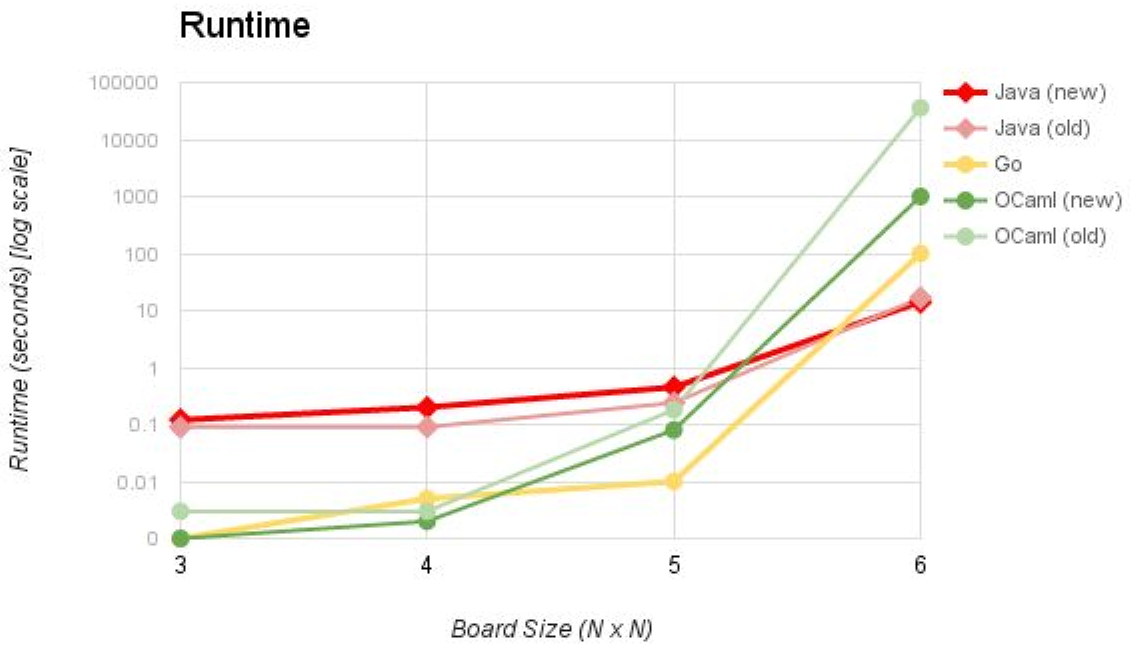
Performance

Runtime

Runtime is measured via the “user”+”sys” time output by the unix time tool (seconds)

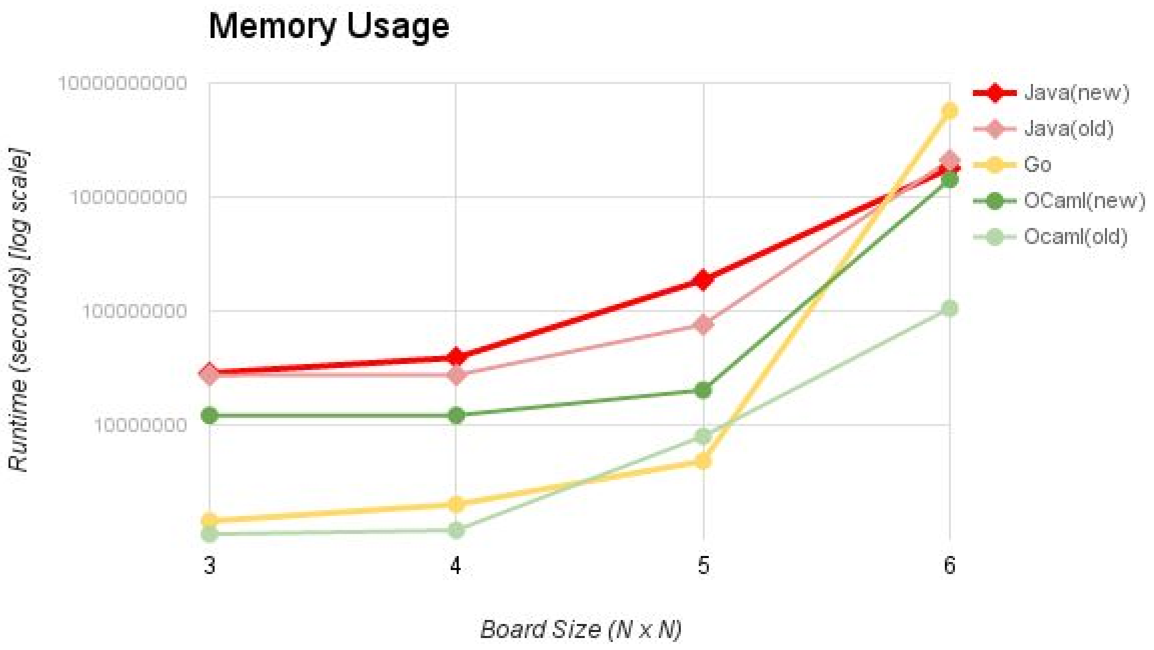
Language	Runtime of N=3	Runtime N=4	Runtime N=5	Runtime N=6
Java	0.09	0.22	0.78	14.49
Go	0.001	0.001	0.01	100 (before oom)
OCaml	0.01	0.01	0.08	36000(before I killed it)



Memory Usage

Memory usage is measured via “maximum resident set size” output by the unix time tool (bytes)

Language	Mem usage of N=3	Mem usage of N=4	Mem usage of N=5	Mem usage of N=6
Java	28499968	38895616	186716160	1780359168
Go	1433600	2007040	4812800	5694324736 (before oom)
OCaml	12128256	12144640	20205568	1425649664 (before killed)



Development

Language	Time(No Code → Robust Impl)	Time(Robust Impl → Concurrent Solution)
Java	6 hours	3 hours
Go	6 hours	2 hours
OCaml	10 hours	4 hours

Language	Time (fighting static checker)	Time(fighting runtime errors)	Grappling with libraries
Java	4 hours	3 hour	2 hour
Go	3 hours	1 hours	2 hours
OCaml	6 hours	2 hours	6 hours

Language	Lines of code (approx)	Notes
Java	350	Had to try 2 different implementations since the first one slowed down execution by adding too much synchronization overhead.
Go	150	Fun, but missing many basic quality of life libraries
OCaml	130	Difficult to benefit from concurrency since multithreading does not occur due to language barriers.