Benchmarking between Lua Script and JavaScript

This benchmarking is based on the hash-list schema and only search operation. The data initialization would be too similar to benchmark since both Lua Script and JavaScript would be mainly calling redis [set] family functions. The search function is chosen to be benchmarked since it involves high frequency I/O. Thus involves high volumes of loops, redis call and if statement, which would be more ideal to compare two languages.

The implementation and algorithm was same as before, which can be referred to documents “Benchmarking between string method and hash”. Benchmark table is shown as below:

Bench-size = 10000

|  |  |  |
| --- | --- | --- |
| Size = 10000 | Lua Script | JavaScript |
| Search Operation | 140.9801059961319 | 1.4848560094833374 |

Bench-size = 25000

|  |  |  |
| --- | --- | --- |
| Size = 25000 | Lua Script | JavaScript |
| Search Operation | 427.57530307769775 | 1.8087010383605957 |

Bench-size = 50000

|  |  |  |
| --- | --- | --- |
| Size = 50000 | Lua Script | JavaScript |
| Search Operation | 1568.6450788974762 | 1.6008570194244385 |

Bench-size = 100000

|  |  |  |
| --- | --- | --- |
| Size = 100000 | Lua Script | JavaScript |
| Search Operation | 605.4833970069885 | 2.584859013557434 |

Lua Script will always be slower, since the language was essentially interpreted twice. First time is through javascript code and second time through built-in integrator in Node.js. It should not be recommended to be used in Node.js, but should be used as .lua script file. However, in that case, it will be hard to be embedded into Node.js.

The sciprt is also included in “/lua/search.lua”.