

Crushing Latency with Vert.x

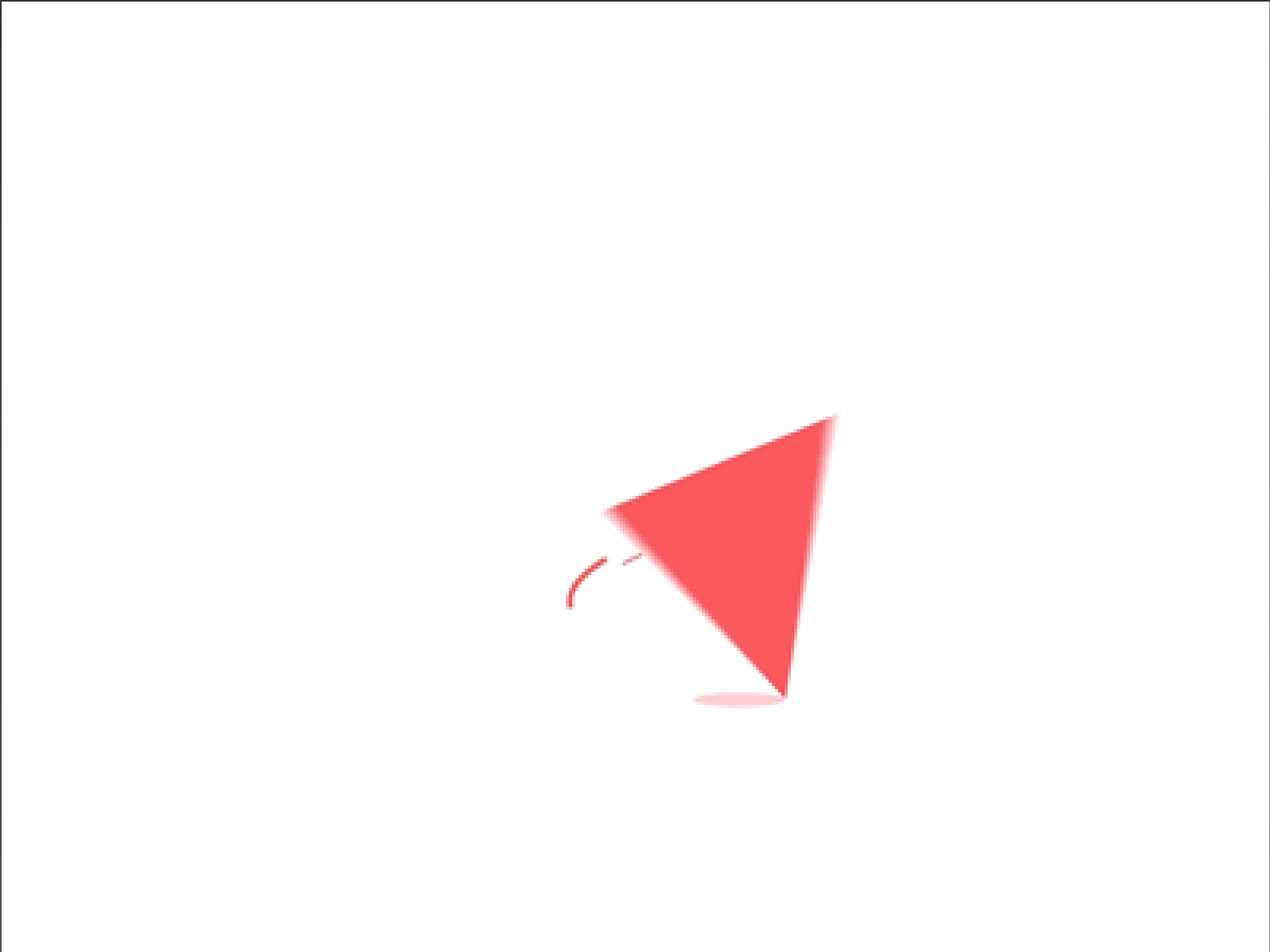
Paulo Lopes 🧢

Principal Software Engineer

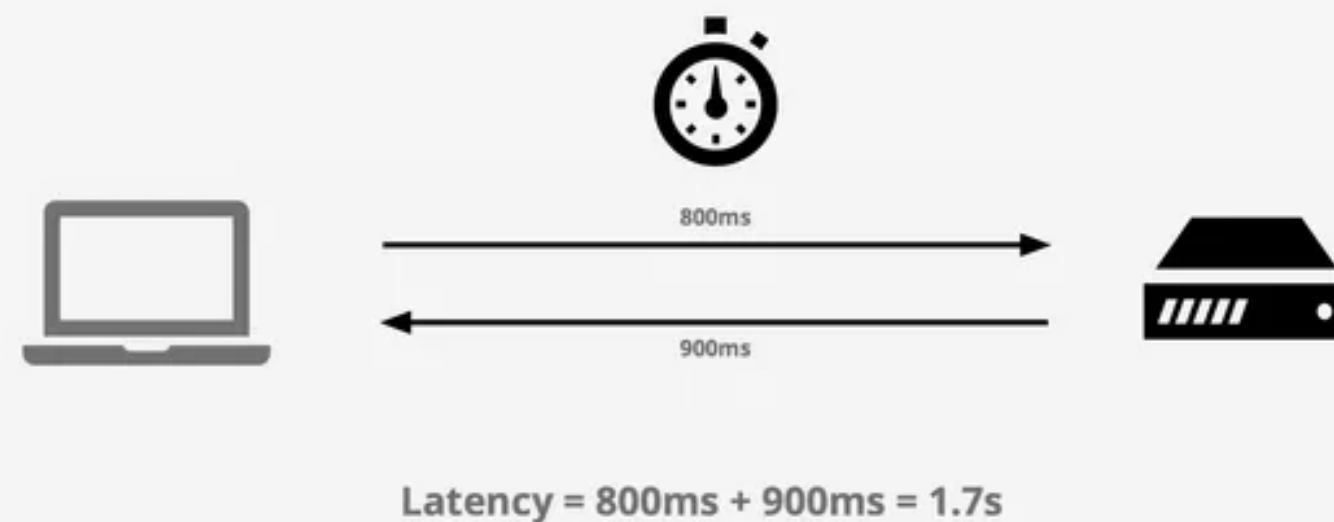
🐦 @pm10pes

🌐 <https://www.linkedin.com/in/pm10pes/>

🐙 pm10pes



latency ^{noun}
la·ten·cy | \ 'lā-tən(t)-sf \



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
- **Google:** an extra 0.5 seconds in search page generation time dropped traffic by 20%
- **A broker:** could lose \$4 million in revenues per millisecond if their electronic trading platform is 5 milliseconds behind the competition

<http://home.blarg.net/~glinden/StanfordDataMining.2006-11-29.ppt>

<http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html>

<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

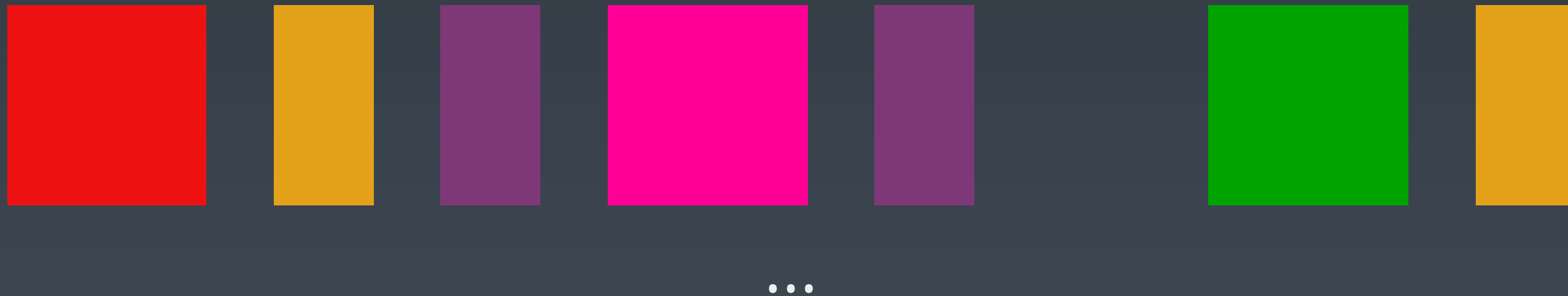
req time grows as threads fight for execution time



```
# 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)
```

**DID YOU
KNOW?**

```
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: $(1000 / 200 - 1) * 5 = 0 \sim 20\text{ms}$

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

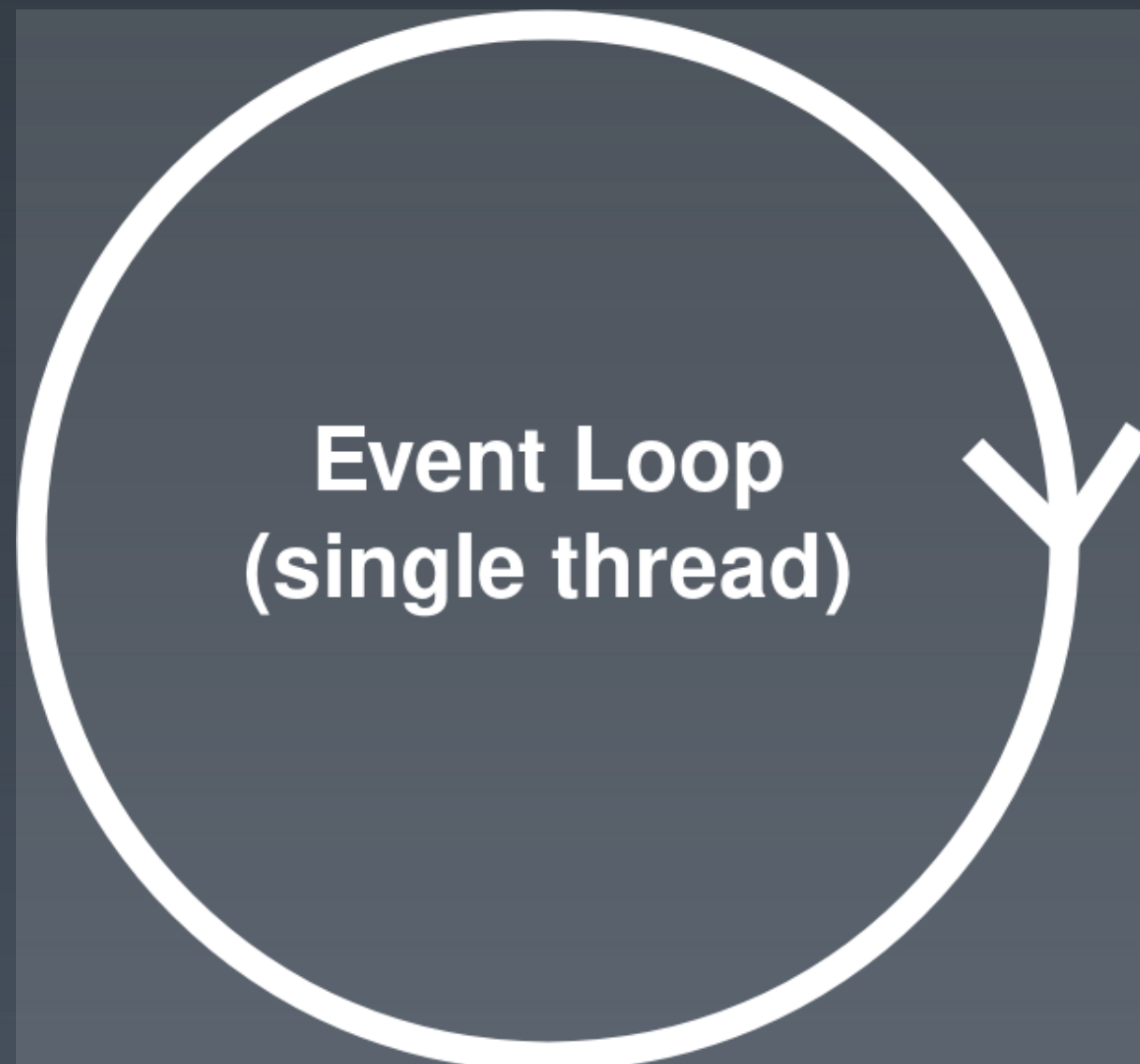
at max utilization

CPU is mostly waiting

Non-Blocking I/O

Vert.x

Events



Request handler



AUTH handler



DB handler



JSON handler

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


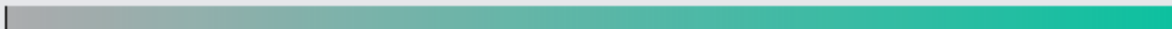









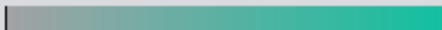








20-queries (bar)

Data table

Latency

Framework overhead

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
```

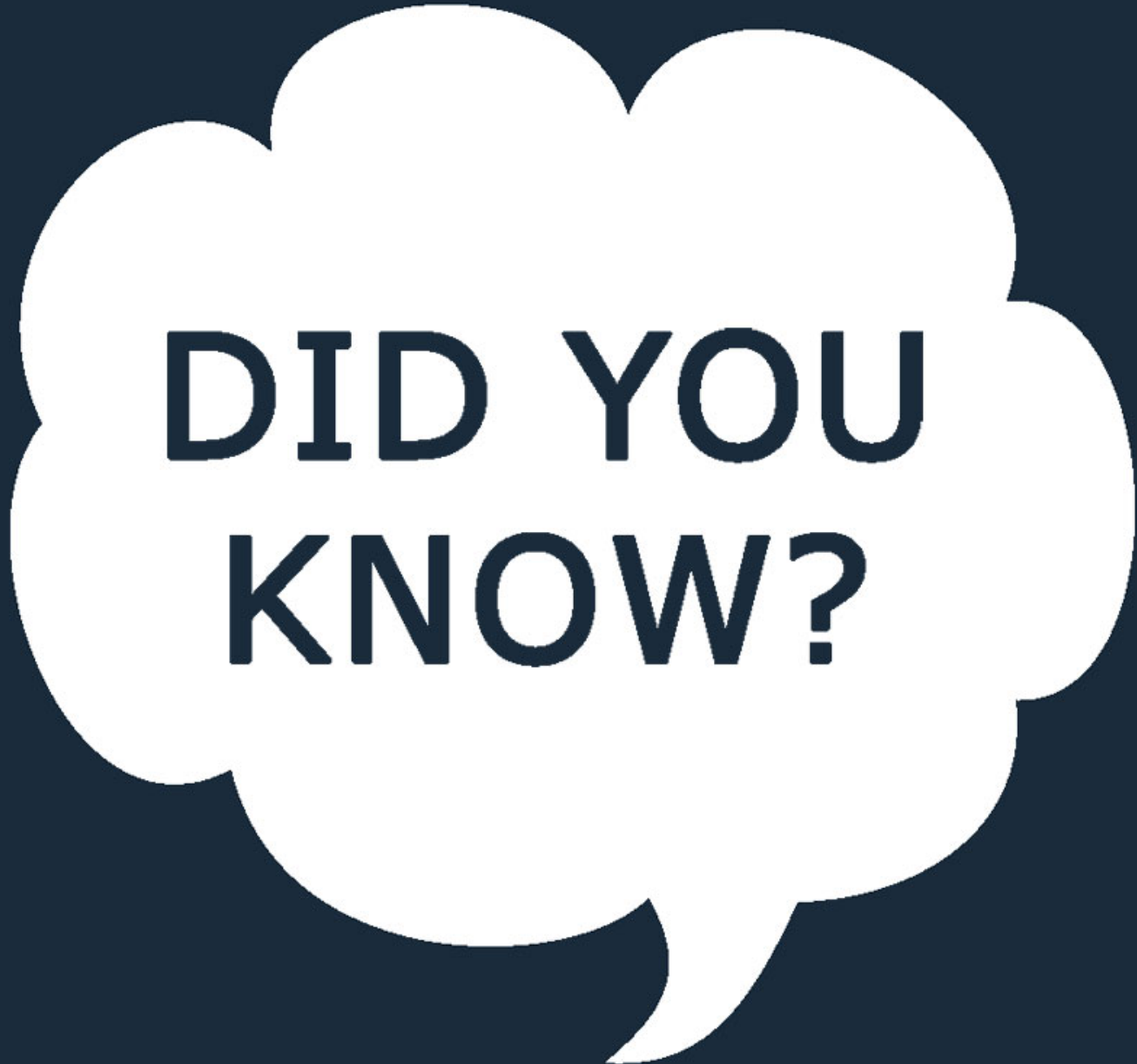
```
[{"id":4174,"randomNumber":331}, {"id":51,"randomNumber":6544}, {"id":4462,"randomNumber":952}, {"id":2221,"randomNumber":532}, {"id":9276,"randomNumber":3097}, {"id":3056,"randomNumber":7293}, {"id":6964,"randomNumber":620}, {"id":675,"randomNumber":6601}, {"id":8414,"randomNumber":6569}, {"id":2753,"randomNumber":4065}]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

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');  
}
```

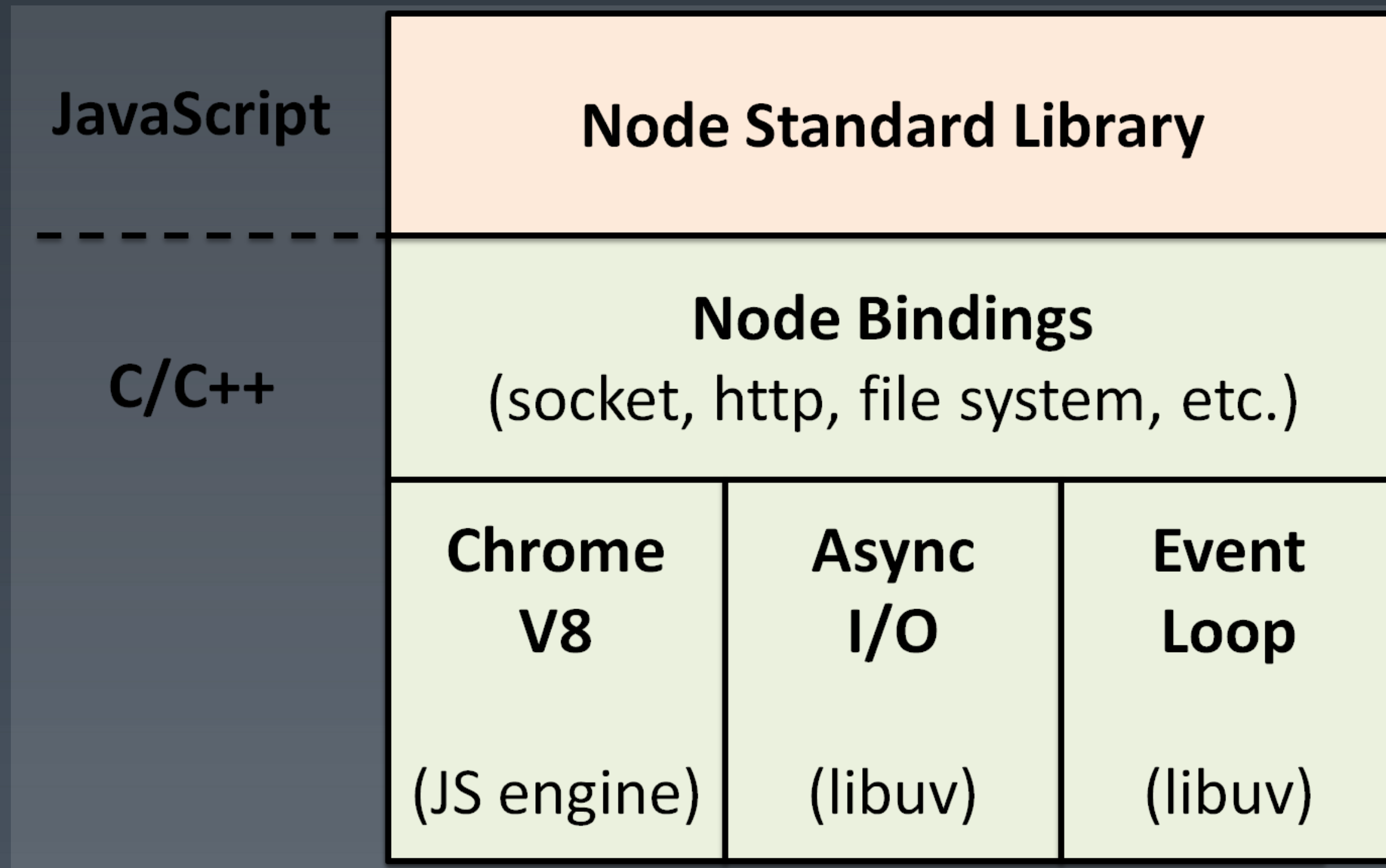
Getting the response object

```
1  this.response = function() {
2    var __args = arguments;
3    if (__args.length === 0) {
4      if (that.cachedresponse == null) {
5        that.cachedresponse = utils.convReturnVertxGen(
6          HttpServerResponse,
7          j_routingContext["response()"]());
8      }
9      return that.cachedresponse;
10   } else if (typeof __super_response !== 'undefined') {
11     return __super_response.apply(this, __args);
12   }
13   else throw new TypeError('invalid arguments');
14 };
```

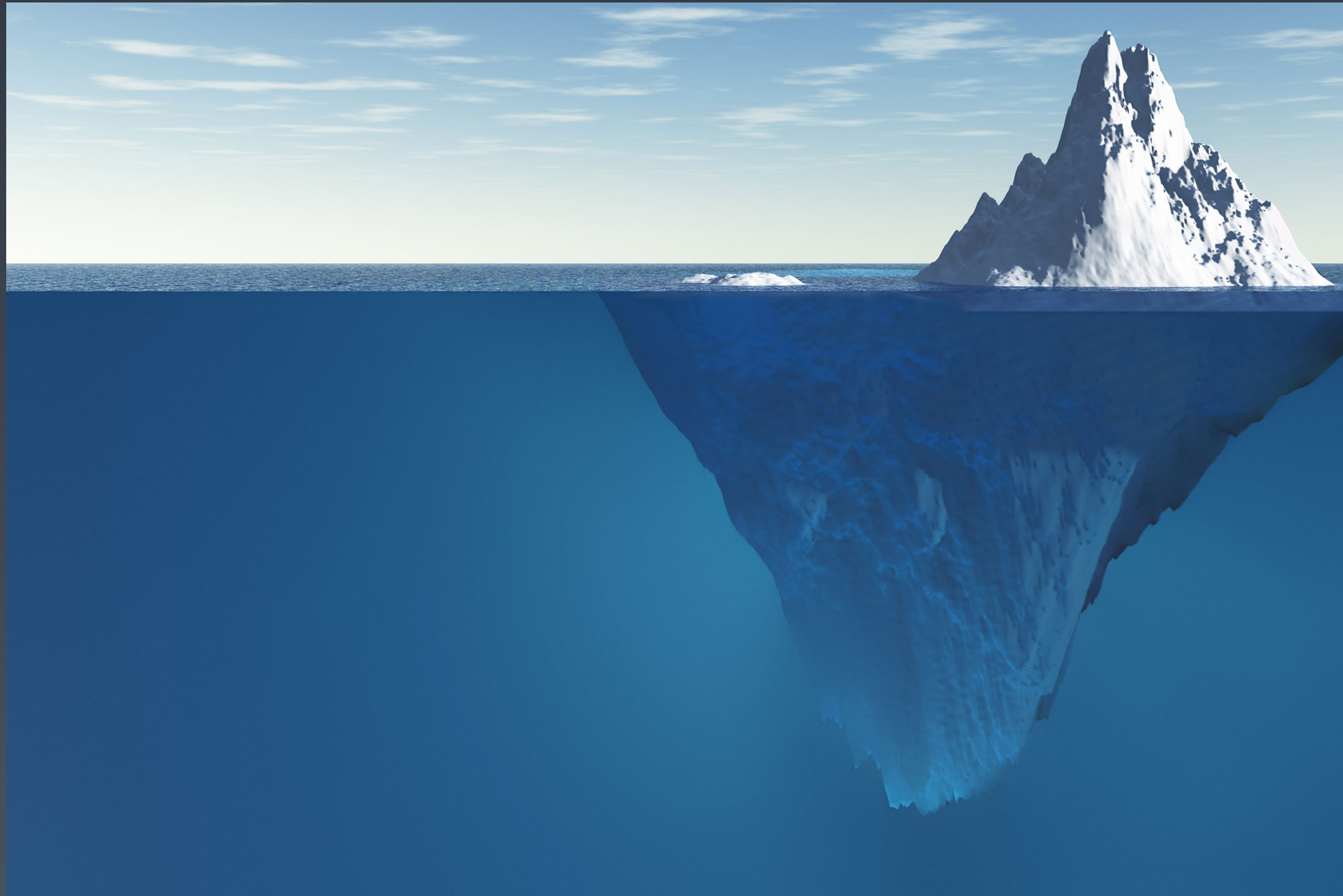
In a nutshell

- 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



JIT can't optimize it all



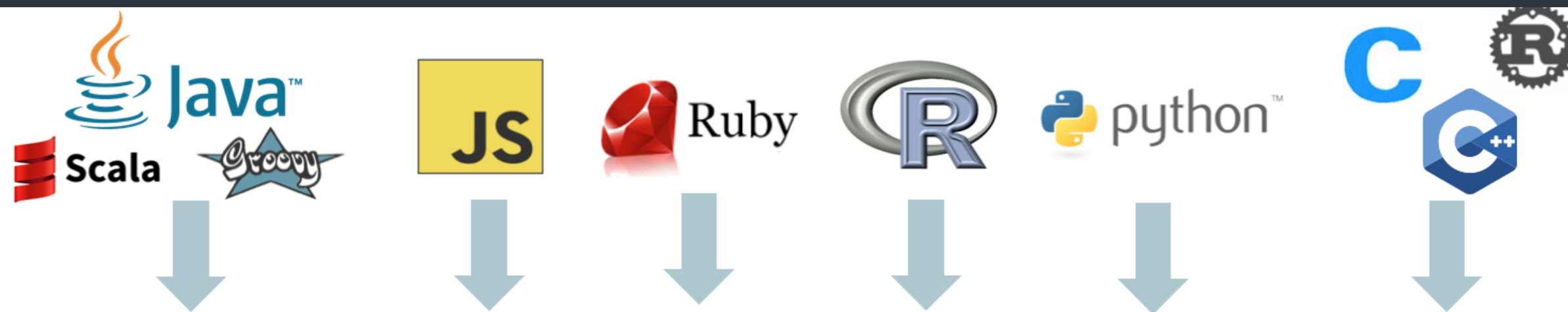
Where's Node?

Best (bar chart)		Data table	Latency	Framework overhead									
Best database-access responses per second, single query, Test environment (398 tests)													
Rank	Framework	Best performance (higher is better)	Errors	Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA	
1	wizardo-http	843,545	<div><div></div></div> 100.0%	0	Mer	Jaw	Non	Non	Lin	Pg	Lin	Raw	Rea
2	vertx-postgres	728,681	<div><div></div></div> 86.4%	0	Ptl	Jaw	ver	Non	Lin	Pg	Lin	Raw	Rea
3	actix-raw	683,025	<div><div></div></div> 81.0%	0	Ptl	Rus	Non	act	Lin	Pg	Lin	Raw	Rea
4	actix-pg	629,423	<div><div></div></div> 74.6%	0	Mer	Lin	Non	act	Lin	Pg	Lin	Raw	Rea
5	vertx-web-postgres	629,402	<div><div></div></div> 74.6%	0	Mer	Jaw	vtx	Non	Lin	Pg	Lin	Raw	Rea
6	es4x	613,564	<div><div></div></div> 72.7%	0	Mer	JS	ver	Non	Lin	Pg	Lin	Raw	Rea
7	h2o	562,260	<div><div></div></div> 66.7%	0	Ptl	C	Non	Non	Lin	Pg	Lin	Raw	Rea
8	drogon-raw	559,356	<div><div></div></div> 66.3%	0	Ful	C++	Non	Non	Lin	Pg	Lin	Raw	Rea
9	greenlightning	518,358	<div><div></div></div> 61.4%	0	Mer	Jaw	Non	Non	Lin	Pg	Lin	Raw	Rea
10	reventi-jvm	480,646	<div><div></div></div> 57.0%	0	Ful	Jaw	Svt	Res	Lin	Pg	Lin	Ful	Rea
11	cpool_cppsp-postgres-raw-threadpool	479,920	<div><div></div></div> 56.9%	0	Ptl	C++	Non	Non	Lin	Pg	Lin	Raw	Rea
12	ulib-postgres	468,484	<div><div></div></div> 55.5%	0	Ptl	C++	Non	ULI	Lin	Pg	Lin	Mcrl	Rea
13	cpool_cppsp-raw	450,749	<div><div></div></div> 53.4%	0	Ptl	C++	Non	Non	Lin	My	Lin	Raw	Rea
14	fasthttp-postgresql-prefork	431,732	<div><div></div></div> 51.2%	0	Ptl	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
15	gemini-postgres	427,197	<div><div></div></div> 50.6%	0	Ful	Jaw	Svt	Res	Lin	Pg	Lin	Mcrl	Rea
16	cpool_cppsp-postgres-raw	426,057	<div><div></div></div> 50.5%	0	Ptl	C++	Non	Non	Lin	Pg	Lin	Raw	Rea
17	swoole	421,262	<div><div></div></div> 49.9%	0	Ptl	PHP	Non	svw	Lin	My	Lin	Raw	Rea
18	proteus	416,329	<div><div></div></div> 49.4%	0	Mer	Jaw	Utw	Non	Lin	Pg	Lin	Raw	Rea
19	fasthttp-prefork	413,519	<div><div></div></div> 49.0%	120	Ptl	Go	Non	Non	Lin	My	Lin	Raw	Rea
20	fasthttp-postgresql	409,504	<div><div></div></div> 48.5%	0	Ptl	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
21	jooby2-undertow	404,192	<div><div></div></div> 47.9%	0	Ful	Jaw	Utw	Non	Lin	Pg	Lin	Raw	Rea
22	http4k-apache	400,068	<div><div></div></div> 47.4%	0	Mer	Kot	apa	Non	Lin	Pg	Lin	Raw	Rea
23	jooby2	388,615	<div><div></div></div> 46.1%	0	Ful	Jaw	Nty	Non	Lin	Pg	Lin	Raw	Rea
24	ratpack-pgclient	383,049	<div><div></div></div> 45.4%	0	Mer	Jaw	Nty	Non	Lin	Pg	Lin	Raw	Rea
25	ratpack-jdbc	373,879	<div><div></div></div> 44.3%	0	Mer	Jaw	Nty	Non	Lin	Pg	Lin	Raw	Rea
26	light-4j	372,095	<div><div></div></div> 44.1%	0	Ptl	Jaw	lig	Non	Lin	Pg	Lin	Raw	Rea
27	aspcore-ado-pg	360,710	<div><div></div></div> 42.8%	0	Ptl	C#	.NE	ies	Lin	Pg	Lin	Raw	Rea
28	undertow-postgresql	348,529	<div><div></div></div> 41.3%	0	Ptl	Jaw	Utw	Non	Lin	Pg	Lin	Raw	Rea
29	jooby2-jetty	328,200	<div><div></div></div> 38.9%	0	Ful	Jaw	Jty	Non	Lin	Pg	Lin	Raw	Rea
30	act-eclipselink-pgsql	325,162	<div><div></div></div> 38.5%	0	Ful	Jaw	Utw	Non	Lin	Pg	Lin	Ful	Rea
31	ulib-postgres_fit	324,497	<div><div></div></div> 38.5%	0	Ptl	C++	Non	ULI	Lin	Pg	Lin	Mcrl	Rea
32	openresty	319,146	<div><div></div></div> 37.8%	0	Ptl	Lua	Op8	ngx	Lin	My	Lin	Raw	Rea
33	chi-sjson-prefork	312,375	<div><div></div></div> 37.0%	0	Mer	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
34	aspcore-rhxx-pg	310,495	<div><div></div></div> 36.8%	0	Ptl	C#	.NE	ies	Lin	Pg	Lin	Raw	Rea
35	chi-gojay-prefork	310,466	<div><div></div></div> 36.8%	0	Mer	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
36	chi-prefork	304,241	<div><div></div></div> 36.1%	0	Mer	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
37	ulib-mysql	302,741	<div><div></div></div> 35.9%	0	Ptl	C++	Non	ULI	Lin	My	Lin	Mcrl	Rea
38	chi	300,992	<div><div></div></div> 35.7%	0	Mer	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
39	proteus-mysql	299,388	<div><div></div></div> 35.5%	0	Mer	Jaw	Utw	Non	Lin	My	Lin	Raw	Rea
40	act-hibernate-pgsql	298,570	<div><div></div></div> 35.4%	0	Ful	Jaw	Utw	Non	Lin	Pg	Lin	Ful	Rea
41	http4k-undertow	298,428	<div><div></div></div> 35.4%	0	Mer	Kot	Utw	Non	Lin	Pg	Lin	Raw	Rea
42	gemini-mysql	291,947	<div><div></div></div> 34.6%	0	Ful	Jaw	Svt	Res	Lin	My	Lin	Mcrl	Rea
43	aspcore-vb-mw-ado-pg	291,563	<div><div></div></div> 34.6%	0	Mer	Vb	.NE	ies	Lin	Pg	Lin	Raw	Rea
44	jawn	290,337	<div><div></div></div> 34.4%	0	Ful	Jaw	Svt	Utw	Lin	Pg	Lin	Raw	Rea
45	chi-sjson	288,315	<div><div></div></div> 34.2%	0	Mer	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
46	aspcore-mw-ado-pg	287,755	<div><div></div></div> 34.1%	0	Mer	C#	.NE	ies	Lin	Pg	Lin	Raw	Rea
47	chi-gojay	285,930	<div><div></div></div> 33.9%	0	Mer	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
48	micronaut	276,034	<div><div></div></div> 32.7%	0	Mer	Jaw	Nty	Non	Lin	Pg	Lin	Raw	Rea
49	http4k	275,879	<div><div></div></div> 32.7%	0	Mer	Kot	Svt	Non	Lin	Pg	Lin	Raw	Rea
50	act-hibernate-mysql	272,742	<div><div></div></div> 32.3%	0	Ful	Jaw	Utw	Non	Lin	My	Lin	Ful	Rea
51	drogon	270,351	<div><div></div></div> 32.0%	0	Ful	C++	Non	Non	Lin	Pg	Lin	Mcrl	Rea
52	aspcore-mw-dap-pg	268,028	<div><div></div></div> 31.8%	0	Mer	C#	.NE	ies	Lin	Pg	Lin	Mcrl	Rea
53	jooby	266,161	<div><div></div></div> 31.6%	0	Ful	Jaw	Nty	Non	Lin	Pg	Lin	Raw	Rea
54	iris-prefork	248,574	<div><div></div></div> 29.5%	0	Ful	Jaw	Nty	Non	Lin	Pg	Lin	Raw	Rea
55	go-my-prefork	246,297	<div><div></div></div> 29.2%	0	Ptl	Go	Non	Non	Lin	My	Lin	Raw	Rea
56	cutelyst-pf-pg-nodelay	242,019	<div><div></div></div> 28.7%	0	Ful	C++	cut	Non	Lin	Pg	Lin	Raw	Rea
57	cutelyst-pf-pg	241,564	<div><div></div></div> 28.6%	0	Ful	C++	cut	Non	Lin	Pg	Lin	Raw	Rea
58	aspcore-ado-my	240,405	<div><div></div></div> 28.5%	0	Ptl	C#	.NE	ies	Lin	My	Lin	Raw	Rea
59	cutelyst-thread-pg-nodelay	238,182	<div><div></div></div> 28.2%	0	Ful	C++	cut	Non	Lin	Pg	Lin	Raw	Rea
60	cutelyst-thread-pg	238,078	<div><div></div></div> 28.2%	0	Ful	C++	cut	Non	Lin	Pg	Lin	Raw	Rea
61	ulib-mongodb	237,399	<div><div></div></div> 28.1%	0	Ptl	C++	Non	ULI	Lin	Mo	Lin	Mcrl	Rea
62	fasthttp	235,047	<div><div></div></div> 27.9%	0	Ptl	Go	Non	Non	Lin	My	Lin	Raw	Rea
63	go-pg-prefork	234,714	<div><div></div></div> 27.8%	0	Ptl	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
64	crystal	233,197	<div><div></div></div> 27.6%	0	Ptl	Cry	Non	Non	Lin	Pg	Lin	Raw	Rea
65	crystal-radix	232,810	<div><div></div></div> 27.6%	0	Ptl	Cry	Non	Non	Lin	Pg	Lin	Raw	Rea
66	raze	232,550	<div><div></div></div> 27.6%	0	Ful	Cry	Non	Non	Lin	Pg	Lin	Mcrl	Rea
67	kemal	231,082	<div><div></div></div> 27.4%	0	Ful	Cry	Non	Non	Lin	Pg	Lin	Mcrl	Rea
68	hexagon-jetty-postgresql	226,597	<div><div></div></div> 26.9%	0	Mer	Kot	Svt	Non	Lin	Pg	Lin	Raw	Rea
69	aah-postgresql	222,048	<div><div></div></div> 26.3%	0	Ful	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
70	vt-postgres	215,181	<div><div></div></div> 25.5%	0	Ful	C++	Non	Non	Lin	Pg	Lin	Ful	Rea
71	act-morphia-mongo	213,961	<div><div></div></div> 25.4%	0	Ful	Jaw	Utw	Non	Lin	Mo	Lin	Ful	Rea
72	compojure-raw	213,285	<div><div></div></div> 25.3%	0	Mer	CJ	Svt	Res	Lin	My	Lin	Raw	Rea
73	go-pg	211,968	<div><div></div></div> 25.1%	0	Ptl	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
74	act-eclipselink-mysql	211,636	<div><div></div></div> 25.1%	0	Ful	Jaw	Utw	Non	Lin	My	Lin	Ful	Rea
75	aspcore-mw-ado-my	205,909	<div><div></div></div> 24.4%	0	Mer	C#	.NE	ies	Lin	My	Lin	Raw	Rea
76	go-my	204,688	<div><div></div></div> 24.3%	0	Ptl	Go	Non	Non	Lin	My	Lin	Raw	Rea
77	aspcore-vb-mw-ado-my	202,585	<div><div></div></div> 24.0%	0	Mer	Vb	.NE	ies	Lin	My	Lin	Raw	Rea
78	falcone	202,444	<div><div></div></div> 24.0%	0	Mer	Go	Non	Non	Lin	My	Lin	Raw	Rea
79	spark	200,785	<div><div></div></div> 23.8%	0	Mer	Jaw	Svt	Jty	Lin	My	Lin	Ful	Rea
80	goji	199,910	<div><div></div></div> 23.7%	0	Mer	Go	Non	Non	Lin	My	Lin	Ful	Rea
81	servilet-postgresql	199,733	<div><div></div></div> 23.7%	0	Ptl	Jaw	Svt	Res	Lin	Pg	Lin	Raw	Rea
82	silicon	197,081	<div><div></div></div> 23.4%	0	Mer	C++	Non	milc	Lin	My	Lin	Ful	Rea
83	Introspect	197,059	<div><div></div></div> 23.4%	0	Mer	Sca	Nty	Non	Lin	My	Lin	Raw	Rea
84	echo	196,727	<div><div></div></div> 23.3%	0	Mer	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
85	aspcore-mw-dap-my	196,616	<div><div></div></div> 23.3%	0	Mer	C#	.NE	ies	Lin	My	Lin	Mcrl	Rea
86	gin	192,491	<div><div></div></div> 22.8%	0	Mer	Go	Non	Go	Lin	My	Lin	Raw	Rea
87	iris	191,953	<div><div></div></div> 22.8%	0	Ful	Go	Non	Non	Lin	Pg	Lin	Raw	Rea
88	akka-http	191,737	<div><div></div></div> 22.7%	0	Mer	Sca	Aldk	Non	Lin	My	Lin	Raw	Rea
89	nodejs-mongodb-raw	191,600	<div><div></div></div> 22.7%	0	Ptl	JS	nlp	Non	Lin	Mo	Lin	Raw	Rea
90	minijax	188,098	<div><div></div></div> 22.3%	0	Ful	Jaw	JAX	Utw	Lin	My	Lin	Ful	Rea
91	workerman	186,361											

Rnk	Framework	Best performance (higher is better)	Errors	Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA
1	<div><div></div>wizzardo-http</div>	843,545	<div><div></div>100.0%</div>	0	Mcr	Jav	Non	Non	Lin	Pg	Lin	Raw
2	<div><div></div>vertx-postgres</div>	728,681	<div><div></div>86.4%</div>	0	Plt	Jav	ver	Non	Lin	Pg	Lin	Raw
3	<div><div></div>actix-raw</div>	683,025	<div><div></div>81.0%</div>	0	Plt	Rus	Non	act	Lin	Pg	Lin	Raw
4	<div><div></div>actix-pg</div>	629,423	<div><div></div>74.6%</div>	0	Mcr	Rus	Non	act	Lin	Pg	Lin	Raw
5	<div><div></div>vertx-web-postgres</div>	629,402	<div><div></div>74.6%</div>	0	Mcr	Jav	vtx	Non	Lin	Pg	Lin	Raw
6	<div><div></div>es4x</div>	613,564	<div><div></div>72.7%</div>	0	Mcr	JS	ver	Non	Lin	Pg	Lin	Raw
7	<div><div></div>h2o</div>	562,260	<div><div></div>66.7%</div>	0	Plt	C	Non	Non	Lin	Pg	Lin	Raw
8	<div><div></div>dragon-raw</div>	559,356	<div><div></div>66.3%</div>	0	Ful	C++	Non	Non	Lin	Pg	Lin	Raw
9	<div><div></div>greenlightning</div>	518,358	<div><div></div>61.4%</div>	0	Mcr	Jav	Non	Non	Lin	Pg	Lin	Raw
10	<div><div></div>revenj-jvm</div>	480,646	<div><div></div>57.0%</div>	0	Ful	Jav	Svt	Res	Lin	Pg	Lin	Ful
11	<div><div></div>cpoll_cppsp-postgres-raw-threadpool</div>	479,920	<div><div></div>56.9%</div>	0	Plt	C++	Non	Non	Lin	Pg	Lin	Raw
12	<div><div></div>ulib-postgres</div>	468,484	<div><div></div>55.5%</div>	0	Plt	C++	Non	ULI	Lin	Pg	Lin	Mcr
13	<div><div></div>cpoll_cppsp-raw</div>	450,749	<div><div></div>53.4%</div>	0	Plt	C++	Non	Non	Lin	My	Lin	Raw
14	<div><div></div>fasthttp-postgresql-prefork</div>	431,732	<div><div></div>51.2%</div>	0	Plt	Go	Non	Non	Lin	Pg	Lin	Raw
15	<div><div></div>gemini-postgres</div>	427,197	<div><div></div>50.6%</div>	0	Ful	Jav	Svt	Res	Lin	Pg	Lin	Mcr
16	<div><div></div>cpoll_cppsp-postgres-raw</div>	426,057	<div><div></div>50.5%</div>	0	Plt	C++	Non	Non	Lin	Pg	Lin	Raw
17	<div><div></div>swoole</div>	421,262	<div><div></div>49.9%</div>	0	Plt	PHP	Non	swo	Lin	My	Lin	Raw
18	<div><div></div>proteus</div>	416,329	<div><div></div>49.4%</div>	0	Mcr	Jav	Utw	Non	Lin	Pg	Lin	Raw
19	<div><div></div>fasthttp-prefork</div>	413,519	<div><div></div>49.0%</div>	120	Plt	Go	Non	Non	Lin	My	Lin	Raw
20	<div><div></div>fasthttp-postgresql</div>	409,504	<div><div></div>48.5%</div>	0	Plt	Go	Non	Non	Lin	Pg	Lin	Raw
21	<div><div></div>jooby2-undertow</div>	404,192	<div><div></div>47.9%</div>	0	Ful	Jav	Utw	Non	Lin	Pg	Lin	Raw
22	<div><div></div>http4k-apache</div>	400,068	<div><div></div>47.4%</div>	0	Mcr	Kot	apa	Non	Lin	Pg	Lin	Raw
23	<div><div></div>jooby2</div>	388,615	<div><div></div>46.1%</div>	0	Ful	Jav	Nty	Non	Lin	Pg	Lin	Raw
24	<div><div></div>ratpack-pgclient</div>	383,049	<div><div></div>45.4%</div>	0	Mcr	Jav	Nty	Non	Lin	Pg	Lin	Raw
25	<div><div></div>ratpack-jdbc</div>	373,879	<div><div></div>44.3%</div>	0	Mcr	Jav	Nty	Non	Lin	Pg	Lin	Raw
26	<div><div></div>light-4j</div>	372,095	<div><div></div>44.1%</div>	0	Plt	Jav	lig	Non	Lin	Pg	Lin	Raw
27	<div><div></div>aspcore-ado-pg</div>	360,710	<div><div></div>42.8%</div>	0	Plt	C#	.NE	kes	Lin	Pg	Lin	Raw
28	<div><div></div>undertow-postgresql</div>	348,529	<div><div></div>41.3%</div>	0	Plt	Jav	Utw	Non	Lin	Pg	Lin	Raw
29	<div><div></div>jooby2-jetty</div>	328,200	<div><div></div>38.9%</div>	0	Ful	Jav	Jty	Non	Lin	Pg	Lin	Raw
30	<div><div></div>act-eclipselink-pgsql</div>	325,162	<div><div></div>38.5%</div>	0	Ful	Jav	Utw	Non	Lin	Pg	Lin	Ful
31	<div><div></div>ulib-postgres_fit</div>	324,497	<div><div></div>38.5%</div>	0	Plt	C++	Non	ULI	Lin	Pg	Lin	Mcr
32	<div><div></div>openresty</div>	319,146	<div><div></div>37.8%</div>	0	Plt	Lua	OpR	ngx	Lin	My	Lin	Raw
33	<div><div></div>chi-sjson-prefork</div>	312,375	<div><div></div>37.0%</div>	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw
34	<div><div></div>aspcore-rhtx-pg</div>	310,495	<div><div></div>36.8%</div>	0	Plt	C#	.NE	kes	Lin	Pg	Lin	Raw
35	<div><div></div>chi-gojay-prefork</div>	310,466	<div><div></div>36.8%</div>	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw
36	<div><div></div>chi-prefork</div>	304,241	<div><div></div>36.1%</div>	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw
37	<div><div></div>ulib-mysql</div>	302,741	<div><div></div>35.9%</div>	0	Plt	C++	Non	ULI	Lin	My	Lin	Mcr
38	<div><div></div>chi</div>	300,992	<div><div></div>35.7%</div>	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw
39	<div><div></div>proteus-mysql</div>	299,388	<div><div></div>35.5%</div>	0	Mcr	Jav	Utw	Non	Lin	My	Lin	Raw
40	<div><div></div>act-hibernate-pgsql</div>	298,570	<div><div></div>35.4%</div>	0	Ful	Jav	Utw	Non	Lin	Pg	Lin	Ful
41	<div><div></div>http4k-undertow</div>	298,428	<div><div></div>35.4%</div>	0	Mcr	Kot	Utw	Non	Lin	Pg	Lin	Raw
42	<div><div></div>gemini-mysql</div>	291,947	<div><div></div>34.6%</div>	0	Ful	Jav	Svt	Res	Lin	My	Lin	Mcr
43	<div><div></div>aspcore-vb-mw-ado-pg</div>	291,563	<div><div></div>34.6%</div>	0	Mcr	vb	.NE	kes	Lin	Pg	Lin	Raw
44	<div><div></div>jawn</div>	290,337	<div><div></div>34.4%</div>	0	Ful	Jav	Svt	Utw	Lin	Pg	Lin	Raw
45	<div><div></div>chi-sjson</div>	288,315	<div><div></div>34.2%</div>	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw
46	<div><div></div>aspcore-mw-ado-pg</div>	287,755	<div><div></div>34.1%</div>	0	Mcr	C#	.NE	kes	Lin	Pg	Lin	Raw
47	<div><div></div>chi-gojay</div>	285,930	<div><div></div>33.9%</div>	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw
48	<div><div></div>micronaut</div>	276,034	<div><div></div>32.7%</div>	0	Ful	Jav	Svt	Non	Lin	Pg	Lin	Raw
49	<div><div></div>http4k</div>	275,879	<div><div></div>32.7%</div>	0	Mcr	Kot	Svt	Non	Lin	Pg	Lin	Raw
50	<div><div></div>act-hibernate-mysql</div>	272,742	<div><div></div>32.3%</div>	0	Ful	Jav	Utw	Non	Lin	My	Lin	Ful
51	<div><div></div>dragon</div>	270,351	<div><div></div>32.0%</div>	0	Ful	C++	Non	Non	Lin	Pg	Lin	Raw
52	<div><div></div>aspcore-mw-dap-pg</div>	268,028	<div><div></div>31.8%</div>	0	Mcr	C#	.NE	kes	Lin	Pg	Lin	Mcr
53	<div><div></div>jooby</div>	266,161	<div><div></div>31.6%</div>	0	Ful	Jav	Nty	Non	Lin	Pg	Lin	Raw
54	<div><div></div>iris-prefork</div>	248,574	<div><div></div>29.5%</div>	0	Ful	Go	Non	Non	Lin	Pg	Lin	Raw
55	<div><div></div>go-my-prefork</div>	246,297	<div><div></div>29.2%</div>	0	Ful	Go	Non	Non	Lin	Pg	Lin	Raw
56	<div><div></div>cutelyst-pf-pg-nodelay</div>	242,019	<div><div></div>28.7%</div>	0	Plt	C++	cut	Non	Lin	My	Lin	Raw
57	<div><div></div>cutelyst-pf-pg</div>	241,564	<div><div></div>28.6%</div>	0	Ful	C++	cut	Non	Lin	Pg	Lin	Raw
58	<div><div></div>aspcore-ado-my</div>	240,405	<div><div></div>28.5%</div>	0	Plt	C#	.NE	kes	Lin	My	Lin	Raw
59	<div><div></div>cutelyst-thread-pg-nodelay</div>	238,182	<div><div></div>28.2%</div>	0	Ful	C++	cut	Non	Lin	Pg	Lin	Raw
60	<div><div></div>cutelyst-thread-pg</div>	238,078	<div><div></div>28.2%</div>	0	Ful	C++	cut	Non	Lin	Pg	Lin	Raw
61	<div><div></div>ulib-mongodb</div>	237,399	<div><div></div>28.1%</div>	0	Plt	C++	cut	Non	Lin	Mo	Lin	Mcr
62	<div><div></div>fasthttp</div>	235,047	<div><div></div>27.9%</div>	0	Plt	Go	Non	Non	Lin	My	Lin	Raw
63	<div><div></div>go-pg-prefork</div>	234,714	<div><div></div>27.8%</div>	0	Plt	Go	Non	Non	Lin	Pg	Lin	Raw
64	<div><div></div>crystal</div>	233,197	<div><div></div>27.6%</div>	0	Ful	Cry	cut	Non	Lin	Pg	Lin	Raw
65	<div><div></div>crystal-radix</div>	232,810	<div><div></div>27.6%</div>	0	Plt	Cry	Non	Non	Lin	Pg	Lin	Raw
66	<div><div></div>raze</div>	232,550	<div><div></div>27.6%</div>	0	Ful	Cry	Non	Non	Lin	Pg	Lin	Raw
67	<div><div></div>kemal</div>	231,082	<div><div></div>27.4%</div>	0	Ful	Cry	Non	Non	Lin	Pg	Lin	Raw
68	<div><div></div>hexagon-jetty-postgresql</div>	226,597	<div><div></div>26.9%</div>	0	Mcr	Kot	Svt	Non	Lin	Pg	Lin	Raw
69	<div><div></div>aah-postgresql</div>	222,048	<div><div></div>26.3%</div>	0	Ful	Go	Non	Non	Lin	Pg	Lin	Raw
70	<div><div></div>wt-postgres</div>	215,181	<div><div></div>25.5%</div>	0	Ful	C++	Non	Non	Lin	Pg	Lin	Raw
71	<div><div></div>act-morphia-mongo</div>	213,961	<div><div></div>25.4%</div>	0	Ful	C++	Non	Non	Lin	Pg	Lin	Raw
72	<div><div></div>compojure-raw</div>	213,285	<div><div></div>25.3%</div>	0	Mcr	Clj	Svt	Res	Lin	My	Lin	Raw
73	<div><div></div>go-pg</div>	211,978	<div><div></div>25.1%</div>	0	Plt	Go	Non	Non	Lin	My	Lin	Raw
74	<div><div></div>act-eclipselink-mysql</div>	211,366	<div><div></div>25.1%</div>	0	Plt	Go	Non	Non	Lin	Pg	Lin	Raw
75	<div><div></div>aspcore-mw-ado-my</div>	207,909	<div><div></div>24.4%</div>	0	Mcr	C#	.NE	kes	Lin	My	Lin	Raw
76	<div><div></div>go-my</div>	204,688	<div><div></div>24.3%</div>	0	Plt	Go	Non	Non	Lin	My	Lin	Raw
77	<div><div></div>aspcore-vb-mw-ado-my</div>	202,585	<div><div></div>24.0%</div>	0	Mcr	vb	.NE	kes	Lin	My	Lin	Raw
78	<div><div></div>falcone</div>	202,444	<div><div></div>24.0%</div>	0	Mcr	Go	Non	Non	Lin	My	Lin	Raw
79	<div><div></div>spark</div>	200,785	<div><div></div>23.8%</div>	0	Mcr	Jav	Svt	Jty	Lin	My	Lin	Ful
80	<div><div></div>goji</div>	199,910	<div><div></div>23.7%</div>	0	Mcr	Go	Non	Non	Lin	My	Lin	Raw
81	<div><div></div>servlet-postgresql</div>	199,733	<div><div></div>23.7%</div>	0	Plt	Jav	Svt	Res	Lin	Pg	Lin	Raw
82	<div><div></div>silicon</div>	197,081	<div><div></div>23.4%</div>	0	Mcr	C++	Non	mic	Lin	My	Lin	Ful
83	<div><div></div>finintrospect</div>	197,059	<div><div></div>23.4%</div>	0	Mcr	Sca	Nty	Non	Lin	My	Lin	Raw
84	<div><div></div>echo</div>	196,727	<div><div></div>23.3%</div>	0	Mcr	Go	Non	Non	Lin	Pg	Lin	Raw
85	<div><div></div>aspcore-mw-dap-my</div>	196,616	<div><div></div>23.3%</div>	0	Mcr	C#	.NE	kes	Lin	My	Lin	Mcr
86	<div><div></div>gin</div>	192,491	<div><div></div>22.8%</div>	0	Mcr	Go	Non	Non	Lin	My	Lin	Raw
87	<div><div></div>iris</div>	191,953	<div><div></div>22.8%</div>	0	Ful	Go	Non	Non	Lin	Pg	Lin	Raw
88	<div><div></div>akka-http</div>	191,737	<div><div></div>22.7%</div>	0	Mcr	Sca	Altk	Non	Lin	My	Lin	Raw
89	<div><div></div>nodejs-mongodb-raw</div>	191,600	<div><div></div>22.7%</div>	0	Plt	JS	njs	Non	Lin	Mo	Lin	Raw
90	<div><div></div>minijax</div>	188,098	<div><div></div>22.3%</div>	0	Ful	Jav	JAX	Utw	Lin	My	Lin	Ful
91	<div><div></div>workerman</div>	186,366	<div><div></div>22.1%</div>	0	Plt	PHP	Non	por	Lin	My	Lin	Raw



Can we make polyglot fast?



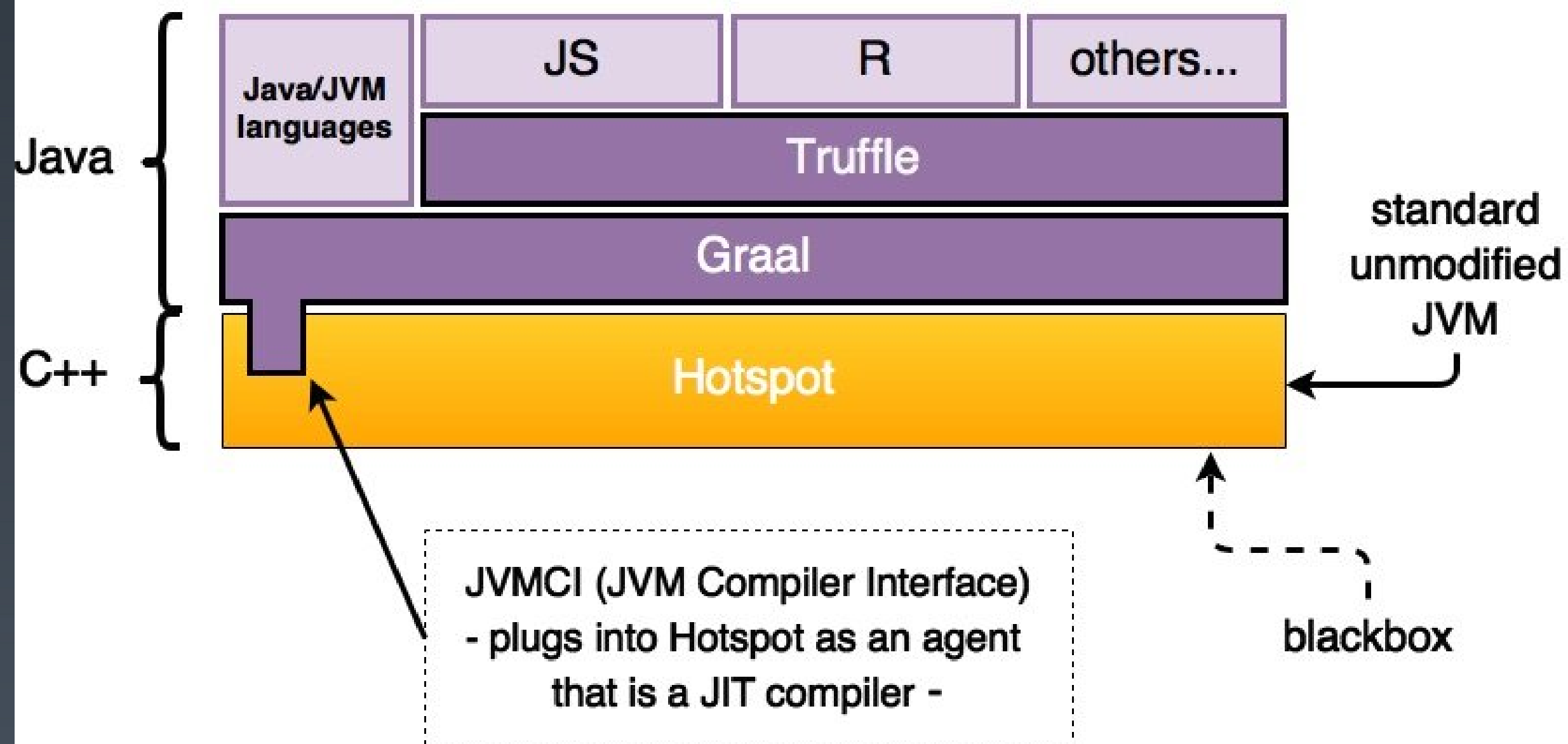
Automatic transform of interpreters to compiler

GraalVM™

Engine integration native and managed



<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

Hardware

JSON serialization

Single query

Multiple queries

Fortunes

Data updates

Plaintext

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 <div><div></div></div> 100.0%	0	Mcr	Jav	Non	Non	Lin	Pg	Lin	Raw	Rea
2	 vertx-postgres	728,681 <div><div></div></div> 86.4%	0	Plt	Jav	ver	Non	Lin	Pg	Lin	Raw	Rea
3	 actix-raw	683,025 <div><div></div></div> 81.0%	0	Plt	Rus	Non	act	Lin	Pg	Lin	Raw	Rea
4	 actix-pg	629,423 <div><div></div></div> 74.6%	0	Mcr	Rus	Non	act	Lin	Pg	Lin	Raw	Rea
5	 vertx-web-postgres	629,402 <div><div></div></div> 74.6%	0	Mcr	Jav	vtx	Non	Lin	Pg	Lin	Raw	Rea
6	 es4x	613,564 <div><div></div></div> 72.7%	0	Mcr	JS	ver	Non	Lin	Pg	Lin	Raw	Rea
7	 h2o	562,260 <div><div></div></div> 66.7%	0	Plt	C	Non	Non	Lin	Pg	Lin	Raw	Rea
8	 drogon-raw	559,356 <div><div></div></div> 66.3%	0	Ful	C++	Non	Non	Lin	Pg	Lin	Raw	Rea
9	 greenlightning	518,358 <div><div></div></div> 61.4%	0	Mcr	Jav	Non	Non	Lin	Pg	Lin	Raw	Rea
10	 revenj-jvm	480,646 <div><div></div></div> 57.0%	0	Ful	Jav	Svt	Res	Lin	Pg	Lin	Ful	Rea
11	 cpoll_cppsp-postgres-raw-threadpool	479,920 <div><div></div></div> 56.9%	0	Plt	C++	Non	Non	Lin	Pg	Lin	Raw	Rea
12	 ulib-postgres	468,484 <div><div></div></div> 55.5%	0	Plt	C++	Non	ULi	Lin	Pg	Lin	Mcr	Rea
13	spell-annon-raw	450,740 <div><div></div></div> 53.4%	0	Plt	C++	Non	Non	Lin	My	Lin	Raw	Rea

20-queries (bar)		Data table		Latency		Framework overhead					
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	<div><div></div></div>	100.0%	0	Mcr	Rus	Non			
2	 greenlightning	45,348	<div><div></div></div>	97.9%	0	Mcr	Jav	Non			
3	 actix-core	45,002	<div><div></div></div>	97.2%	0	Plt	Rus	Non			
4	 vertx-postgres	43,494	<div><div></div></div>	93.9%	0	Plt	Jav	ver			
5	 wizzardo-http	43,448	<div><div></div></div>	93.8%	0	Mcr	Jav	Non			
6	 es4x	42,974	<div><div></div></div>	92.8%	0	Mcr	JS	ver			
7	 ratpack-pgclient	42,972	<div><div></div></div>	92.8%	0	Mcr	Jav	Nty			
8	 vertx-web-postgres	41,576	<div><div></div></div>	89.8%	0	Mcr	Jav	vtx			
9	 micronaut	33,685	<div><div></div></div>	72.7%	0	Mcr	Jav	Nty			
10	 h2o	27,936	<div><div></div></div>	60.3%	0	Plt	C	Non			
11	 http4k-undertow	27,696	<div><div></div></div>	59.8%	0	Mcr	Kot	Utw			
12	 hexagon-jetty-postgresql	27,624	<div><div></div></div>	59.7%	0	Mcr	Kot	Svt			
13	 actix-diesel	27,594	<div><div></div></div>	59.6%	0	Mcr	Rus	Non			
14	 dragon-raw	27,581	<div><div></div></div>	59.6%	0	Ful	C++	Non			

Polyglot GraalVM is fast.

what about latency?

Data updates













20-updates (bar)

Data table

Latency

Framework overhead

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!

-  @pm10pes
-  pm10pes
-  <https://www.linkedin.com/in/pm10pes/>
- <https://reactiverse.io/es4x>
- <https://vertx.io>
- <https://graalvm.org>

THANKS! | QCon th