

Benchmark Results

all times in μ s Benchmark:	n	C++98 gcc -O2		C++98 no opt		FP 3.2.2 -O4		FP 3.2.2 no opt		Node.js 12.16		LuaJIT 2.0 Lua 5.1		OBXMC 21-08 Mono5		ObxIDE 0.9.38 C gen, with GC		ObxIDE 0.9.38 C gen, with GC	
		average	factor	average	factor	average	factor	average	factor	average	factor	average	factor	average	factor	average	factor	average	factor
...																			
Bounce	1500/1	51	0.20	111	0.4	74	0.3	146	0.6	119	0.5	249	1.0	126	0.5	68	0.3	160	0.6
List	1500/1	78	0.12	496	0.7	165	0.2	308	0.5	208	0.3	676	1.0	222	0.3	90	0.1	200	0.3
Mandelbrot	500/1	1	0.50	1	0.5	1	0.5	1	0.5	12	6.0	2	1.0	11	5.5	1	0.5	1	0.5
NBody	250000/1	1	0.13	4	0.5	2	0.3	3	0.4	3	0.4	8	1.0	9	1.1	3	0.4	4	0.5
Permute	1000/1	98	0.30	239	0.7	133	0.4	196	0.6	168	0.5	328	1.0	272	0.8	132	0.4	294	0.9
Queens	1000/1	96	0.32	173	0.6	118	0.4	209	0.7	231	0.8	297	1.0	228	0.8	148	0.5	334	1.1
Sieve	3000/1	31	0.26	89	0.7	34	0.3	81	0.7	103	0.9	119	1.0	99	0.8	31	0.3	144	1.2
Storage	1000/1	756	0.34	1'019	0.5	2'514	1.1	2'637	1.2	310	0.1	2'202	1.0	384	0.2	533	0.2	709	0.3
Towers	600/1	159	0.53	531	1.8	271	0.9	592	2.0	307	1.0	299	1.0	482	1.6	260	0.9	489	1.6
sum of averages:		1'271		2'663		3'312		4'173		1'461		4'180		1'833		1'266		2'335	
geomean of factors:			0.27		0.65		0.42		0.69		0.63		1.0		0.81		0.35		0.68
1/geomean:			3.75		1.53		2.37		1.46		1.58		1.00		1.23		2.87		1.47

Benchmarks used from <https://github.com/smarr/are-we-fast-yet> commit 770c664 3.4.2020

and <https://github.com/rochus-keller/Are-we-fast-yet>

Testmachine: HP EliteBook 2530p, Intel Core Duo L9400 1.86GHz, 4GB RAM, Linux i386

All binaries compiled with GCC 4.8.2 and FPC 3.2.2

LuaJIT params, deviations from default values:

maxtrace 100000
maxrecord 40000
maxside 100
maxsnap 1000
sizemcode 64
maxmcode 5120

Mono 5.20.1.34

gcc 4.8.2 -O2 gcc 4.8.2 no opt
Boem GC 7.2d Boem GC 7.2d

NOTE that this report only includes the Are-we-fast-yet microbenchmarks so far; the macrobenchmarks are work in progress.