Crushing Latency with Vert.x

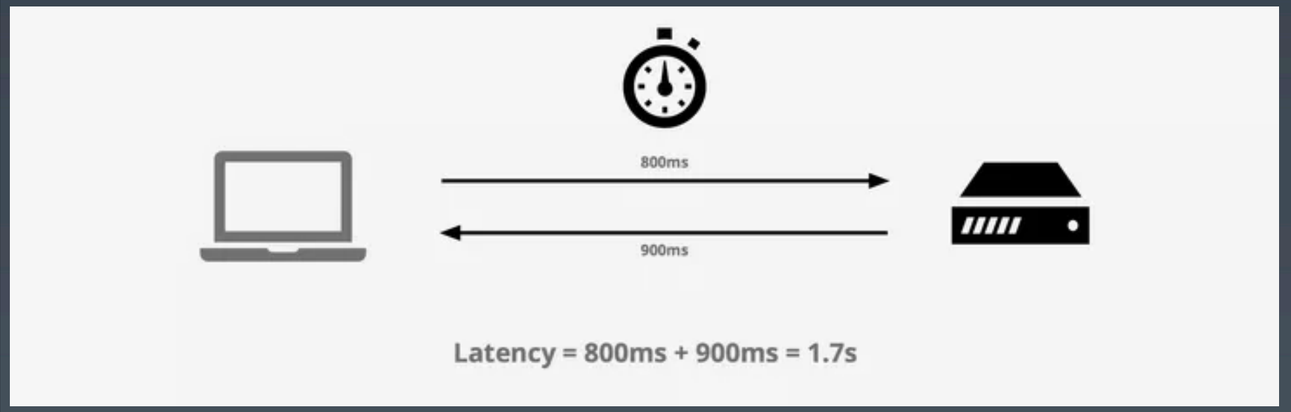
Paulo Lopes

Principal Software Engineer

@pml0pes
https://www.linkedin.com/in/pmlopes/
pmlopes

latency noun

*la·ten·cy | \ ˈlā-tᵊn(t)-sē *



Network latency is the term used to indicate any kind of delay that happens in data communication over a network.

(techopedia.com)

Latency by the numbers

- **Amazon**: every 100ms of latency costs 1% in sales
  http://home.blarg.net/~glinden/StanfordDataMining.2006-11-29.ppt
- **Google**: an extra 0.5 seconds in search page generation time dropped traffic by 20%
  http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html
- **A broker**: could lose $4 million in revenues per millisecond if their electronic trading platform is 5 milliseconds behind the competition
  http://www.tabbgroup.com/PublicationDetail.aspx?PublicationID=346

Latency is not the problem

it's the symptom!

2007: Dan Pritchett

- Loosely Couple Components
- Use Asynchronous Interfaces
- Horizontally Scale from the Start
- Create an Active/Active Architecture
- Use a **BASE** instead of **ACID** Shared Storage Mode

www.infoq.com/articles/pritchett-latency

2011 (Tim Fox): Vert.x

- Loosely Couple Components (event bus)
- Use Asynchronous Interfaces (non blocking I/O)
- Horizontally Scale from the Start (clustered)

Eclipse Vert.x is a tool-kit for building reactive applications on the JVM. https://vertx.io/

Why Non-Blocking I/O?

5ms / req time

...

```
# In optimal circumstances

1 Thread => 200 req/sec
8 Cores => 1600 req/sec
```

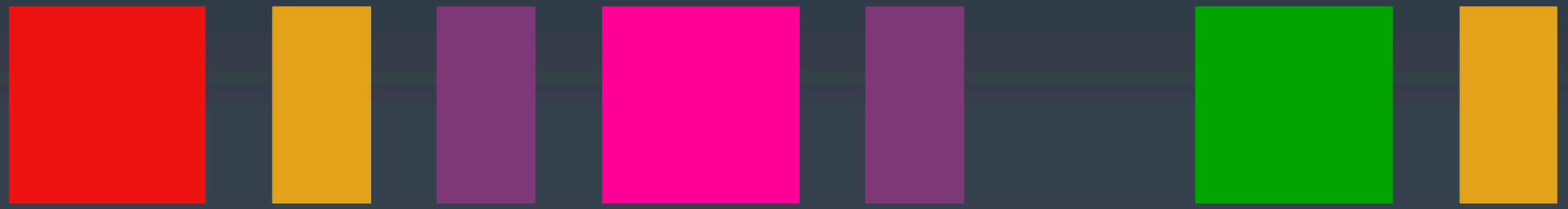
req time grows as threads fight for execution time

...

```
# PROCESS STATE CODES
#   D    Uninterruptible sleep (usually IO)

ps aux | awk '$8 ~ /D/  { print $0 }'
root 9324 0.0 0.0 8316 436 ? D<   Oct15 0:00 /usr/bin/java...
```

when load is higher than max threads queuing builds up



...

```
# git@github.com:tsuna/contextswitch.git

./cpubench.sh
model name : Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz
1 physical CPUs, 4 cores/CPU, 2 hardware threads/core
2000000  thread context switches in 2231974869ns
                                   (1116.0ns/ctxsw)
```


```
grep 'CONFIG_HZ=' /boot/config-$(uname -r)
# CONFIG_HZ=1000
```

Practical example:

Tomcat 9.0

- Default `maxThreads`: **200**
- Avg req time: **5ms**
- Hypothetical High load: **1000 req**
- Wasted wait/queue time: **(1000 / 200 - 1) * 5 = 0~20ms**

https://tomcat.apache.org/tomcat-9.0-doc/config/executor.html

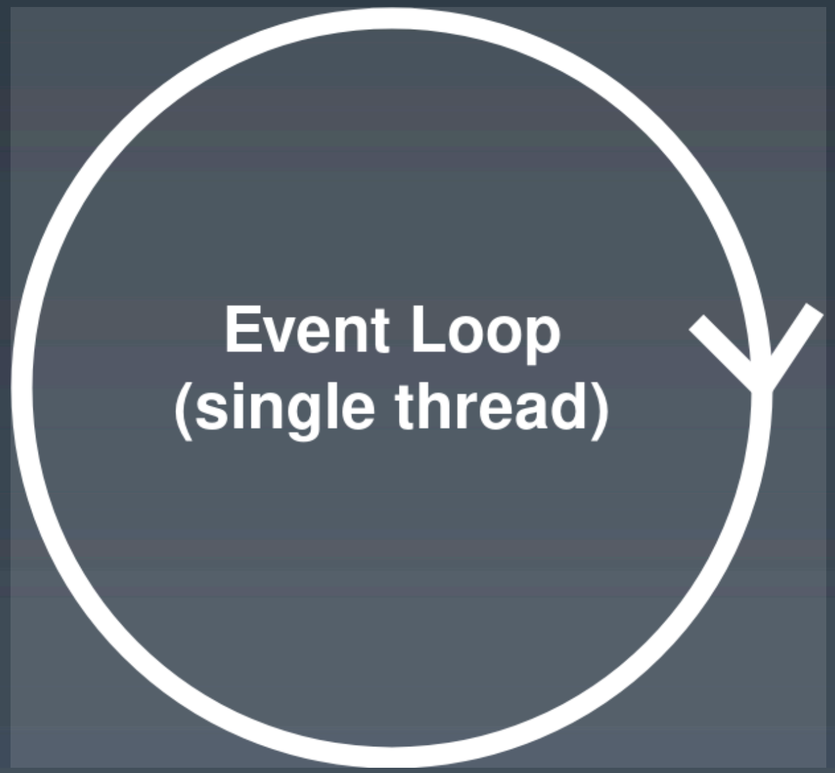
at max utilization

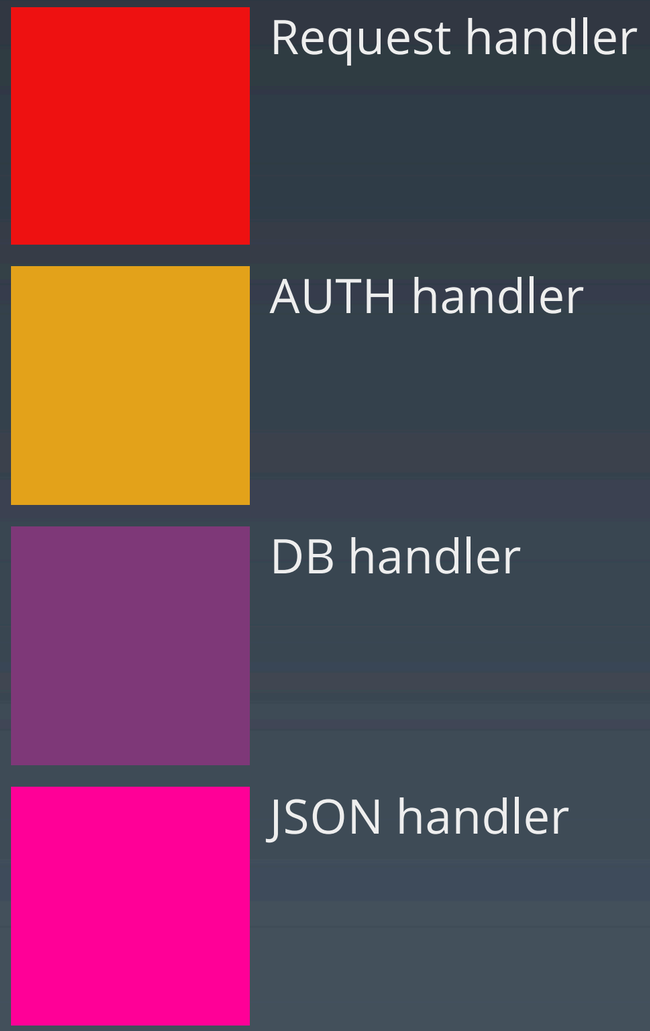
CPU is mostly waiting

Non-Blocking I/O

Vert.x

Events

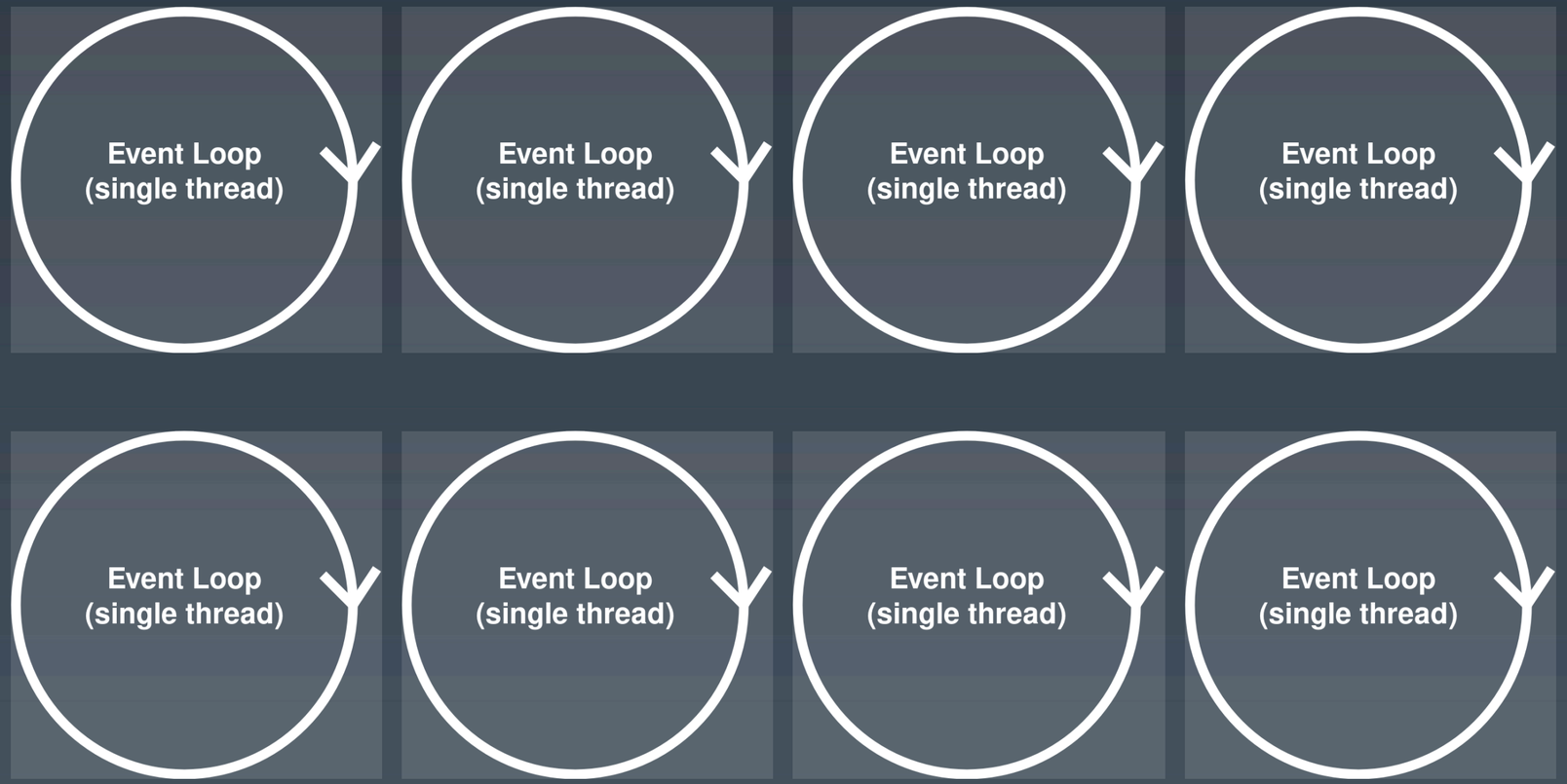




1 CPU core fully used!

Vert.x

Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz



100% CPU cores used!

Benchmarking is hard

- *Meaningful benchmarks are even harder*
- Techempower Framework Benchmarks
  - Contributors: **528**
  - Pull Requests: **4022**
  - Commits: **11095**

https://github.com/TechEmpower/FrameworkBenchmarks

Baseline: JAX-RS

- Blocking API
- Thread Based
- Java

jax-rs

```
@GET
@Path("/queries")
World[]
queries(@QueryParam("queries") String queries)
{
  World[] worlds = new World[queries];
  Session session = emf.createEntityManager();

  for (int i = 0; i < queries; i++) {
    worlds[i] = session
      .byId(World.class)
      .load(randomWorld());
  }

  return worlds;
}
```

vert.x

```
void
queriesHandler(final RoutingContext ctx) {

  World[] worlds = new World[getQueries(ctx)];
  AtomicInteger cnt = new AtomicInteger();

  for (int i = 0; i < getQueries(ctx); i++) {
    db.preparedQuery(FIND_WORLD, ..., res -> {
      final Row row = res.result()
                  .iterator()
                  .next();

      worlds[cnt.incrementAndGet()] =
        new World(row);

      if (cnt.get() == queries) {
        ctx.response()
          .end(Json.encodeToBuffer(worlds));
      }
    });
  }
}
```

Multiple queries

Responses per second at 20 queries per request, Test environment (4 tests)

Rnk	Framework	Performance (higher is better)		Errors	Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA
1	vertx-web-postgres	37,157	100.0% (50.4%)	0	Mcr	Jav	ver	Non	Lin	Pg	Lin	Raw	Rea
3	undertow-jersey-hikaricp	14,493	39.0% (19.6%)	0	Plt	Jav	JAX	Non	Lin	My	Lin	Ful	Rea

Requirements summary

In this test, each request is processed by fetching multiple rows from a simple database table and serializing these rows as a JSON response. The test is run multiple times: testing 1, 5, 10, 15, and 20 queries per request. All tests are run at 512 concurrency.

Example response for 10 queries:

```
HTTP/1.1 200 OK
Content-Length: 315
Content-Type: application/json
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT

[{"id":4174,"randomNumber":331},{"id":51,"randomNumber":6544},{"id":4462,"randomNumber":952},{"id":2221,"randomNumber":532},{"id":9276,"randomNumber":3097},
{"id":3056,"randomNumber":7293},{"id":6964,"randomNumber":620},{"id":675,"randomNumber":6601},{"id":8414,"randomNumber":6569},
{"id":2753,"randomNumber":4065}]
```

For a more detailed description of the requirements, see the Source Code and Requirements section.

Simple results

- Vert.x: **37,157** req/s
- Jax-RS: **14,493** req/s


DID YOU
KNOW?

Polyglot

English - 简体中文 - Português

What happens when you say Hello?

```
function handler (context) {
  // the exchange context
  context
  // get the response object
  .response()
  // send the message and end
  // the response
  .end('你好');
}
```

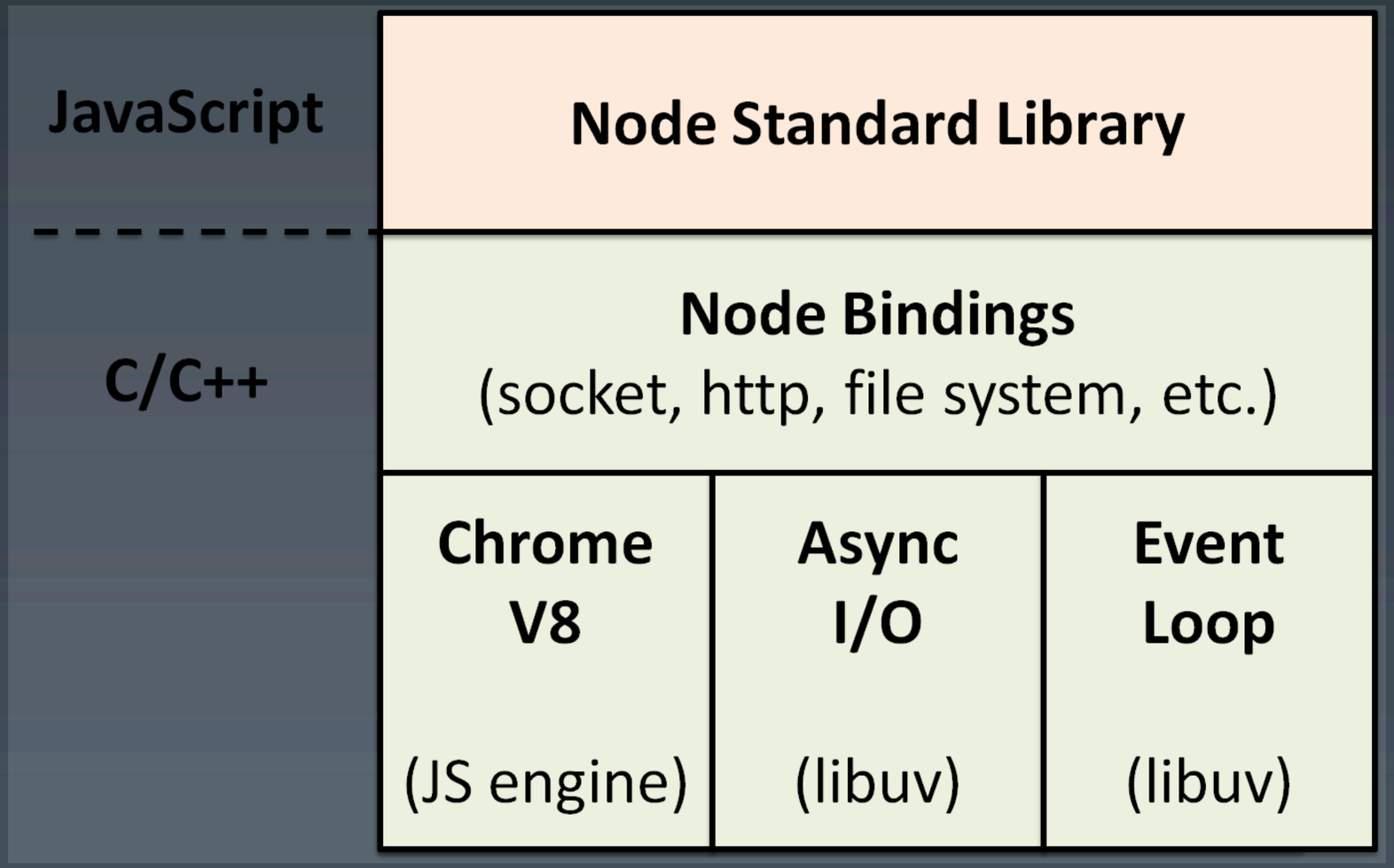
Getting the response object

```
this.response =  function() {
  var __args = arguments;
  if (__args.length === 0) {
    if (that.cachedresponse == null) {
      that.cachedresponse = utils.convReturnVertxGen(
        HttpServerResponse,
        j_routingContext["response()"]());
    }
    return that.cachedresponse;
  } else if (typeof __super_response != 'undefined') {
    return __super_response.apply(this, __args);
  }
  else throw new TypeError('invalid arguments');
};
```

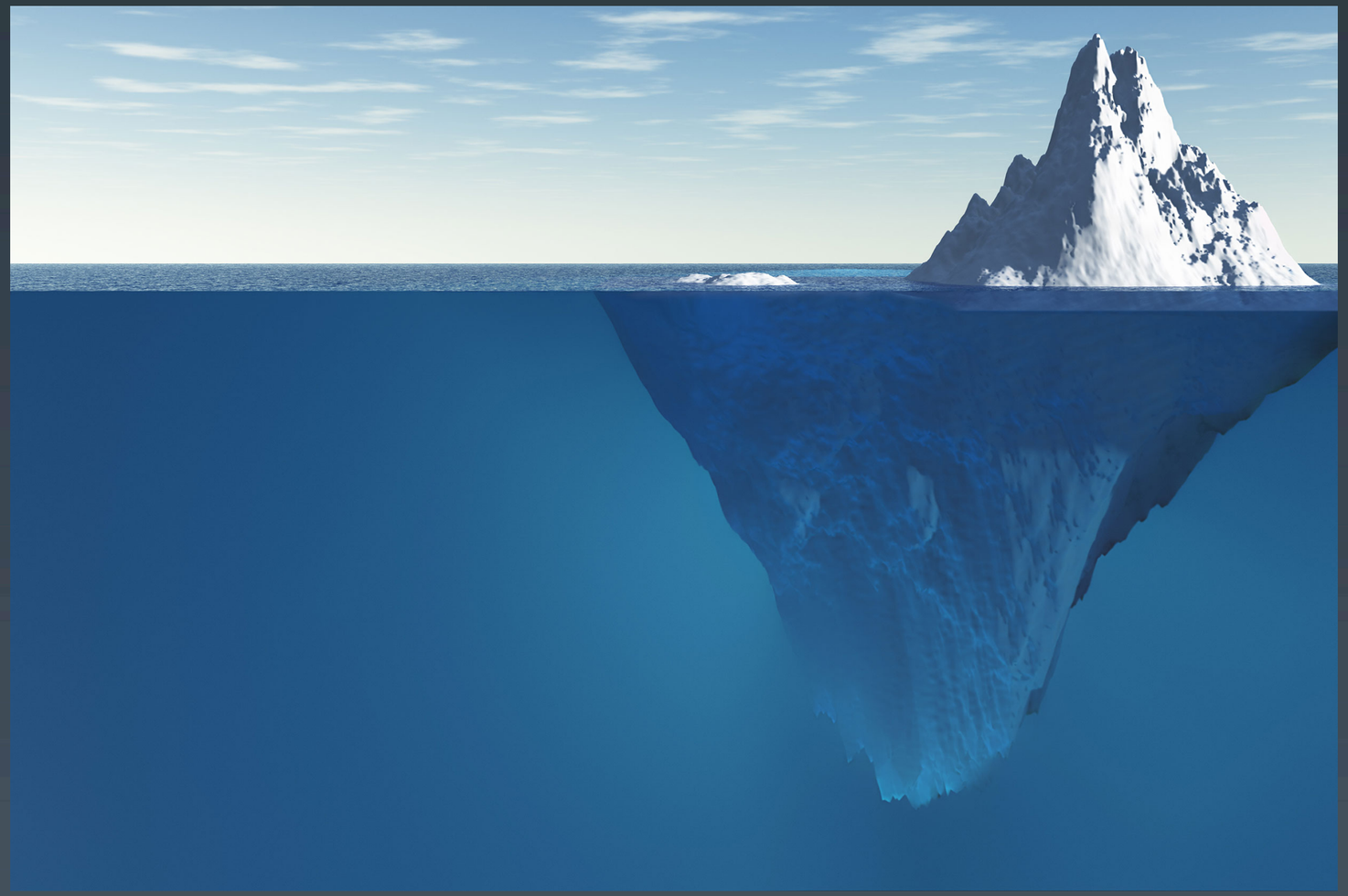
In a nutshell

- Lots of conversions (GC)
- Constant switch from JS engine to Java code (somehow similar to context switching)
- Not suited for performance
- **JIT optimization will stop at language cross**

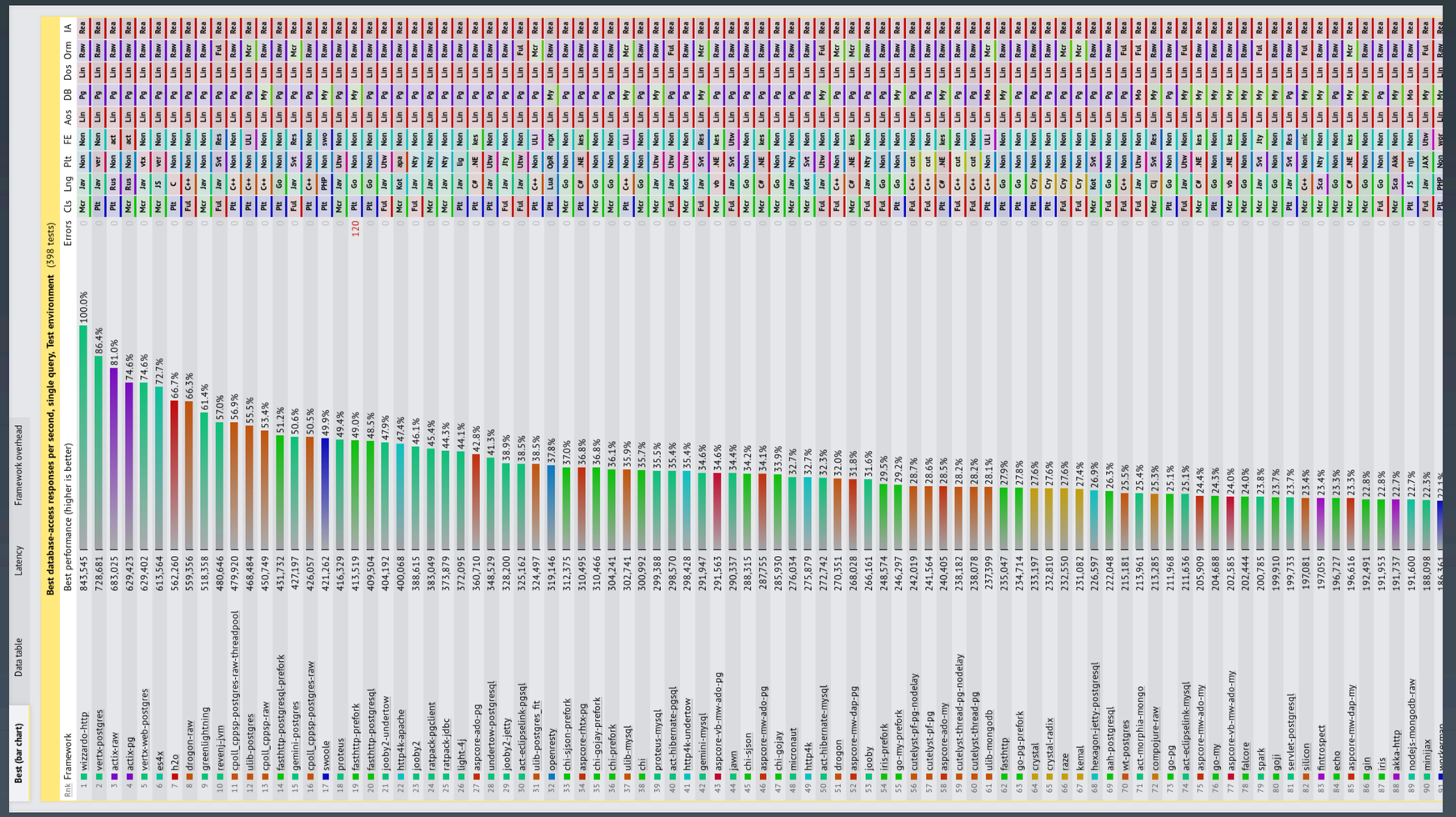
Node.js


JavaScript
C/C++
Node Standard Library
Node Bindings
(socket, http, file system, etc.)
Chrome V8
(JS engine)
Async I/O
(libuv)
Event Loop
(libuv)

JIT can't optimize it all



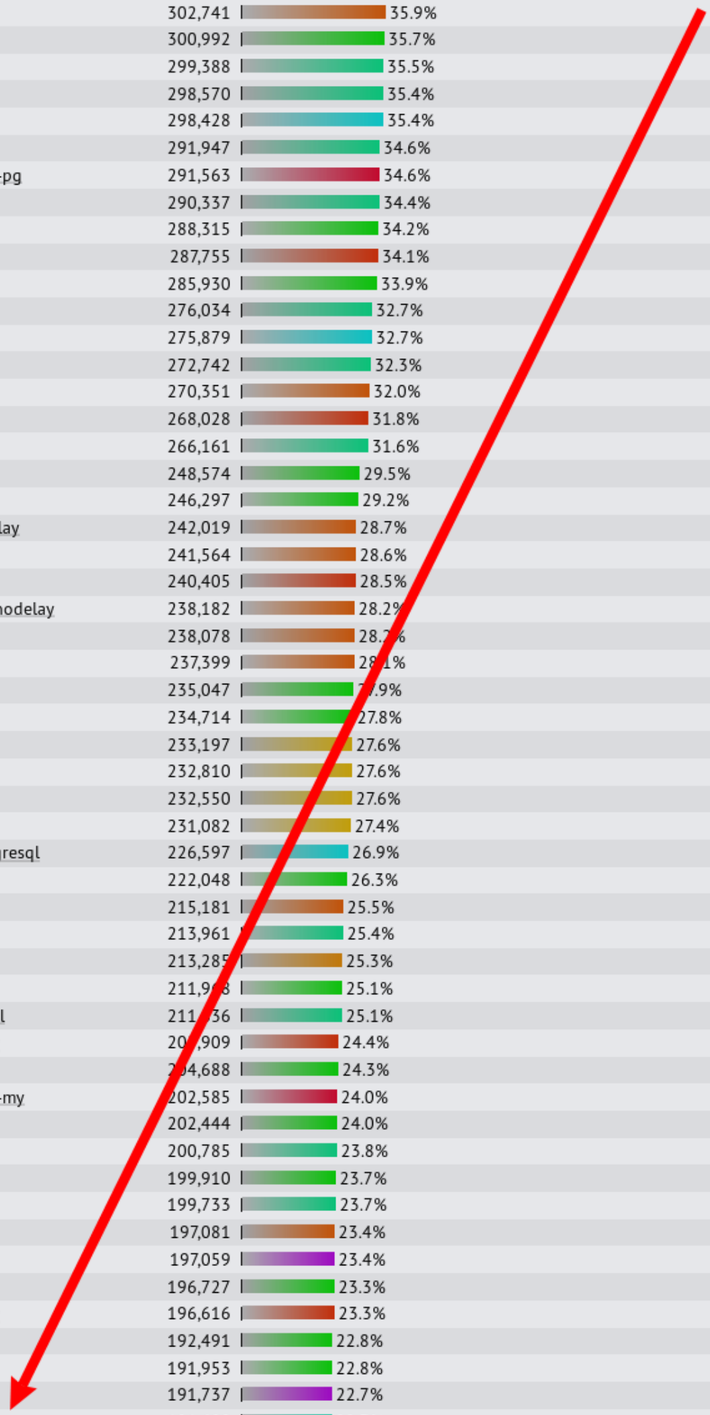
Where's Node?


Best (bar chart)
Data table
Latency
Framework overhead
Best database-access responses per second, single query. Test environment (398 tests)
Best performance (higher is better)
Rnk Framework
Errors Cls Lng Plt FE Aos DB Dos Orm IA
1 wizzardo-http 0 Mcr Jav Non Lin Pg Lin Raw Rea
2 vertx-postgres 0 Plt Jav ver Non Lin Pg Lin Raw Rea
3 actix-raw 0 Plt Rus Non act Lin Pg Lin Raw Rea
4 actix-pg 0 Mcr Rus Non act Lin Pg Lin Raw Rea
5 vertx-web-postgres 0 Mcr Jav vtx Non Lin Pg Lin Raw Rea
6 es4x 0 Mcr JS ver Non Lin Pg Lin Raw Rea
7 h2o 0 Plt C Non Non Lin Pg Lin Raw Rea
8 drogon-raw 0 Ful C++ Non Non Lin Pg Lin Raw Rea
9 greenlightning 0 Mcr Jav Non Non Lin Pg Lin Raw Rea
10 revenj-jvm 0 Ful Jav Svt Res Lin Pg Lin Ful Rea
11 cpoll_cppsp-postgres-raw-threadpool 0 Plt C++ Non Non Lin Pg Lin Raw Rea
12 ulib-postgres 0 Plt C++ Non ULi Lin Pg Lin Raw Rea
13 cpoll_cppsp-raw 0 Plt C++ Non Non Lin My Lin Raw Rea
14 fasthttp-postgresql-prefork 0 Plt Go Non Non Lin Pg Lin Raw Rea
15 gemini-postgres 0 Ful Jav Svt Res Lin Pg Lin Mcr Rea
16 cpoll_cppsp-postgres-raw 0 Plt C++ Non Non Lin Pg Lin Raw Rea
17 swoole 0 Plt PHP Non swo Lin My Lin Raw Rea
18 proteus 0 Mcr Jav Non Non Lin My Lin Raw Rea
19 fasthttp-prefork 120 Plt Go Non Non Lin My Lin Raw Rea
20 fasthttp-postgresql 0 Plt Go Non Non Lin Pg Lin Raw Rea
21 jooby2-undertow 0 Ful Jav Svt Utw Non Lin Pg Lin Raw Rea
22 http4k-apache 0 Mcr Kot apa Non Lin Pg Lin Raw Rea
23 jooby2 0 Ful Jav Nty Non Lin Pg Lin Raw Rea
24 ratpack-pgclient 0 Mcr Jav Nty Non Lin Pg Lin Raw Rea
25 ratpack-jdbc 0 Mcr Jav Nty Non Lin Pg Lin Raw Rea
26 light-4j 0 Plt Jav lig Non Lin Pg Lin Raw Rea
27 aspcore-ado-pg 0 Plt C# .NE kes Non Lin Pg Lin Raw Rea
28 undertow-postgresql 0 Plt Jav Utw Non Lin Pg Lin Raw Rea
29 jooby2-jetty 0 Ful Jav Jty Non Lin Pg Lin Raw Rea
30 act-eclipselink-pgsql 0 Ful Jav Utw Non Lin Pg Lin Ful Rea
31 ulib-postgres_ft 0 Plt C++ Non ULi Lin Pg Lin Mcr Rea
32 openresty 0 Plt Lua OpR ngx Non Lin My Lin Raw Rea
33 chi-sjson-prefork 0 Mcr Go Non Non Lin Pg Lin Raw Rea
34 aspcore-fhtx-pg 0 Plt C# .NE kes Non Lin Pg Lin Raw Rea
35 chi-gojay-prefork 0 Mcr Go Non Non Lin Pg Lin Raw Rea
36 chi-prefork 0 Plt C++ Non Non Lin My Lin Raw Rea
37 ulib-mysql 0 Plt C++ Non ULi Lin My Lin Mcr Rea
38 chi 0 Mcr Go Non Non Lin My Lin Raw Rea
39 proteus-mysql 0 Mcr Jav Utw Non Lin My Lin Raw Rea
40 act-hibernate-pgsql 0 Ful Jav Utw Non Lin Pg Lin Ful Rea
41 http4k-undertow 0 Mcr Kot Utw Non Lin Pg Lin Raw Rea
42 gemini-mysql 0 Ful Jav Svt Res Lin My Lin Mcr Rea
43 aspcore-vb-mw-ado-pg 0 Mcr vb .NE kes Lin Pg Lin Raw Rea
44 jawn 0 Ful Jav Svt Utw Lin Pg Lin Raw Rea
45 chi-sjson 0 Mcr Go Non Non Lin Pg Lin Raw Rea
46 aspcore-mw-ado-pg 0 Mcr C# .NE kes Lin Pg Lin Raw Rea
47 chi-gojay 0 Mcr Go Non Non Lin Pg Lin Raw Rea
48 micronaut 0 Mcr Jav Nty Non Lin Pg Lin Raw Rea
49 http4k 0 Mcr Kot Svt Non Lin Pg Lin Raw Rea
50 act-hibernate-mysql 0 Ful Jav Utw Non Lin My Lin Ful Rea
51 drogon 0 Ful C++ Non Non Lin Pg Lin Mcr Rea
52 aspcore-mw-dap-pg 0 Mcr C# .NE kes Lin Pg Lin Raw Rea
53 jooby 0 Ful Jav Nty Non Lin Pg Lin Ful Rea
54 iris-prefork 0 Ful Go Non Non Lin Pg Lin Raw Rea
55 go-my-prefork 0 Plt Go Non Non Lin My Lin Raw Rea
56 cutelyst-pf-pg-nodelay 0 Ful C++ cut Non Lin Pg Lin Raw Rea
57 cutelyst-pf-pg 0 Ful C++ cut Non Lin Pg Lin Raw Rea
58 aspcore-ado-my 0 Plt C# .NE kes Non Lin My Lin Raw Rea
59 cutelyst-thread-pg-nodelay 0 Ful C++ cut Non Lin Pg Lin Raw Rea
60 cutelyst-thread-pg 0 Ful C++ cut Non Lin Pg Lin Raw Rea
61 ulib-mongodb 0 Plt C++ Non ULi Non Lin Mo Lin Mcr Rea
62 fasthttp 0 Plt Go Non Non Lin My Lin Raw Rea
63 go-pg-prefork 0 Plt Go Non Non Lin Pg Lin Raw Rea
64 crystal 0 Plt Cry Non Non Lin Pg Lin Raw Rea
65 crystal-radix 0 Plt Cry Non Non Lin Pg Lin Raw Rea
66 raze 0 Ful Cry Non Non Lin Pg Lin Mcr Rea
67 kemal 0 Plt Cry Non Non Lin Pg Lin Mcr Rea
68 hexagon-jetty-postgresql 0 Mcr Kot Svt Non Lin Pg Lin Raw Rea
69 aah-postgresql 0 Ful Go Non Non Lin Pg Lin Raw Rea
70 wt-postgres 0 Ful C++ Non Non Lin Pg Lin Ful Rea
71 act-morphia-mongo 0 Ful Jav Utw Non Lin Mo Lin Ful Rea
72 compojure-raw 0 Mcr Clj Svt Res Lin My Lin Raw Rea
73 go-pg 0 Plt Go Non Non Lin Pg Lin Raw Rea
74 act-eclipselink-mysql 0 Ful Jav Utw Non Lin My Lin Ful Rea
75 aspcore-mw-ado-my 0 Mcr C# .NE kes Lin My Lin Raw Rea
76 go-my 0 Plt Go Non Non Lin My Lin Raw Rea
77 aspcore-vb-mw-ado-my 0 Mcr vb .NE kes Lin My Lin Raw Rea
78 falcore 0 Mcr Go Non Non Lin My Lin Raw Rea
79 spark 0 Mcr Jav Svt Jty Lin My Lin Ful Rea
80 goji 0 Mcr Go Non Non Lin My Lin Raw Rea
81 servlet-postgresql 0 Plt Jav Svt Res Lin Pg Lin Raw Rea
82 silicon 0 Mcr C++ Non mic Lin My Lin Ful Rea
83 fintrospect 0 Mcr Sca Nty Non Lin My Lin Raw Rea
84 echo 0 Mcr Go Non Non Lin My Lin Raw Rea
85 aspcore-mw-dap-my 0 Mcr C# .NE kes Lin My Lin Mcr Rea
86 gin 0 Mcr Go Non Non Lin My Lin Raw Rea
87 iris 0 Ful Go Non Non Lin Pg Lin Raw Rea
88 akka-http 0 Mcr Sca Akk Non Lin My Lin Raw Rea
89 nodejs-mongodb-raw 0 Plt JS njs Non Lin Mo Lin Raw Rea
90 minijax 0 Ful Jav JAX Utw Non Lin My Lin Ful Rea
91 woodeaman 0 Plt PHP Non wor Lin My Lin Raw Rea
843,545 100.0%
728,681 86.4%
683,025 81.0%
629,423 74.6%
629,402 74.6%
613,564 72.7%
562,260 66.7%
559,356 66.3%
518,358 61.4%
480,646 57.0%
479,920 56.9%
468,484 55.5%
450,749 53.4%
431,732 51.2%
427,197 50.6%
426,057 50.5%
421,262 49.9%
416,329 49.4%
413,519 49.0%
409,504 48.5%
404,192 47.9%
400,068 47.4%
388,615 46.1%
383,049 45.4%
373,879 44.3%
372,095 44.1%
360,710 42.8%
348,529 41.3%
328,200 38.9%
325,162 38.5%
324,497 38.5%
319,146 37.8%
312,375 37.0%
310,495 36.8%
310,466 36.8%
304,241 36.1%
302,741 35.9%
300,992 35.7%
299,388 35.5%
298,570 35.4%
298,428 35.4%
291,947 34.6%
291,563 34.6%
290,337 34.4%
288,315 34.2%
287,755 34.1%
285,930 33.9%
276,034 32.7%
275,879 32.7%
272,742 32.3%
270,551 32.0%
268,028 31.8%
266,161 31.6%
248,574 29.5%
246,297 29.2%
242,019 28.7%
241,564 28.6%
240,405 28.5%
238,182 28.2%
238,078 28.2%
237,399 28.1%
235,047 27.9%
234,714 27.8%
233,197 27.6%
232,810 27.6%
232,550 27.6%
231,082 27.4%
226,597 26.9%
222,048 26.3%
215,181 25.5%
213,961 25.4%
213,285 25.3%
211,968 25.1%
211,636 25.1%
205,909 24.4%
204,688 24.3%
202,585 24.0%
202,444 24.0%
200,785 23.8%
199,910 23.7%
199,733 23.7%
197,081 23.4%
197,059 23.4%
196,727 23.3%
196,616 23.3%
192,491 22.8%
191,953 22.8%
191,737 22.7%
191,600 22.7%
188,098 22.3%

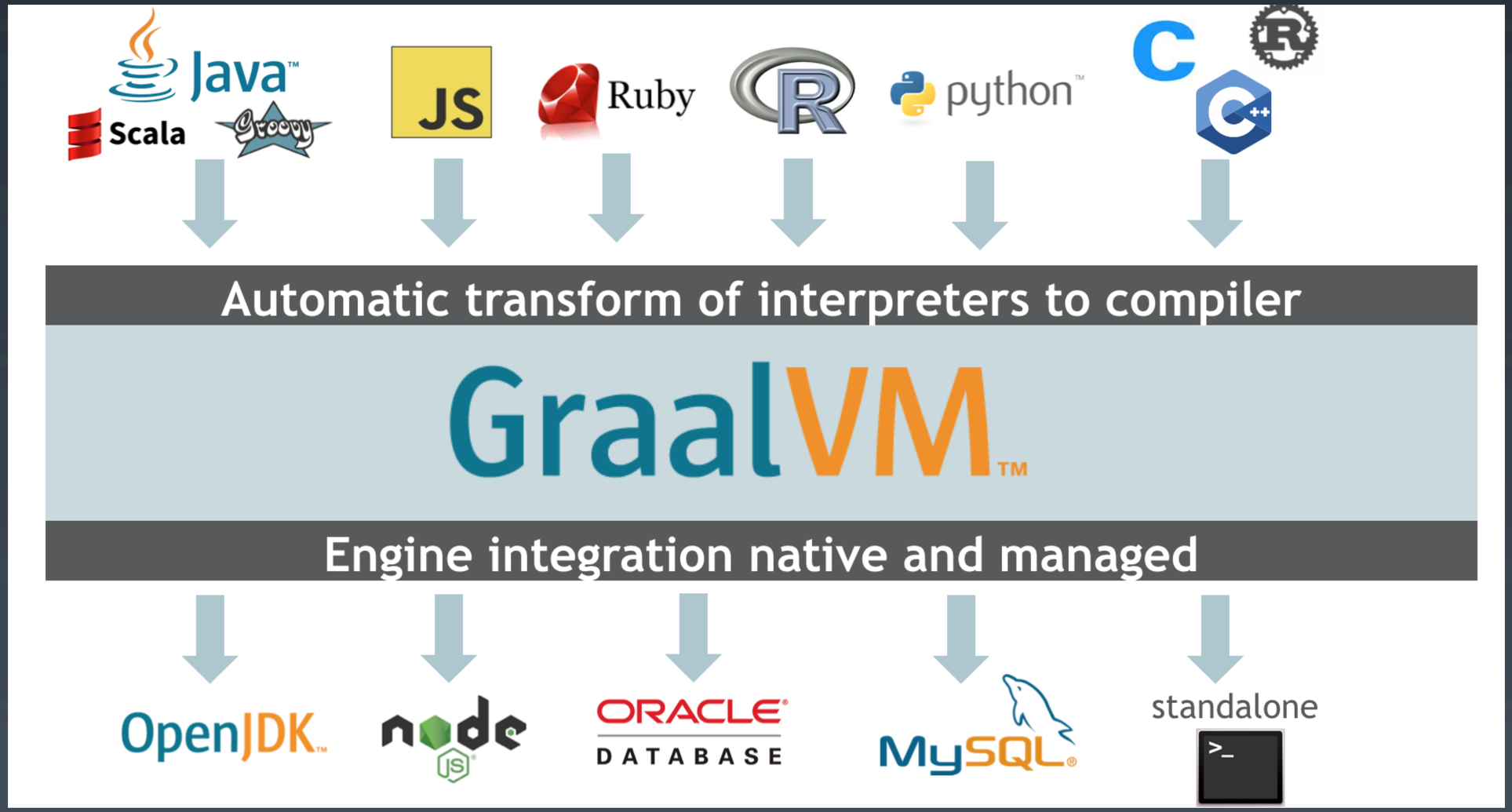
Best (bar chart) | Data table | Latency | Framework overhead

Best database-access responses per second, single query, Test environment (398 tests)

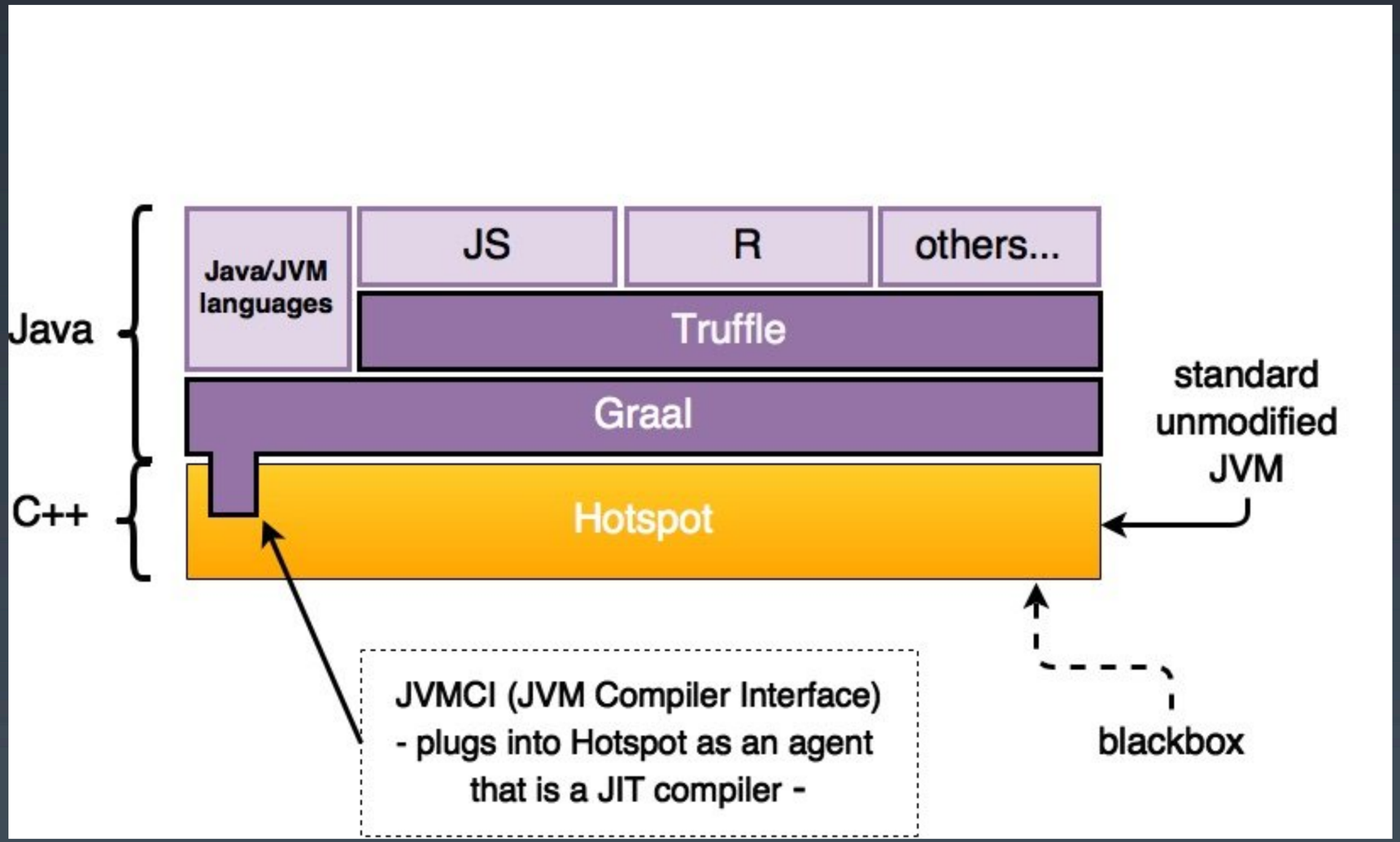
Rnk	Framework	Best performance (higher is better)	Errors	Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA
1	wizzardo-http	843,545 100.0%	0	Mcr	Jav	Non	Non	Lin	Pg	Lin	Raw	Rea
2	vertx-postgres	728,681 86.4%	0	Plt	Jav	ver	Non	Lin	Pg	Lin	Raw	Rea
3	actix-raw	683,025 81.0%	0	Plt	Rus	Non	act	Lin	Pg	Lin	Raw	Rea
4	actix-pg	629,423 74.6%	0	Mcr	Rus	Non	act	Lin	Pg	Lin	Raw	Rea
5	vertx-web-postgres	629,402 74.6%	0	Mcr	Jav	vtx	Non	Lin	Pg	Lin	Raw	Rea
6	es4x	613,564 72.7%	0	Mcr	JS	ver	Non	Lin	Pg	Lin	Raw	Rea
7	h2o	562,260 66.7%	0	Plt	C	Non	Non	Lin	Pg	Lin	Raw	Rea
8	drogon-raw	559,356 66.3%	0	Ful	C++	Non	Non	Lin	Pg	Lin	Raw	Rea
9	greenlightning	518,358 61.4%	0	Mcr	Jav	Non	Non	Lin	Pg	Lin	Raw	Rea
10	revenj-jvm	480,646 57.0%	0	Ful	Jav	Svt	Res	Lin	Pg	Lin	Ful	Rea
11	cpoll_cppsp-postgres-raw-threadpool	479,920 56.9%	0	Plt	C++	Non	Non	Lin	Pg	Lin	Raw	Rea
12	ulib-postgres	468,484 55.5%	0	Plt	C++	Non	ULi	Lin	Pg	Lin	Mcr	Rea
13	cpoll_cppsp-raw	450,749 53.4%	0	Plt	C++	Non	Non	Lin	My	Lin	Raw	Rea
14	fasthttp-postgresql-prefork	431,732 51.2%	0	Plt	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
15	gemini-postgres	427,197 50.6%	0	Ful	Jav	Svt	Res	Lin	Pg	Lin	Mcr	Rea
16	cpoll_cppsp-postgres-raw	426,057 50.5%	0	Plt	C++	Non	Non	Lin	Pg	Lin	Raw	Rea
17	swoole	421,262 49.9%	0	Plt	PHP	Non	swo	Lin	My	Lin	Raw	Rea
18	proteus	416,329 49.4%	0	Mcr	Jav	Utw	Non	Lin	Pg	Lin	Raw	Rea
19	fasthttp-prefork	413,519 49.0%	120	Plt	Go	Non	Non	Lin	My	Lin	Raw	Rea
20	fasthttp-postgresql	409,504 48.5%	0	Plt	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
21	jooby2-undertow	404,192 47.9%	0	Ful	Jav	Utw	Non	Lin	Pg	Lin	Raw	Rea
22	http4k-apache	400,068 47.4%	0	Mcr	Kot	apa	Non	Lin	Pg	Lin	Raw	Rea
23	jooby2	388,615 46.1%	0	Ful	Jav	Nty	Non	Lin	Pg	Lin	Raw	Rea
24	ratpack-pgclient	383,049 45.4%	0	Mcr	Jav	Nty	Non	Lin	Pg	Lin	Raw	Rea
25	ratpack-jdbc	373,879 44.3%	0	Mcr	Jav	Nty	Non	Lin	Pg	Lin	Raw	Rea
26	light-4j	372,095 44.1%	0	Plt	Jav	lig	Non	Lin	Pg	Lin	Raw	Rea
27	aspcore-ado-pg	360,710 42.8%	0	Plt	C#	.NE	kes	Lin	Pg	Lin	Raw	Rea
28	undertow-postgresql	348,529 41.3%	0	Plt	Jav	Utw	Non	Lin	Pg	Lin	Raw	Rea
29	jooby2-jetty	328,200 38.9%	0	Ful	Jav	Jty	Non	Lin	Pg	Lin	Raw	Rea
30	act-eclipselink-pgsql	325,162 38.5%	0	Ful	Jav	Utw	Non	Lin	Pg	Lin	Ful	Rea
31	ulib-postgres_fit	324,497 38.5%	0	Plt	C++	Non	ULi	Lin	Pg	Lin	Mcr	Rea
32	openresty	319,146 37.8%	0	Plt	Lua	OpR	ngx	Lin	My	Lin	Raw	Rea
33	chi-sjson-prefork	312,375 37.0%	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
34	aspcore-rhtx-pg	310,495 36.8%	0	Plt	C#	.NE	kes	Lin	Pg	Lin	Raw	Rea
35	chi-gojay-prefork	310,466 36.8%	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
36	chi-prefork	304,241 36.1%	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
37	ulib-mysql	302,741 35.9%	0	Plt	C++	Non	ULi	Lin	My	Lin	Mcr	Rea
38	chi	300,992 35.7%	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
39	proteus-mysql	299,388 35.5%	0	Mcr	Jav	Utw	Non	Lin	My	Lin	Raw	Rea
40	act-hibernate-pgsql	298,570 35.4%	0	Ful	Jav	Utw	Non	Lin	Pg	Lin	Ful	Rea
41	http4k-undertow	298,428 35.4%	0	Mcr	Kot	Utw	Non	Lin	Pg	Lin	Raw	Rea
42	gemini-mysql	291,947 34.6%	0	Ful	Jav	Svt	Res	Lin	My	Lin	Mcr	Rea
43	aspcore-vb-mw-ado-pg	291,563 34.6%	0	Mcr	vb	.NE	kes	Lin	Pg	Lin	Raw	Rea
44	jawn	290,337 34.4%	0	Ful	Jav	Svt	Utw	Lin	Pg	Lin	Raw	Rea
45	chi-sjson	288,315 34.2%	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
46	aspcore-mw-ado-pg	287,755 34.1%	0	Mcr	C#	.NE	kes	Lin	Pg	Lin	Raw	Rea
47	chi-gojay	285,930 33.9%	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
48	micronaut	276,034 32.7%	0	Mcr	Jav	Nty	Non	Lin	Pg	Lin	Raw	Rea
49	http4k	275,879 32.7%	0	Mcr	Kot	Svt	Non	Lin	Pg	Lin	Raw	Rea
50	act-hibernate-mysql	272,742 32.3%	0	Ful	Jav	Utw	Non	Lin	My	Lin	Ful	Rea
51	drogon	270,351 32.0%	0	Ful	C++	Non	Non	Lin	Pg	Lin	Mcr	Rea
52	aspcore-mw-dap-pg	268,028 31.8%	0	Mcr	C#	.NE	kes	Lin	Pg	Lin	Mcr	Rea
53	jooby	266,161 31.6%	0	Ful	Jav	Nty	Non	Lin	Pg	Lin	Raw	Rea
54	iris-prefork	248,574 29.5%	0	Ful	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
55	go-my-prefork	246,297 29.2%	0	Plt	Go	Non	Non	Lin	My	Lin	Raw	Rea
56	cutelyst-pf-pg-nodelay	242,019 28.7%	0	Ful	C++	cut	Non	Lin	Pg	Lin	Raw	Rea
57	cutelyst-pf-pg	241,564 28.6%	0	Ful	C++	cut	Non	Lin	Pg	Lin	Raw	Rea
58	aspcore-ado-my	240,405 28.5%	0	Plt	C#	.NE	kes	Lin	My	Lin	Raw	Rea
59	cutelyst-thread-pg-nodelay	238,182 28.2%	0	Ful	C++	cut	Non	Lin	Pg	Lin	Raw	Rea
60	cutelyst-thread-pg	238,078 28.2%	0	Ful	C++	cut	Non	Lin	Pg	Lin	Raw	Rea
61	ulib-mongodb	237,399 28.1%	0	Plt	C++	Non	ULi	Lin	Mo	Lin	Mcr	Rea
62	fasthttp	235,047 27.9%	0	Plt	Go	Non	Non	Lin	My	Lin	Raw	Rea
63	go-pg-prefork	234,714 27.8%	0	Plt	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
64	crystal	233,197 27.6%	0	Plt	Cry	Non	Non	Lin	Pg	Lin	Raw	Rea
65	crystal-radix	232,810 27.6%	0	Plt	Cry	Non	Non	Lin	Pg	Lin	Raw	Rea
66	raze	232,550 27.6%	0	Ful	Cry	Non	Non	Lin	Pg	Lin	Mcr	Rea
67	kemal	231,082 27.4%	0	Ful	Cry	Non	Non	Lin	Pg	Lin	Mcr	Rea
68	hexagon-jetty-postgresql	226,597 26.9%	0	Mcr	Kot	Svt	Non	Lin	Pg	Lin	Raw	Rea
69	aah-postgresql	222,048 26.3%	0	Ful	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
70	wt-postgres	215,181 25.5%	0	Ful	C++	Non	Non	Lin	Pg	Lin	Ful	Rea
71	act-morphia-mongo	213,961 25.4%	0	Ful	Jav	Utw	Non	Lin	Mo	Lin	Ful	Rea
72	compojure-raw	213,285 25.3%	0	Mcr	Clj	Svt	Res	Lin	My	Lin	Raw	Rea
73	go-pg	211,968 25.1%	0	Plt	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
74	act-eclipselink-mysql	211,736 25.1%	0	Ful	Jav	Utw	Non	Lin	My	Lin	Ful	Rea
75	aspcore-mw-ado-my	206,909 24.4%	0	Mcr	C#	.NE	kes	Lin	My	Lin	Raw	Rea
76	go-my	204,688 24.3%	0	Plt	Go	Non	Non	Lin	My	Lin	Raw	Rea
77	aspcore-vb-mw-ado-my	202,585 24.0%	0	Mcr	vb	.NE	kes	Lin	My	Lin	Raw	Rea
78	falcore	202,444 24.0%	0	Mcr	Go	Non	Non	Lin	My	Lin	Raw	Rea
79	spark	200,785 23.8%	0	Mcr	Jav	Svt	Jty	Lin	My	Lin	Ful	Rea
80	goji	199,910 23.7%	0	Mcr	Go	Non	Non	Lin	My	Lin	Raw	Rea
81	servlet-postgresql	199,733 23.7%	0	Plt	Jav	Svt	Res	Lin	Pg	Lin	Raw	Rea
82	silicon	197,081 23.4%	0	Mcr	C++	Non	mic	Lin	My	Lin	Ful	Rea
83	fintrospect	197,059 23.4%	0	Mcr	Sca	Nty	Non	Lin	My	Lin	Raw	Rea
84	echo	196,727 23.3%	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
85	aspcore-mw-dap-my	196,616 23.3%	0	Mcr	C#	.NE	kes	Lin	My	Lin	Mcr	Rea
86	gin	192,491 22.8%	0	Mcr	Go	Non	Non	Lin	My	Lin	Raw	Rea
87	iris	191,953 22.8%	0	Ful	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
88	akka-http	191,737 22.7%	0	Mcr	Sca	Akk	Non	Lin	My	Lin	Raw	Rea
89	nodejs-mongodb-raw	191,600 22.7%	0	Plt	JS	njs	Non	Lin	Mo	Lin	Raw	Rea
90	minijax	188,098 22.3%	0	Ful	Jav	JAX	Utw	Lin	My	Lin	Ful	Rea
91	workerman	186,361 22.1%	0	Plt	PHP	Non	wor	Lin	My	Lin	Raw	Rea



Can we make polyglot fast?



https://graalvm.org



https://graalvm.org

GraalVM: In a nutshell

- ~~Lots of conversions (GC)~~
- ~~Constant switch from JS engine to Java code (somehow similar to context switching)~~
- ~~Not suited for performance~~
- **~~JIT optimization will stop at language cross~~**

ES4X

- GraalVM based
- Vert.x for I/O
- commonjs and ESM loader
- debug/profile chrome-devtools

https://reactiverse.io/es4x

ES4X design principles

- GraalJS (for fast JS runtime)
- Vert.x (for fast I/O + event loops)
- GraalVM (for full JIT)
- `.d.ts` (for IDE support)

```
github.com/AlDanial/cloc v 1.72  T=0.09 s (401.5 files/s, 51963.8 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Java                            26            389            683           1778
JavaScript                       9            201            253           1226
-------------------------------------------------------------------------------
SUM:                            35            590            936           3004
-------------------------------------------------------------------------------
```

Node.js vs ES4X

```
const cluster = require('cluster'),
  numCPUs = require('os').cpus().length,
  express = require('express');

if (cluster.isMaster) {
  for (let i = 0; i < numCPUs; i++)
    cluster.fork();
} else {
  const app = module.exports = express();
  app.get('/plaintext', (req, res) =>
    res
      .header('Content-Type', 'text/plain')
      .send('Hello, World!'));
}
```

```
import { Router } from '@vertx/web';

const app = Router.router(vertx);

app.get("/plaintext").handler(ctx => {
  ctx.response()
    .putHeader("Content-Type", "text/plain")
    .end('Hello, World!');
});
```

Showing **196** of **198** frameworks.

Show filters panel

Test types

JSON serialization | Single query | Multiple queries | Fortunes | Data updates | Plaintext

Hardware

Physical | Cloud

Single query

Best (bar chart) | Data table | Latency | Framework overhead

Best database-access responses per second, single query, Test environment (398 tests)

Rnk	Framework	Best performance (higher is better)		Errors	Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA
1	wizzardo-http	843,545	100.0%	0	Mcr	Jav	Non	Non	Lin	Pg	Lin	Raw	Rea
2	vertx-postgres	728,681	86.4%	0	Plt	Jav	ver	Non	Lin	Pg	Lin	Raw	Rea
3	actix-raw	683,025	81.0%	0	Plt	Rus	Non	act	Lin	Pg	Lin	Raw	Rea
4	actix-pg	629,423	74.6%	0	Mcr	Rus	Non	act	Lin	Pg	Lin	Raw	Rea
5	vertx-web-postgres	629,402	74.6%	0	Mcr	Jav	vtx	Non	Lin	Pg	Lin	Raw	Rea
6	es4x	613,564	72.7%	0	Mcr	JS	ver	Non	Lin	Pg	Lin	Raw	Rea
7	h2o	562,260	66.7%	0	Plt	C	Non	Non	Lin	Pg	Lin	Raw	Rea
8	drogon-raw	559,356	66.3%	0	Ful	C++	Non	Non	Lin	Pg	Lin	Raw	Rea
9	greenlightning	518,358	61.4%	0	Mcr	Jav	Non	Non	Lin	Pg	Lin	Raw	Rea
10	revenj-jvm	480,646	57.0%	0	Ful	Jav	Svt	Res	Lin	Pg	Lin	Ful	Rea
11	cpoll_cppsp-postgres-raw-threadpool	479,920	56.9%	0	Plt	C++	Non	Non	Lin	Pg	Lin	Raw	Rea
12	ulib-postgres	468,484	55.5%	0	Plt	C++	Non	ULi	Lin	Pg	Lin	Mcr	Rea

Multiple queries

Responses per second at 20 queries per request, Test environment (405 tests)

Rnk	Framework	Performance (higher is better)		Errors	Cls	Lng	Plt
1	actix-pg	46,303	100.0%	0	Mcr	Rus	Non
2	greenlightning	45,348	97.9%	0	Mcr	Jav	Non
3	actix-core	45,002	97.2%	0	Plt	Rus	Non
4	vertx-postgres	43,494	93.9%	0	Plt	Jav	ver
5	wizzardo-http	43,448	93.8%	0	Mcr	Jav	Non
6	es4x	42,974	92.8%	0	Mcr	JS	ver
7	ratpack-pgclient	42,972	92.8%	0	Mcr	Jav	Nty
8	vertx-web-postgres	41,576	89.8%	0	Mcr	Jav	vtx
9	micronaut	33,685	72.7%	0	Mcr	Jav	Nty
10	h2o	27,936	60.3%	0	Plt	C	Non
11	http4k-undertow	27,696	59.8%	0	Mcr	Kot	Utw
12	hexagon-jetty-postgresql	27,624	59.7%	0	Mcr	Kot	Svt
13	actix-diesel	27,594	59.6%	0	Mcr	Rus	Non
14	drogon-raw	27,581	59.6%	0	Ful	C++	Non

Polyglot GraalVM is fast.

what about latency?

Data updates

Latency of 20-update responses, Test environment

Framework	Average latency (lower is better)	σ (SD)	Max	Errors
es4x	26.5 ms 6.4%	11.3 ms	203.8 ms	0
vertx-web-postgres	30.1 ms 7.3%	17.1 ms	90.5 ms	0
vertx-postgres	40.9 ms 9.9%	25.2 ms	521.0 ms	0
undertow-jersey-hikaricp	119.0 ms 28.8%	19.8 ms	320.8 ms	434
nodejs	214.5 ms 51.9%	52.4 ms	405.0 ms	0
nodejs-postgres	267.9 ms 64.8%	35.7 ms	564.7 ms	0

Requirements summary

This test exercises database writes. Each request is processed by fetching multiple rows from a simple database table, converting the rows to in-memory objects, modifying one attribute of each object in memory, updating each associated row in the database individually, and then serializing the list of objects as a JSON response. The test is run multiple times: testing 1, 5, 10, 15, and 20 updates per request. Note that the number of **statements** per request is twice the number of updates since each update is paired with one query to fetch the object. All tests are run at 512 concurrency.

The response is analogous to the multiple-query test. Example response for 10 updates:

```
HTTP/1.1 200 OK
Content-Length: 315
Content-Type: application/json
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT
```

Conclusion

- latency is not a problem, it's a symptom
- use non-blocking to fully use the CPU
- **use vert.x ;-)**
- polyglot is fine
- use graalvm for polyglot JIT optimizations
- node can be slow for server applications
- **use ES4X ;-)**

Let's connect!

- @pml0pes
- pmlopes
- https://www.linkedin.com/in/pmlopes/
- https://reactiverse.io/es4x
- https://vertx.io
- https://graalvm.org

THANKS!