Benchmark Results

all times in µs		**Crystal 1.0 --release**		**Node.js 12.16**		**LuaJIT 2.0 Lua 5.1**		**OBXLJ 2021-07-11**		**OBXMC 21-09-12 Mono3**		**OBXMC 21-08-28 Mono5**		**ObxIDE 0.9.25 C gen, no GC**			**ObxIDE 0.9.28 C gen, with GC**	
Benchmark:	*n*	*average*	*factor*	*average*	*factor*	*average*	*factor*	*average*	*factor*	*average*	*factor*	*average*	*factor*	*average*	*factor*		*average*	*factor*
DeltaBlue	12000/1	18	0.1	63	0.2	348	1.0	113	0.3	32	0.1	41	0.1	25	0.1	[a]	22	0.1
Richards	100/1	1'877	0.0	5'773	0.1	39'705	1.0	22'785	0.6	6'479	0.2	7'321	0.2	3'152	0.1		2'972	0.1
Json	100/1	3'196	0.4	4'803	0.6	7'859	1.0	162'598	20.7	7'073	0.9	8'948	1.1	5'921	0.8	[b]	3'301	0.4
Havlak	10/1	984'323	0.1	481'671	0.1	8'185'360	1.0	4'441'696	0.5	1'069'822	0.1	1'151'781	0.1	1'020'599	0.1	[c]	1'132'802	0.1 [e]
CD	250/2	1'755	0.1	2'019	0.1	14'751	1.0	12'122	0.8	1'820	0.1	1'949	0.1	1'794	0.1	[d]	1'166	0.1
Bounce	1500/1	61	0.2	119	0.5	249	1.0	189	0.8	116	0.5	126	0.5	72	0.3		71	0.3
List	1500/1	67	0.1	208	0.3	676	1.0	666	1.0	199	0.3	222	0.3	90	0.1		90	0.1
Mandelbrot	500/1	1	0.5	12	6.0	2	1.0	2	1.0	2	1.0	11	5.5	1	0.5		1	0.5
NBody	250000/1			3	0.4	8	1.0	5	0.6	4	0.5	9	1.1	4	0.5		3	0.4
Permute	1000/1	202	0.6	168	0.5	328	1.0	566	1.7	220	0.7	272	0.8	124	0.4		123	0.4
Queens	1000/1	160	0.5	231	0.8	297	1.0	297	1.0	210	0.7	228	0.8	147	0.5		147	0.5
Sieve	3000/1	56	0.5	103	0.9	119	1.0	93	0.8	84	0.7	99	0.8	30	0.3		32	0.3
Storage	1000/1	778	0.4	310	0.1	2'202	1.0	2'214	1.0	337	0.2	384	0.2	733	0.3		603	0.3
Towers	600/1	275	0.9	307	1.0	299	1.0	507	1.7	500	1.7	482	1.6	263	0.9		262	0.9
sum of averages:		992'769	0.12	495'790	0.06	8'252'203	1.0	4'643'853	0.6	1'086'898	0.13	1'171'873	0.14	1'032'955	0.13		1'141'595	0.14 [f]
geomean of factors:			0.24		0.39		1.0		1.04		0.38		0.51		0.27			0.24
1/geomean:			4.15		2.54		1.00		0.96		2.64		1.98		3.77			4.20
														1.90			2.12	
										Mono 3.12.1		Mono 5.20.1.34		gcc 4.8.2 -O2 just std malloc			gcc 4.8.2 -O2 Boem GC 7.2d	

Benchmarks used from https://github.com/smarr/are-we-fast-yet commit 770c664 3.4.2020
and https://github.com/rochus-keller/Oberon/tree/master/testcases/Are-we-fast-yet
Measurements done between 2021-07-07 - 2021-12-17
Testmachine: HP EliteBook 2530p, Intel Core Duo L9400 1.86GHz, 4GB RAM, Linux i386
All binaries compiled with GCC 4.8.2
LuaJIT params, deviations from default values:

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

Notes:
[a] v0.9.28, failed chain test fixed in v0.9.29
[b] ok in v0.9.27
[c] run with v0.9.28, crash with -On, thus no opt, only 5 rounds
1'024'854 with 2, 1'028'855 with 1 rounds
[d] ok in v0.9.27
[e] crash with -On, thus no optimization, all 10 rounds
1'145'306 with 5 rounds
also crashes under gcc 5.4.0
no crash under clang 3.8.0 on Linux x86_64!!!
no crash under clang 4.0.1 on Linux i386, same overall performance as gcc
no crash under tcc 0.9.27 on Linux i386, overall factor 2 speed-down to gcc
[f] between -O2 and no opt is a speed-down factor 1.8