μOpTime: Statically Reducing the Execution Time of Microbenchmark Suites Using Stability Metrics – Online Appendix

NILS JAPKE, TU Berlin & ECDF, Germany
MARTIN GRAMBOW, TU Berlin & ECDF, Germany
CHRISTOPH LAABER, Simula Research Laboratory, Norway
DAVID BERMBACH, TU Berlin & ECDF, Germany

1 TABLES

1.1 Data Set 1 Commit Hashes

Table 1. Detailed overview of versions and commit hashes in our data set (data set 1). All projects are hosted on GitHub.

Project	Ver. (Tag)	Commit Hash
prometheus/common	`v0.35.0`	26d4974
	`v0.36.0`	c5e1b60
	`v0.37.0`	49b3603
	`v0.38.0`	a33c32f
	`v0.39.0`	296ec92
pelletier/go-toml	`v2.0.1`	0a422e3
	`v2.0.2`	2166282
	`v2.0.3`	7baa23f
	`v2.0.4`	28f1efc
	`v2.0.5`	9428417
uber-go/zap	`v1.19.1`	eaeb0fc
	`v1.20.0`	ad0b02d
	`v1.21.0`	6f34060
	`v1.22.0`	4b03bc5
	`v1.23.0`	1ae5819

1.2 RQ4

Authors' addresses: Nils Japke, TU Berlin & ECDF, Scalable Software Systems Research Group, Berlin, Germany, nj@3s.tu-berlin.de; Martin Grambow, TU Berlin & ECDF, Scalable Software Systems Research Group, Berlin, Germany, mg@mcc.tu-berlin.de; Christoph Laaber, Simula Research Laboratory, Oslo, Norway, laaber@simula.no; David Bermbach, TU Berlin & ECDF, Scalable Software Systems Research Group, Berlin, Germany, db@3s.tu-berlin.de.

Table 2. Detected performance changes by the full and minimimal execution configurations in Go projects using the minimum baseline. Different versions of the studied software projects are given as Git tags.

Project	v_1	v_2	# MB	# Perf. Changes		FPR	FNR
				full	min		
prometheus/common	v0.35.0	v0.36.0	31	0	0	0%	0%
	v0.36.0	v0.37.0	31	0	0	0%	0%
	v0.37.0	v0.38.0	31	28	27	0%	3.57%
	v0.38.0	v0.39.0	31	0	0	0%	0%
pelletier/go-toml	v2.0.1	v2.0.2	35	1	1	0%	0%
	v2.0.2	v2.0.3	35	12	11	0%	8.33%
	v2.0.3	v2.0.4	35	1	1	0%	0%
	v2.0.4	v2.0.5	35	12	12	0%	0%
uber-go/zap	v1.19.1	v1.20.0	21	1	3	10%	0%
	v1.20.0	v1.21.0	21	0	0	0%	0%
	v1.21.0	v1.22.0	21	0	0	0%	0%
	v1.22.0	v1.23.0	21	2	1	0%	50%

Table 3. Detected performance changes by the full and minimimal execution configurations in Go projects using the random baseline. Different versions of the studied software projects are given as Git tags.

Project	v_1	v_2	# MB	# Perf. Changes		FPR	FNR
				full	min		
prometheus/common	v0.35.0	v0.36.0	31	0	0	0%	0%
	v0.36.0	v0.37.0	31	0	0	0%	0%
	v0.37.0	v0.38.0	31	28	27	0%	3.57%
	v0.38.0	v0.39.0	31	0	0	0%	0%
pelletier/go-toml	v2.0.1	v2.0.2	35	1	1	0%	0%
	v2.0.2	v2.0.3	35	12	12	0%	0%
	v2.0.3	v2.0.4	35	1	1	0%	0%
	v2.0.4	v2.0.5	35	12	12	0%	0%
uber-go/zap	v1.19.1	v1.20.0	21	1	2	5%	0%
	v1.20.0	v1.21.0	21	0	0	0%	0%
	v1.21.0	v1.22.0	21	0	0	0%	0%
	v1.22.0	v1.23.0	21	2	2	0%	0%

Table 4. Detected performance changes by the full and minimimal execution configurations in Go projects using µOpTime. Different versions of the studied software projects are given as Git tags.

Project	v_1	v_2	# MB	# Perf. Changes		FPR	FNR
				full	min		
prometheus/common	`v0.35.0`	`v0.36.0`	31	0	0	0%	0%
	`v0.36.0`	`v0.37.0`	31	0	0	0%	0%
	`v0.37.0`	`v0.38.0`	31	28	28	0%	0%
	`v0.38.0`	`v0.39.0`	31	0	0	0%	0%
pelletier/go-toml	`v2.0.1`	`v2.0.2`	35	1	1	0%	0%
	`v2.0.2`	`v2.0.3`	35	12	12	4.35%	8.33%
	`v2.0.3`	`v2.0.4`	35	1	1	0%	0%
	`v2.0.4`	`v2.0.5`	35	12	12	4.35%	8.33%
uber-go/zap	`v1.19.1`	`v1.20.0`	21	1	2	5%	0%
	`v1.20.0`	`v1.21.0`	21	0	0	0%	0%
	`v1.21.0`	`v1.22.0`	21	0	0	0%	0%
	`v1.22.0`	`v1.23.0`	21	2	2	0%	0%

Table 5. Detected performance changes by the full and minimimal execution configurations in Java projects using the minimum baseline. Different versions of the studied software projects are given as in release versions. Only versions with detected changes (either full or minimal configuration) are listed.

Project	v_1	v_2	# MB	# Perf. Changes		FPR	FNR
				full	min		
raphw/byte-buddy	v1.0.0	v1.0.3	20	0	1	5%	0%
	v1.0.3	v1.1.0	20	3	3	17.65%	100%
	v1.1.0	v1.1.1	20	0	1	5%	0%
	v1.1.1	v1.2.0	20	0	1	5%	0%
	v1.2.0	v1.2.3	20	0	3	15%	0%
	v1.2.3	v1.3.0	20	4	6	18.75%	25%
	v1.3.0	v1.3.10	20	0	1	5%	0%
	v1.3.20	v1.4.0	20	1	1	5.26%	100%
	v1.4.0	v1.4.11	20	0	1	5%	0%
	v1.4.11	v1.4.22	20	0	5	25%	0%
	v1.4.22	v1.4.33	20	0	2	10%	0%
	v1.4.33	v1.5.0	20	0	4	20%	0%
	v1.5.7	v1.5.13	20	0	1	5%	0%
	v1.6.0	v1.6.7	18	2	1	6.25%	100%
	v1.6.7	v1.6.14	18	1	1	5.88%	100%
	v1.6.14	v1.7.0	18	0	1	5.56%	0%
	v1.7.0	v1.7.6	18	0	2	11.11%	0%
	v1.7.6	v1.7.11	18	0	2	11.11%	0%
	v1.7.11	v1.8.0	18	0	2	11.11%	0%
	v1.8.0	v1.8.11	18	0	2	11.11%	0%
	v1.8.11	v1.8.22	18	0	1	5.56%	0%
	v1.8.22	v1.9.0	18	0	2	11.11%	0%
	v1.9.0	v1.9.8	18	0	2	11.11%	0%
	v1.9.8	v1.9.16	18	0	1	5.56%	0%
	v1.10.3	v1.10.7	18	0	1	5.56%	0%
jenetics/jenetics	v3.0.0	v3.0.1	39	1	3	7.89%	100%
	v3.0.1	v3.1.0	39	1	6	13.16%	0%
	v3.1.0	v3.2.0	39	0	6	15.16%	0%
	v3.2.0	v3.3.0	34	1	2	6.06%	100%
	v3.3.0	v3.4.0	34	2	5	15.63%	100%
	v3.4.0	v3.5.0	34	0	1	2.94%	0%
	v3.5.0	v3.5.1	34	1	4	9.09%	0%
	v3.5.1	v3.6.0	34	0	1	2.94%	0%
	v3.6.0	v3.7.0	34	0	2	5.88%	0%
	v3.7.0	v3.8.0	34	1	3	9.09%	100%
	v3.8.0	v3.9.0	34	1	6	15.15%	0%
JetBrains/xodus	v1.0.5	v1.0.6	47	3	5	6.82%	33.33%
	v1.0.6	v1.1.0	47	2	3	4.44%	50%
	v1.1.0	v1.2.0	47	1	4	8.70%	100%
	v1.2.0	v1.2.1	47	4	6	11.63%	75%
	v1.2.1	v1.2.2	47	1	4	8.70%	100%
	v1.2.2	v1.2.3	47	1	8	17.39%	100%
	v1.2.3	v1.3.0	47	0	5	10.64%	0%
	v1.3.0	v1.3.91	47	4	4	9.30%	100%
	v1.3.9	v1.3.232	47	5	9	14.29%	40%
openzipkin/zipkin	v2.0.0	v2.1.0	52	1	5	7.84%	0%
	v2.1.0	v2.2.0	52	3	10	14.29%	0%
	v2.2.0	v2.3.0	52	2	9	16%	50%
	v2.3.0	v2.4.0	52	5	9	10.64%	20%
	v2.4.0	v2.5.0	52	5	10	14.89%	40%
	v2.5.0	v2.6.0	52	1	6	11.76%	100%
	v2.6.0	v2.7.0	52	3	4	6.12%	66.67%
	v2.7.0	v2.8.0	49	22	17	14.81%	40.91%
	v2.8.0	v2.9.0	49	7	8	7.14%	28.57%

Table 6. Detected performance changes by the full and minimimal execution configurations in Java projects using the random baseline. Different versions of the studied software projects are given as in release versions. Only versions with detected changes (either full or minimal configuration) are listed.

Project	v_1	v_2	# MB	# Perf. Changes		FPR	FNR
				full	min		
raphw/byte-buddy	`v1.0.0`	`v1.0.3`	20	0	1	5%	0%
	`v1.0.3`	`v1.1.0`	20	3	1	0%	66.67%
	`v1.2.3`	`v1.3.0`	20	4	2	0%	50%
	`v1.3.20`	`v1.4.0`	20	1	0	0%	100%
	`v1.4.11`	`v1.4.22`	20	0	2	10%	0%
	`v1.6.0`	`v1.6.7`	18	2	0	0%	100%
	`v1.6.7`	`v1.6.14`	18	1	0	%	100%
	`v1.8.11`	`v1.8.22`	18	0	1	5.56%	0%
jenetics/jenetics	`v3.0.0`	`v3.0.1`	39	1	2	5.26%	100%
	`v3.0.1`	`v3.1.0`	39	1	2	2.63%	0%
	`v3.1.0`	`v3.2.0`	39	0	2	5.13%	0%
	`v3.2.0`	`v3.3.0`	34	1	0	0%	100%
	`v3.3.0`	`v3.4.0`	34	2	3	9.38%	100%
	`v3.4.0`	`v3.5.0`	34	0	1	2.94%	0%
	`v3.5.0`	`v3.5.1`	34	1	2	3.03%	0%
	`v3.6.0`	`v3.7.0`	34	0	1	2.94%	0%
	`v3.7.0`	`v3.8.0`	34	1	1	3.03%	100%
	`v3.8.0`	`v3.9.0`	34	1	1	0%	0%
JetBrains/xodus	`v1.0.5`	`v1.0.6`	47	3	4	2.27%	0%
	`v1.0.6`	`v1.1.0`	47	2	3	4.44%	50%
	`v1.1.0`	`v1.2.0`	47	1	0	0%	100%
	`v1.2.0`	`v1.2.1`	47	4	5	6.98%	50%
	`v1.2.1`	`v1.2.2`	47	0	4	8.51%	0%
	`v1.2.2`	`v1.2.3`	47	1	2	2.17%	0%
	`v1.2.3`	`v1.3.0`	47	0	6	12.77%	0%
	`v1.3.0`	`v1.3.91`	47	4	2	0%	50%
	`v1.3.9`	`v1.3.232`	47	5	5	7.14%	60%
openzipkin/zipkin	`v2.0.0`	`v2.1.0`	52	1	2	3.92%	100%
	`v2.1.0`	`v2.2.0`	52	3	3	2.04%	33.33%
	`v2.2.0`	`v2.3.0`	52	2	2	2%	50%
	`v2.3.0`	`v2.4.0`	52	5	7	4.26%	0%
	`v2.4.0`	`v2.5.0`	52	5	7	10.64%	60%
	`v2.5.0`	`v2.6.0`	52	1	3	5.88%	100%
	`v2.6.0`	`v2.7.0`	52	3	0	0%	100%
	`v2.7.0`	`v2.8.0`	49	22	18	11.11%	31.82%
	`v2.8.0`	`v2.9.0`	49	7	5	0%	28.57%

Table 7. Detected performance changes by the full and minimimal execution configurations in Java projects using µOpTime. Different versions of the studied software projects are given as in release versions. Only versions with detected changes (either full or minimal configuration) are listed.

Project	v_1	v_2	# MB	# Perf. Changes		FPR	FNR
				full	min		
raphw/byte-buddy	v1.0.0	v1.0.3	20	0	1	5%	0%
	v1.0.3	v1.1.0	20	3	1	0%	66.67%
	v1.2.0	v1.2.3	20	0	2	10%	0%
	v1.2.3	v1.3.0	20	4	3	0%	25%
	v1.3.10	v1.3.20	20	0	1	5%	0%
	v1.3.20	v1.4.0	20	1	1	0%	0%
	v1.6.0	v1.6.7	18	2	1	0%	50%
	v1.6.7	v1.6.14	18	1	2	5.88%	0%
	v1.7.11	v1.8.0	18	0	1	5.56%	0%
	v1.9.0	v1.9.8	18	0	1	5.56%	0%
	v1.9.8	v1.9.16	18	1	1	0%	0%
jenetics/jenetics	v3.0.0	v3.0.1	39	1	1	0%	0%
	v3.0.1	v3.1.0	39	1	1	0%	0%
	v3.1.0	v3.2.0	39	0	1	2.56%	0%
	v3.2.0	v3.3.0	34	1	2	3.03%	0%
	v3.3.0	v3.4.0	34	2	2	0%	0%
	v3.5.0	v3.5.1	34	1	1	0%	0%
	v3.7.0	v3.8.0	34	1	1	0%	0%
	v3.8.0	v3.9.0	34	1	2	3.03%	0%
JetBrains/xodus	v1.0.5	v1.0.6	47	3	3	0%	0%
	v1.0.6	v1.1.0	47	2	2	0%	0%
	v1.1.0	v1.2.0	47	1	1	0%	0%
	v1.2.0	v1.2.1	47	4	5	2.33%	0%
	v1.2.1	v1.2.2	47	1	1	0%	0%
	v1.2.2	v1.2.3	47	1	3	4.35%	0%
	v1.2.3	v1.3.0	47	0	1	2.13%	0%
	v1.3.0	v1.3.91	47	4	5	2.33%	0%
	v1.3.9	v1.3.232	47	4	7	6.98%	0%
openzipkin/zipkin	v2.0.0	v2.1.0	52	1	2	1.96%	0%
	v2.1.0	v2.2.0	52	3	4	2.04%	0%
	v2.2.0	v2.3.0	52	2	3	2%	0%
	v2.3.0	v2.4.0	52	5	8	6.38%	0%
	v2.4.0	v2.5.0	52	5	7	4.26%	0%
	v2.5.0	v2.6.0	52	1	2	3.92%	100%
	v2.6.0	v2.7.0	52	3	3	0%	0%
	v2.7.0	v2.8.0	49	22	22	3.70%	4.55%
	v2.8.0	v2.9.0	49	7	6	0%	14.29%