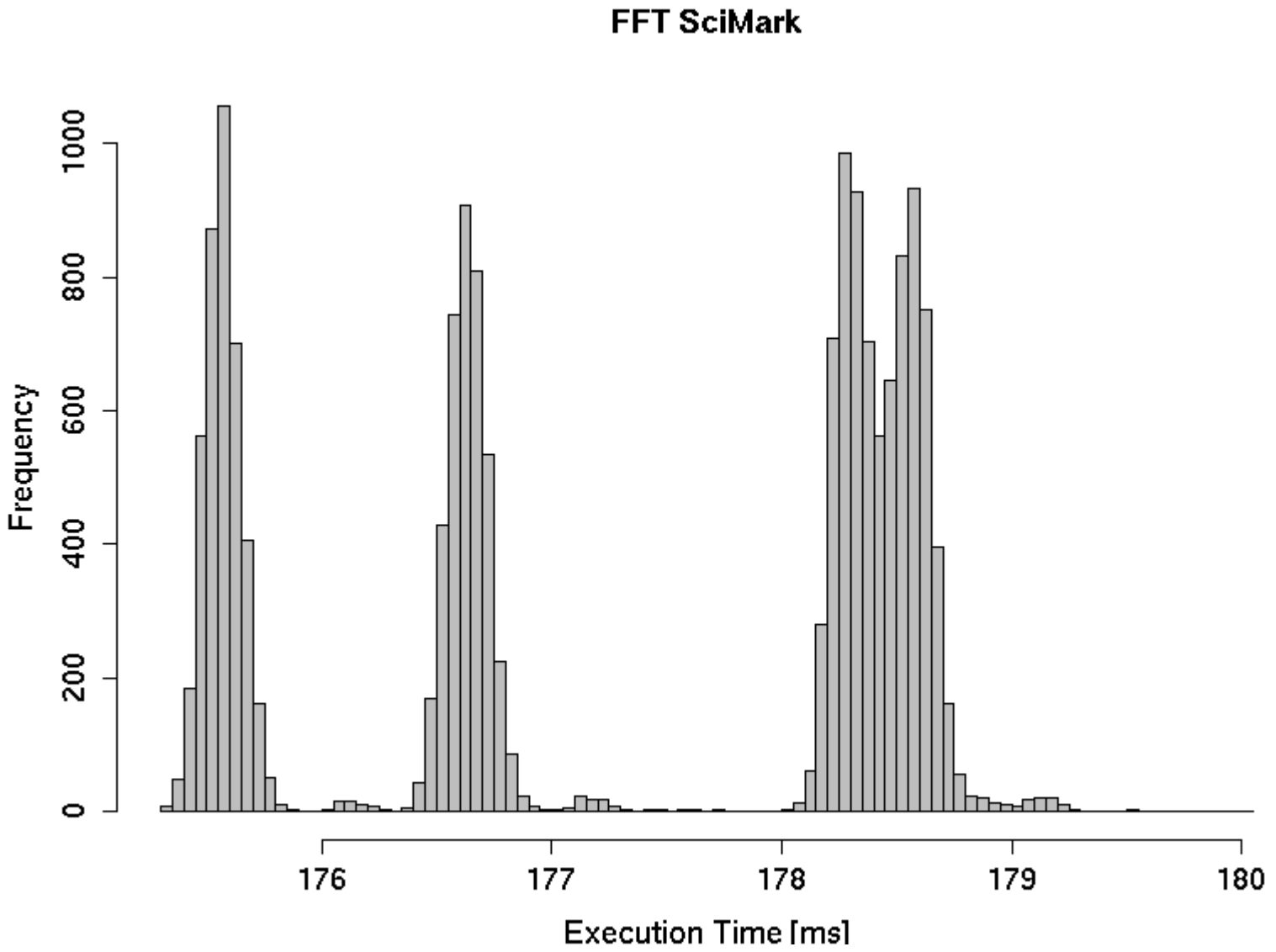
Initial experiments – explore results

FFT SciMark – run sequence plot



Initial experiments – explore results

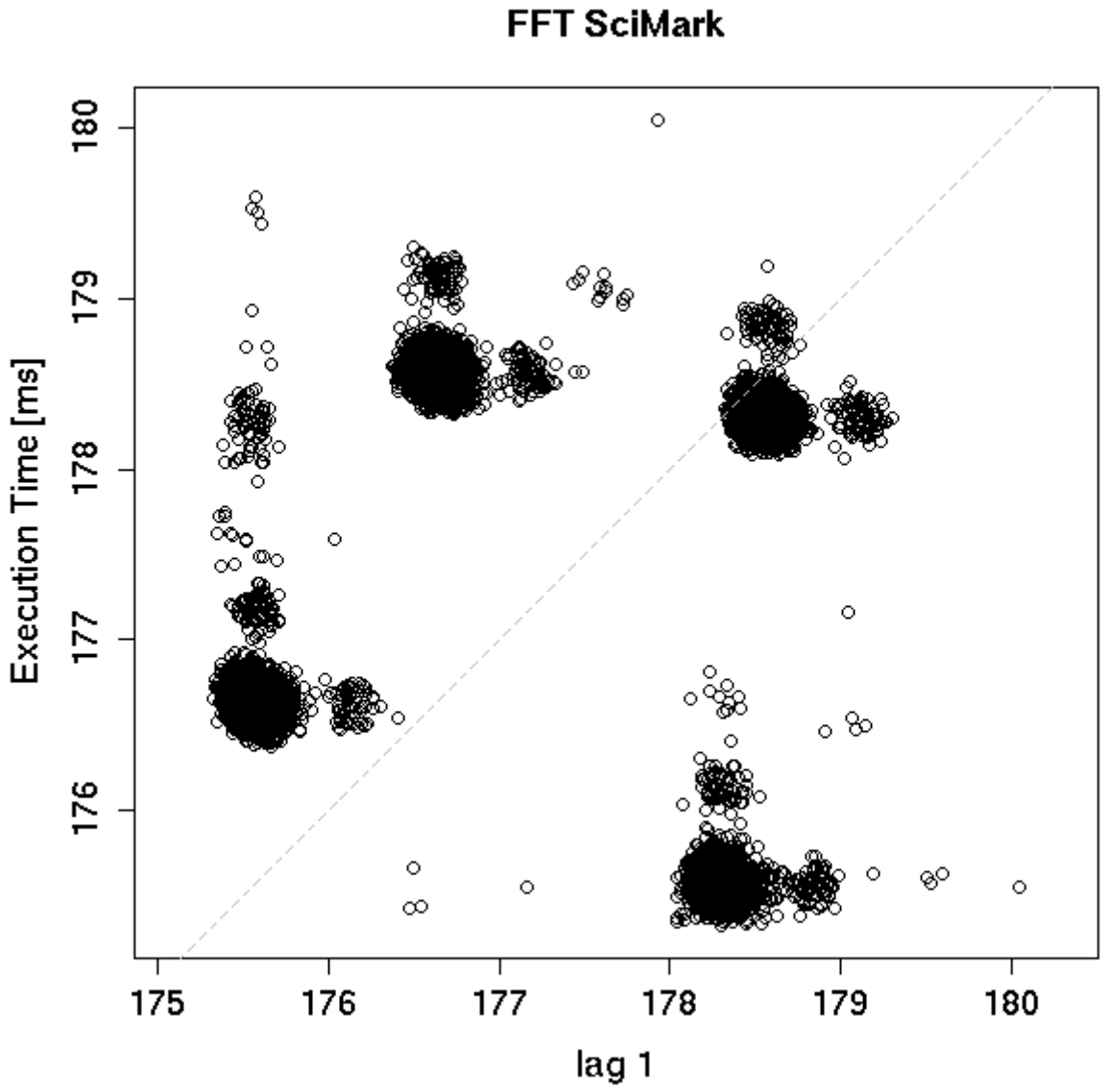
FFT SciMark - histogram



Initial experiments – explore results

FFT SciMark – lag plot

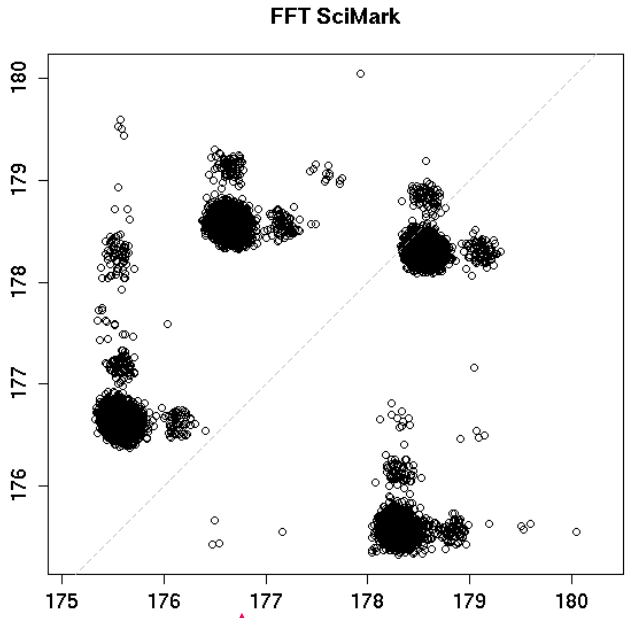
X_i



X_{i+1}

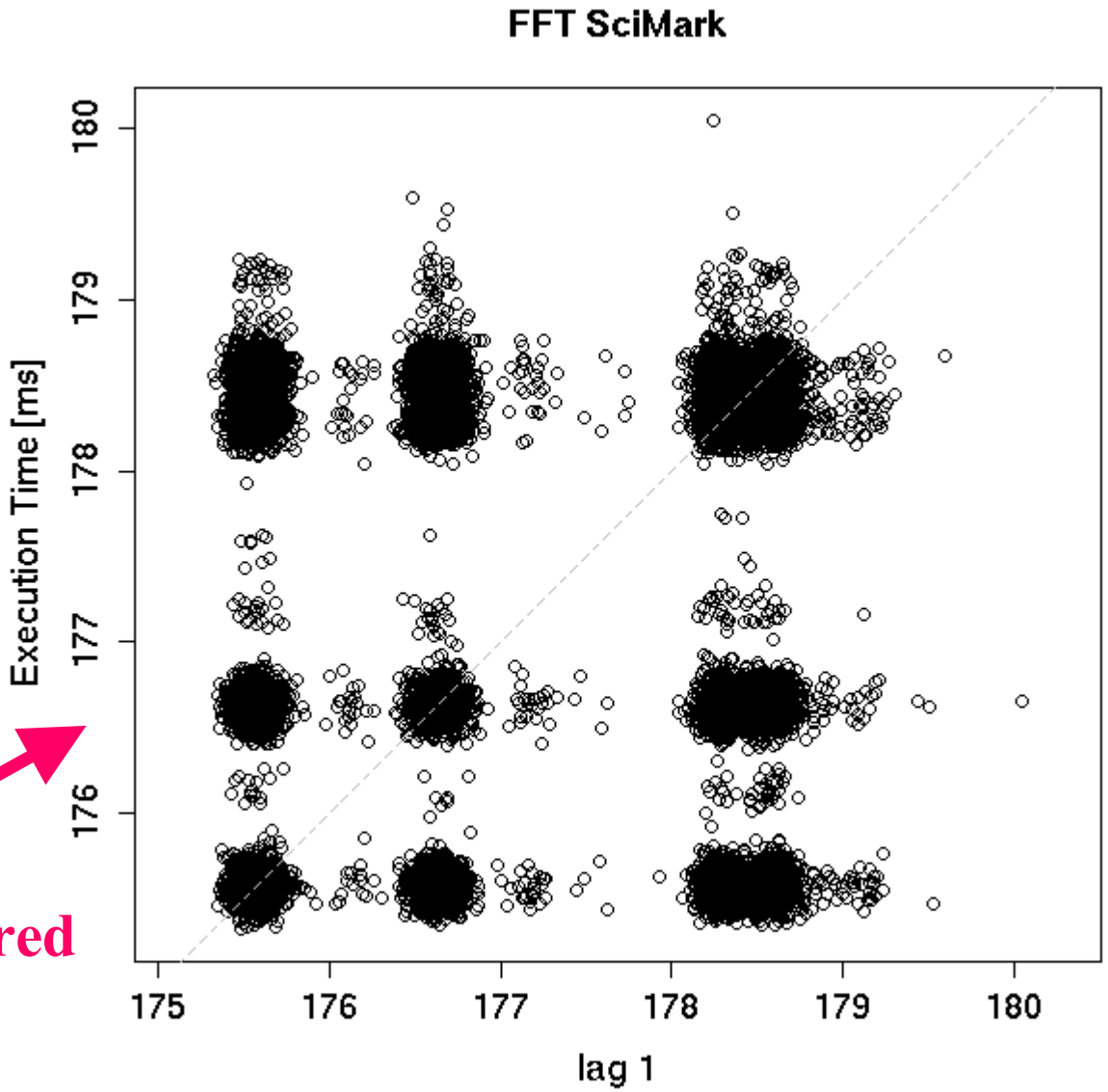
Initial experiments – explore results

FFT SciMark – lag plot



Original

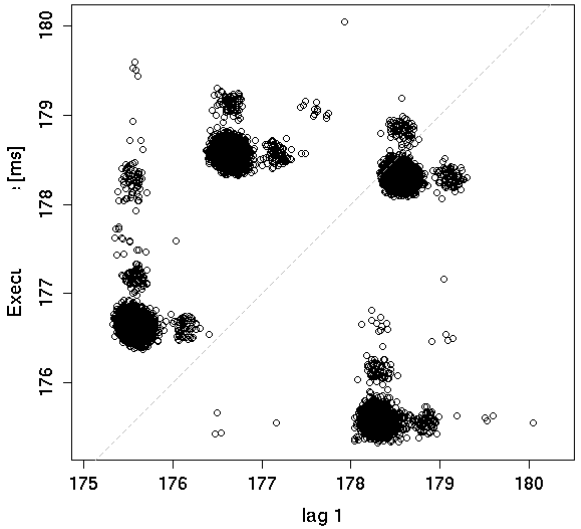
Randomly reordered

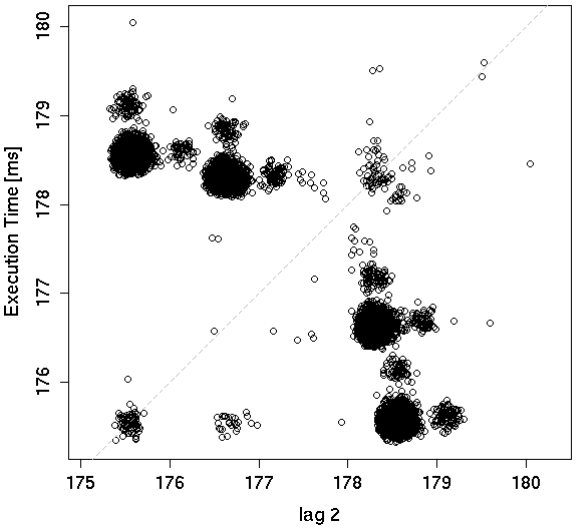


Initial experiments – explore results

FFT SciMark – lag plot (lags 1,2,3,4)

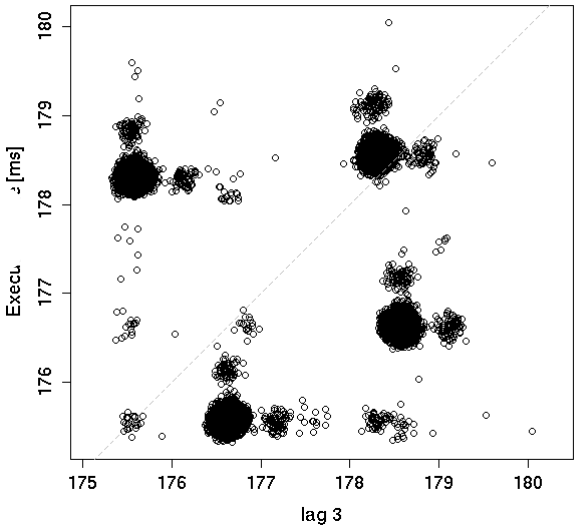
L=1

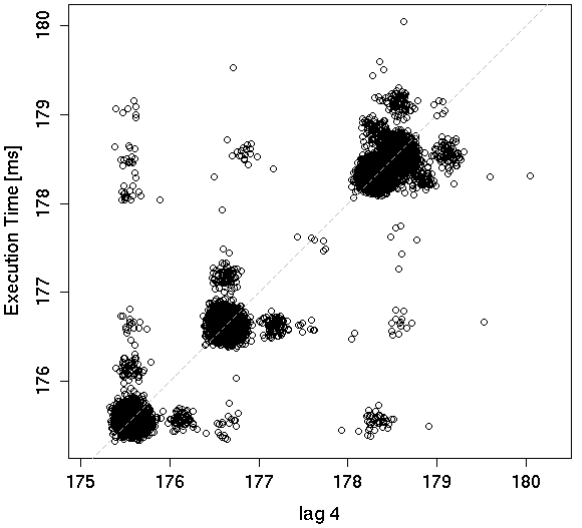




L=2

L=3





L=4

Choose appropriate statistics

- Interpret goal in precise/statistical terms, e.g.
  - Ratio of mean execution times of new / old system
  - Maximum pause time
  - Maximum live size
- Decide on summarization method
  - Confidence interval, which kind
  - How to summarize over benchmarks, platforms
  - Report raw results “we have seen max live size”...
- Validate assumptions

Choose appropriate statistics

Confidence interval for the ratio of means

- Bootstrap method – statistical simulation
- Fieller's method
- Delta method

Kalibera, Jones: Quantifying performance changes with effect size confidence intervals.
http://www.cs.kent.ac.uk/pubs/2012/3233/

Kalibera, Jones: Rigorous benchmarking in reasonable time.
ISMM'13

Choose appropriate statistics

```
# confidence interval for the ratio ex / ey

cfratio <- function(x, y, R = 10000, conf = 0.95) {

  means <- sapply(1:R, function(i) {
    xs <- sample(x, replace = TRUE)
    ys <- sample(y, replace = TRUE)
    mean(xs) / mean(ys)
  })

  alpha <- (1 - conf) / 2
  cfi <- quantile( means, probs=c( alpha, 1 - alpha ) )
  names(cfi) <- NULL

  c(lower = cfi[1], upper = cfi[2], meansRatio = mean(x) / mean(y))
}
```

Percentile bootstrap method.

Plan, dimension experiments

- Identify variables impacting performance
  - Controlled, fixed – record and later report
  - Random (non-determinism) – **plan for repetition**
  - Uncontrolled fixed – turn into either of above
- Typical, known sources of non-determinism include
  - Code layout (linker, JIT)
  - Memory layout (memory allocator in OS/VM/app)
  - Scheduler, GC, system services, network
  - Randomized algorithms, random (unique) names
  - Self-optimization, profiling

Plan repetition

- Must repeat at highest “level” of experiment with non-determinism
  - If compilation is non-deterministic or randomized due to layout dependency, must run benchmarks for multiple builds
- Can also repeat at lower “levels” of experiment
  - This can save experimentation time, if repeating at levels with high contribution to result variance
  - Repetition count to be chosen via initial experiments

Kalibera, Jones: Rigorous benchmarking in reasonable time. ISMM'13

Decide on warm-up iterations

For confidence interval, inference, need independent identically distributed values

- If benchmark iterations are **dependent**, there is no point repeating them (must repeat at higher level)

For steady-state measurements

- Drop initial values that are prone to obvious initialization noise

Dimensioning, warm-up is platform dependent.

Kalibera, Jones: Rigorous benchmarking in reasonable time. ISMM'13

Decide on warm-up iterations

Warm-up in Dacapo6 (platform, VM dependent!)

	Initialized	Independent
bloat	2	4
chart	3	∞
eclipse	5	7
fop	10	180
hsqldb	6	6
jython	3	∞
luindex	13	∞
lusearch	10	85
pmd	7	∞
xalan	6	13

# of iterations to drop

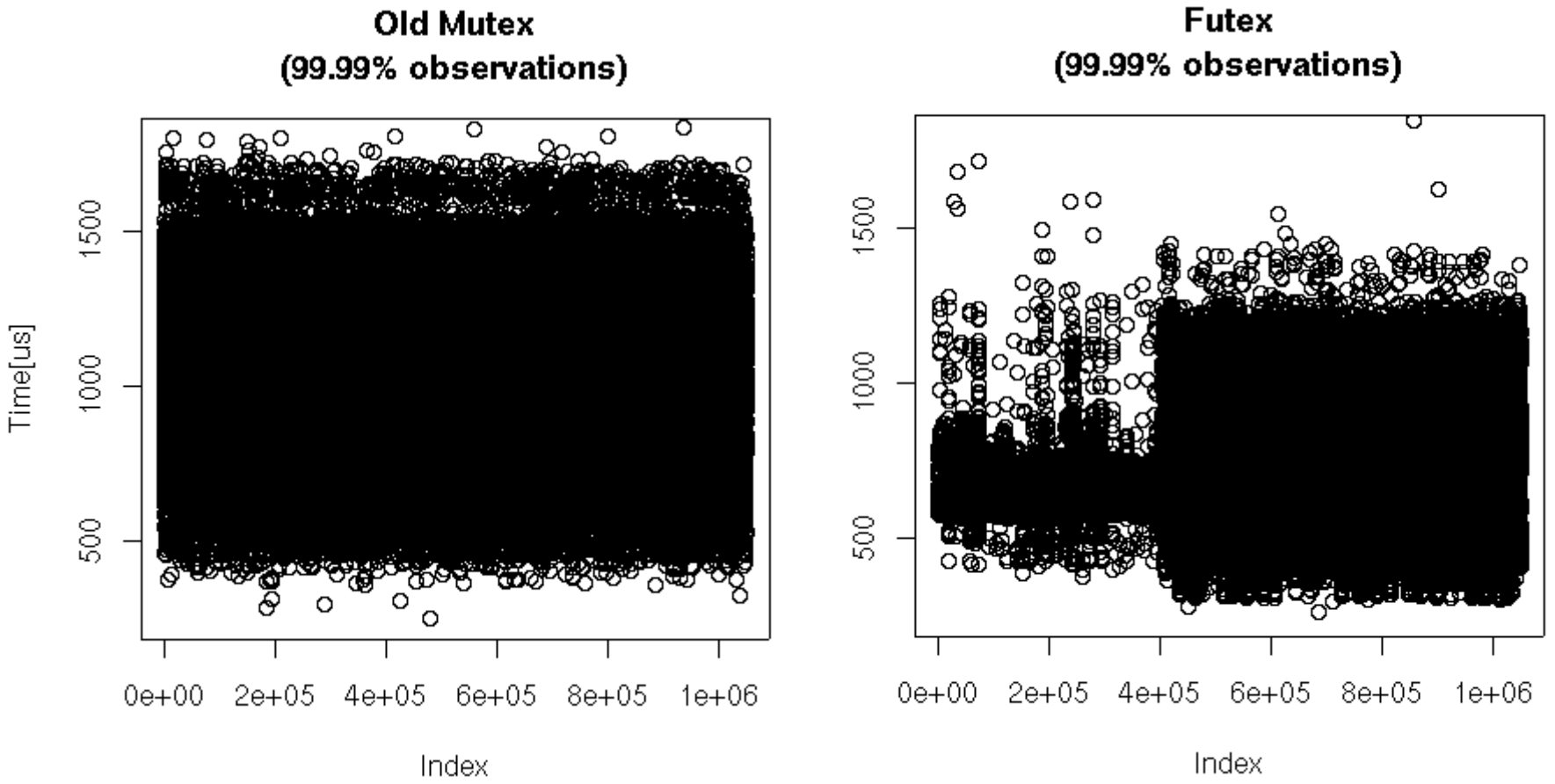
Kalibera, Jones: Rigorous benchmarking in reasonable time. ISMM'13

Warm-up: automated heuristics fail

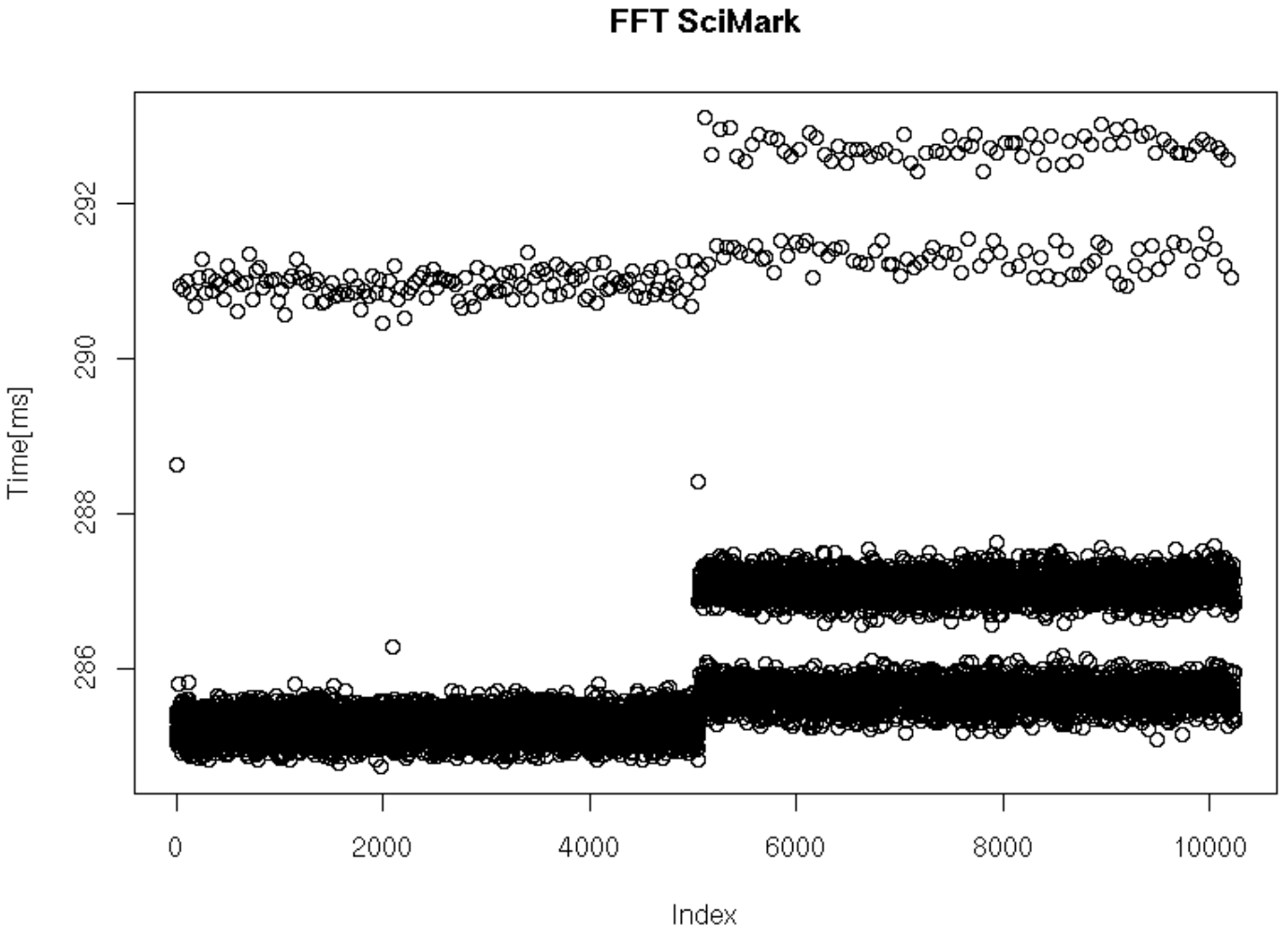
Warm-up in Dacapo6 (platform, VM dependent!)

	Initialized	Independent	Harness	Georges
bloat	2	4	8	∞
chart	3	∞	4	1
eclipse	5	7	7	4
fop	10	180	7	8
hsqldb	6	6	8	15
jython	3	∞	5	2
luindex	13	∞	4	8
lusearch	10	85	7	8
pmd	7	∞	4	1
xalan	6	13	15	139

Kalibera, Jones: Rigorous benchmarking in reasonable time.
ISMM'13



Ping benchmark, Linux, TAO/C++, Dual Pentium 3 SMP



FFT-SciMark benchmark, Linux/Mono, C#, Pentium 4

More tips and advice

Automate experiment execution

Archive scripts, data for reference

Obtain and archive machines config

Keep a log of decisions

Archive graphs used for decisions

Archive code used for evaluation

Use R

All results must be subject to critical (self)-review.
Attempts for automated or mechanical-checklist evaluation produces garbage.

References

Kalibera, Jones: Rigorous benchmarking in reasonable time. ISMM'13.

Kalibera, Jones: Quantifying performance changes with effect size confidence intervals. http://www.cs.kent.ac.uk/pubs/2012/3233/

Kalibera, Mole, Jones, Vitek: A black-box approach to understanding concurrency in DaCapo. OOPSLA'12.

Vitek, Kalibera: R3: repeatability, reproducibility and rigor. ACM Sigplan. Not. 2012

My favorite sources

NIST Engineering Handbook: Engineering Statistics, http://www.itl.nist.gov/div898/handbook/

Jain: The Art of Computer Systems Performance Analysis:
Techniques for Experimental Design, Measurement, Simulation, and Modeling. Wiley'91

Lilja: Measuring Computer Performance: A Practitioner's Guide. Cambridge. U. P.'05

Wasserman: All of Statistics, A Concise Course in Statistical Inference. Springer'04

(and many more as cited in the Uni of Kent technical report)