#ibminterconnect

Java vs JavaScript for Enterprise Web Applications

Chris Bailey: STSM, IBM Runtime Monitoring

IBM

InterConnect2015

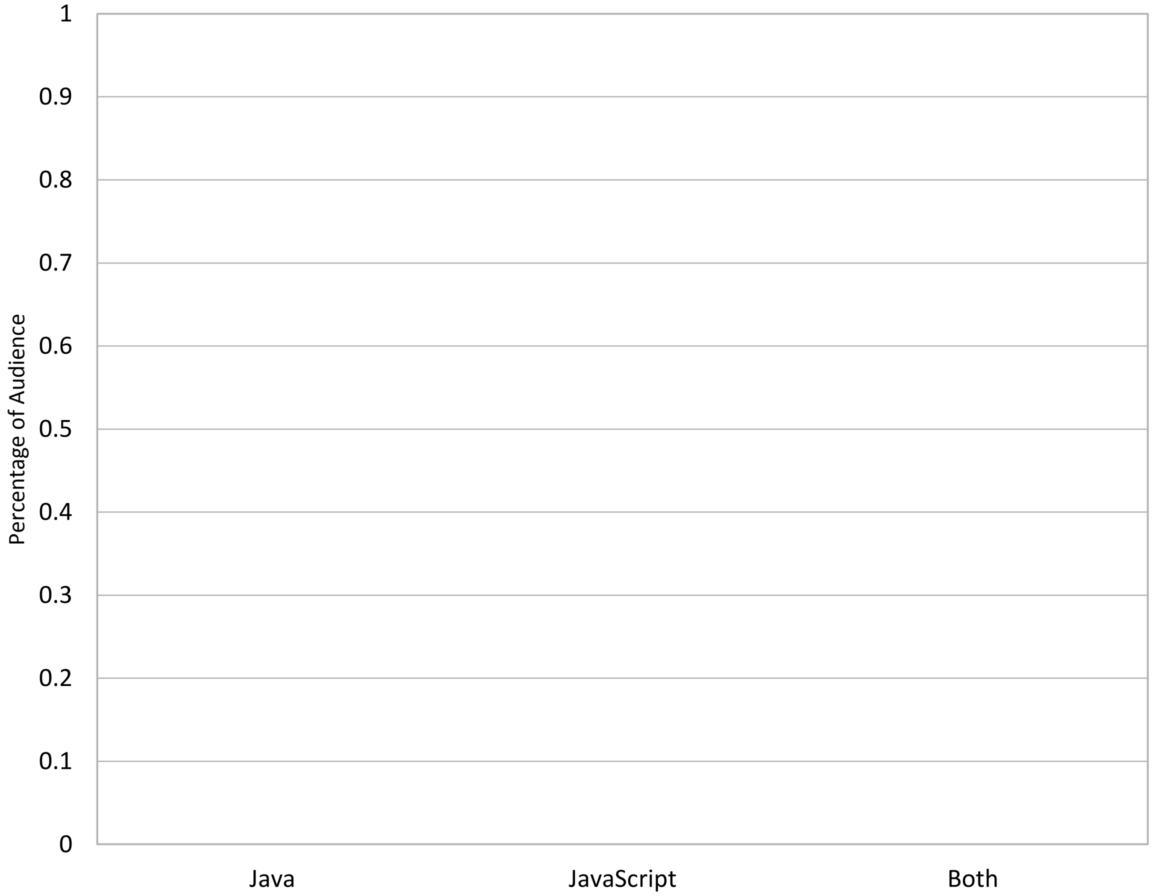
The Premier Cloud & Mobile Conference

February 22 – 26

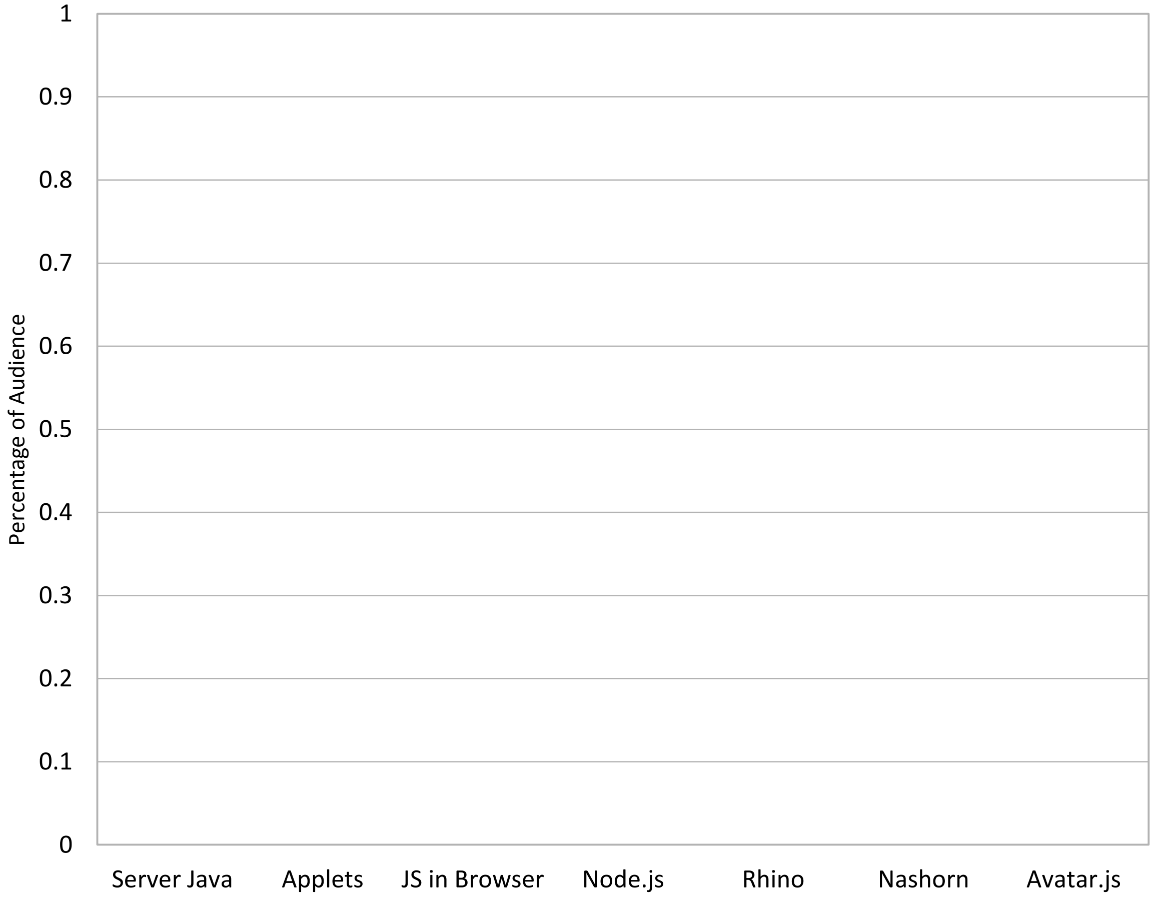
MGM Grand & Mandalay Bay | Las Vegas, Nevada

A Quick Survey

What languages do you use?



What runtimes do you use for them?

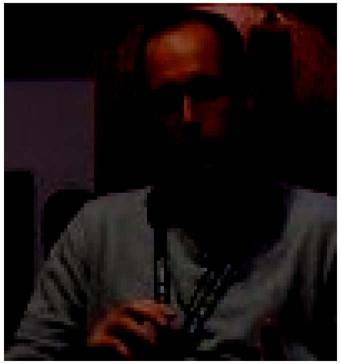


Introduction to the Speakers

- **Chris Bailey**

STSM, IBM Runtime Monitoring and Diagnostics Architect

-14 years working with Java and JVM technologies
-1 year working with Node.js and V8
-6 months working with Ruby and Python



- Recent work focus:

-Java monitoring, diagnostics and troubleshooting
-Java integration into the cloud
-JavaScript monitoring, diagnostics and troubleshooting

- My contact information:

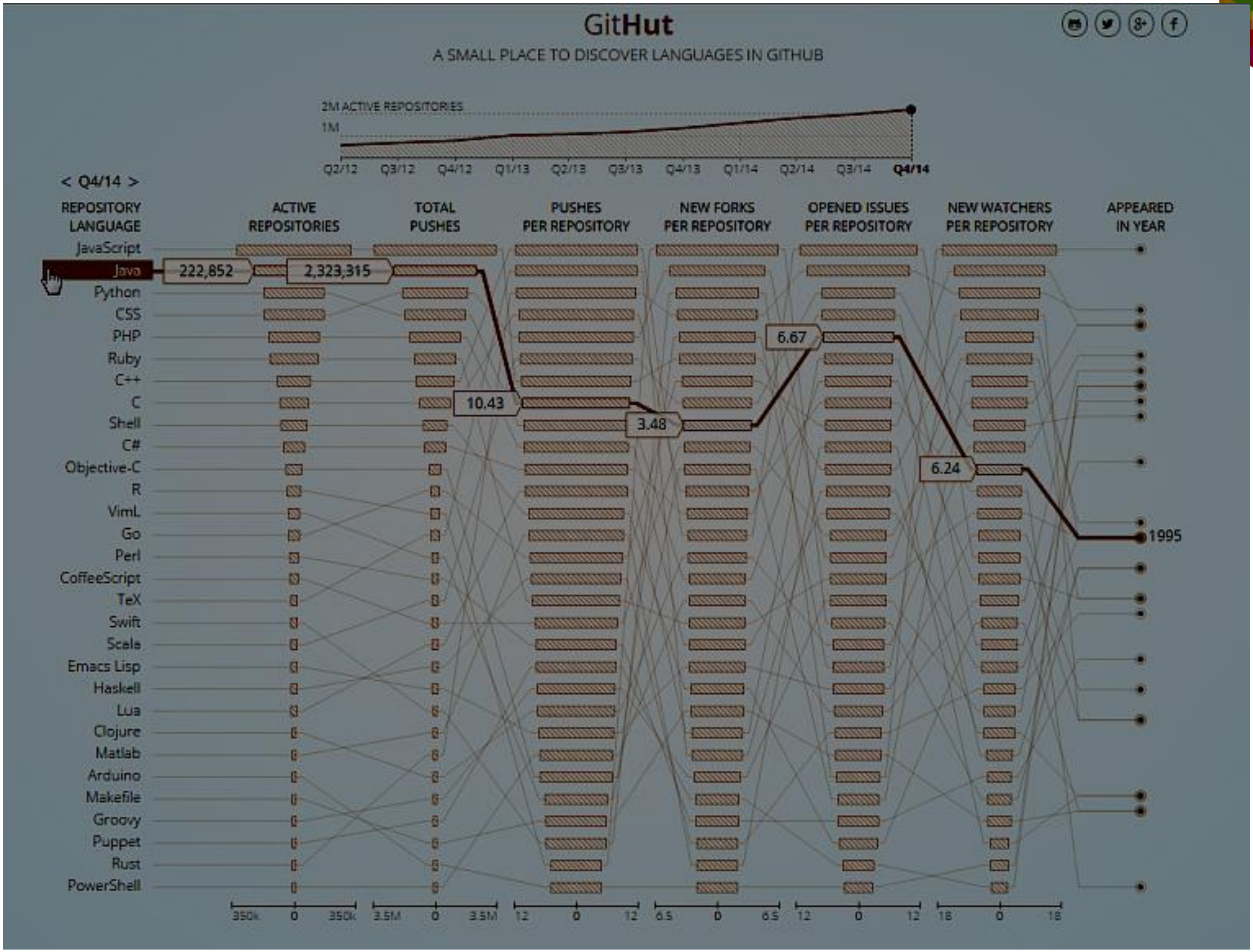
-baileyc@uk.ibm.com
-http://www.linkedin.com/in/chrisbaileyibm
-http://www.slideshare.net/cnbailey/
-@Chris__Bailey

Agenda

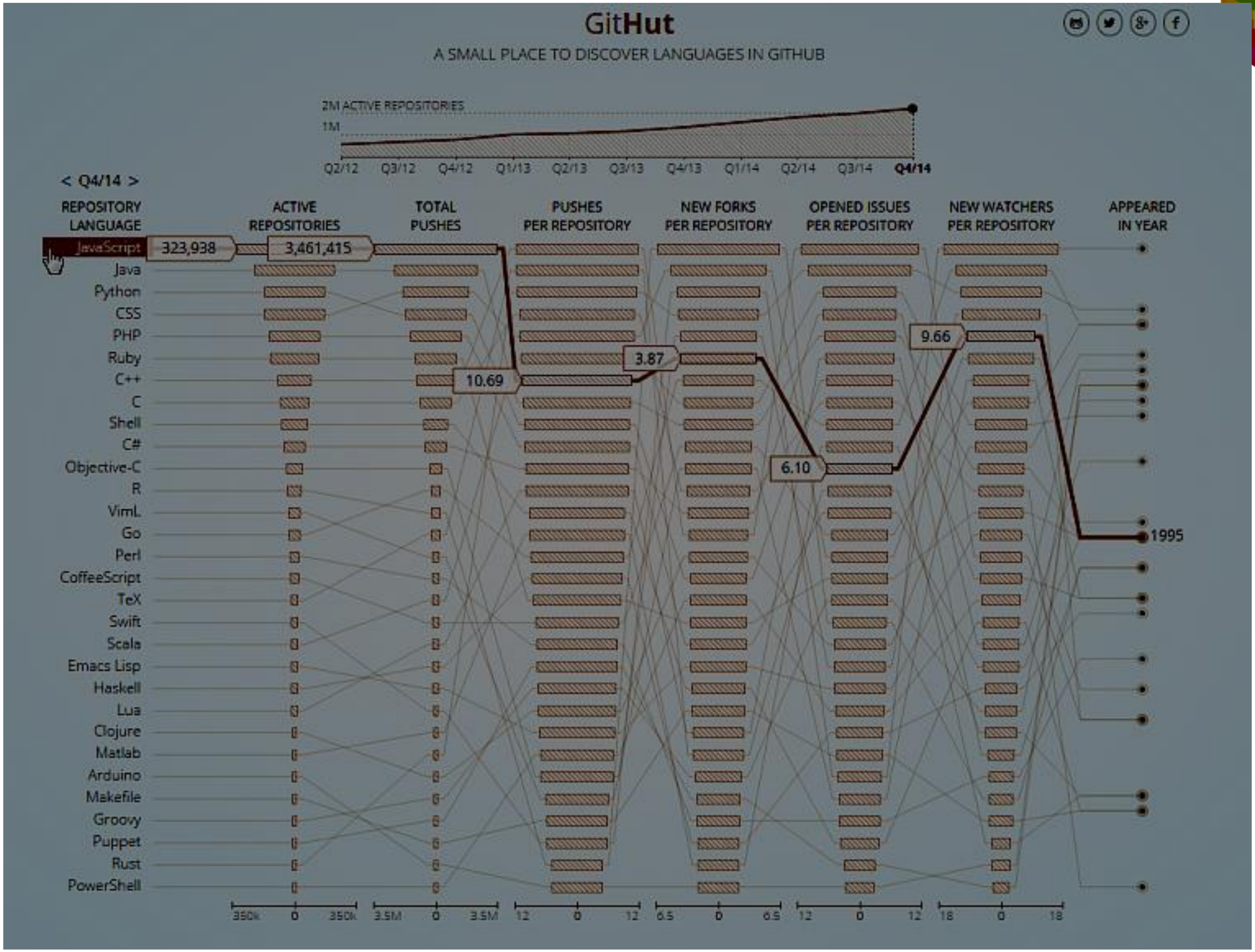
- Language Adoption
- Deployment Modes
- Asynchronous IO
- WebApplication Performance
- Under the Hood
- Enterprise Deployments
- IBM and Node.js

Language Adoption

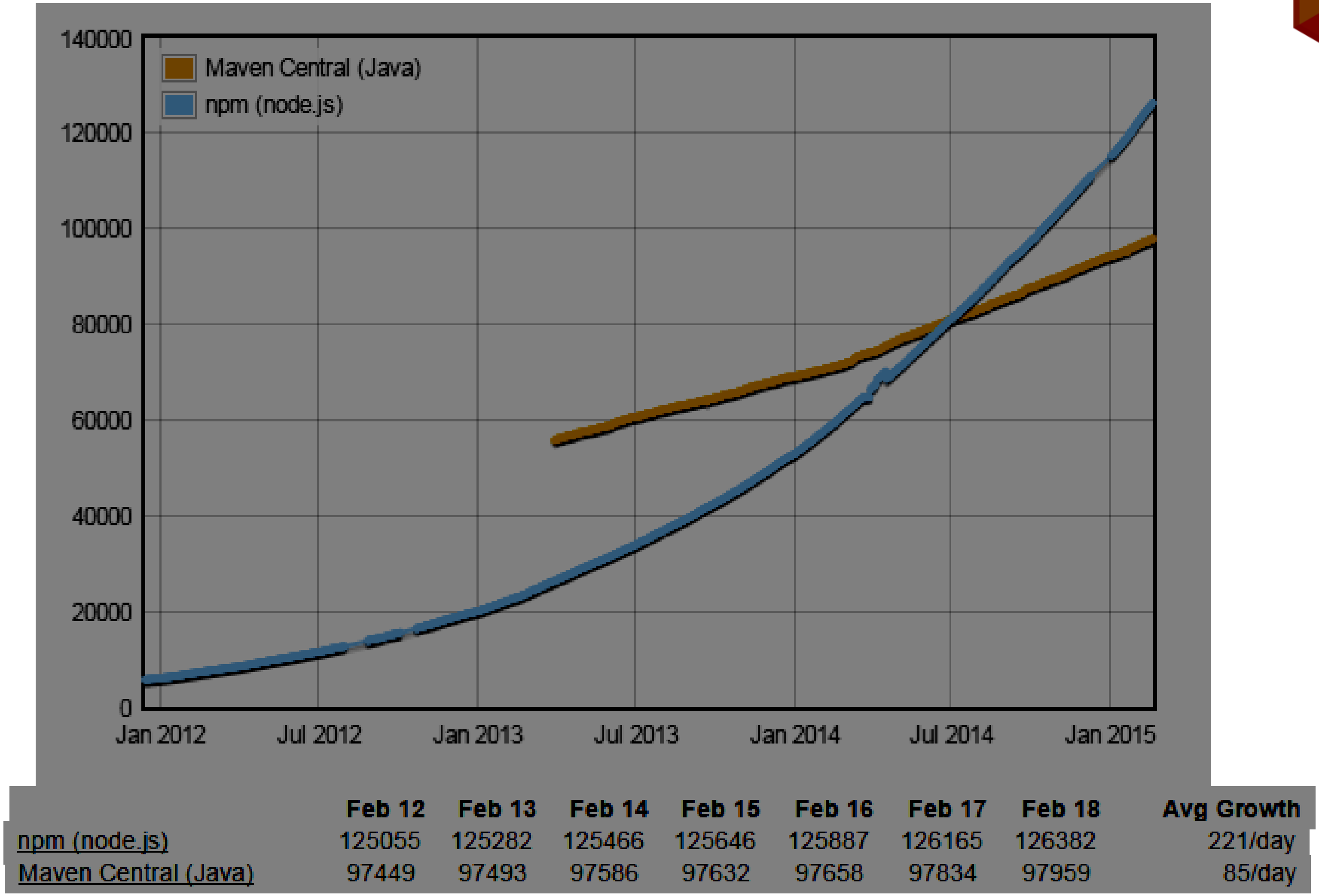
GitHub Adoption: Java



GitHub Adoption: JavaScript

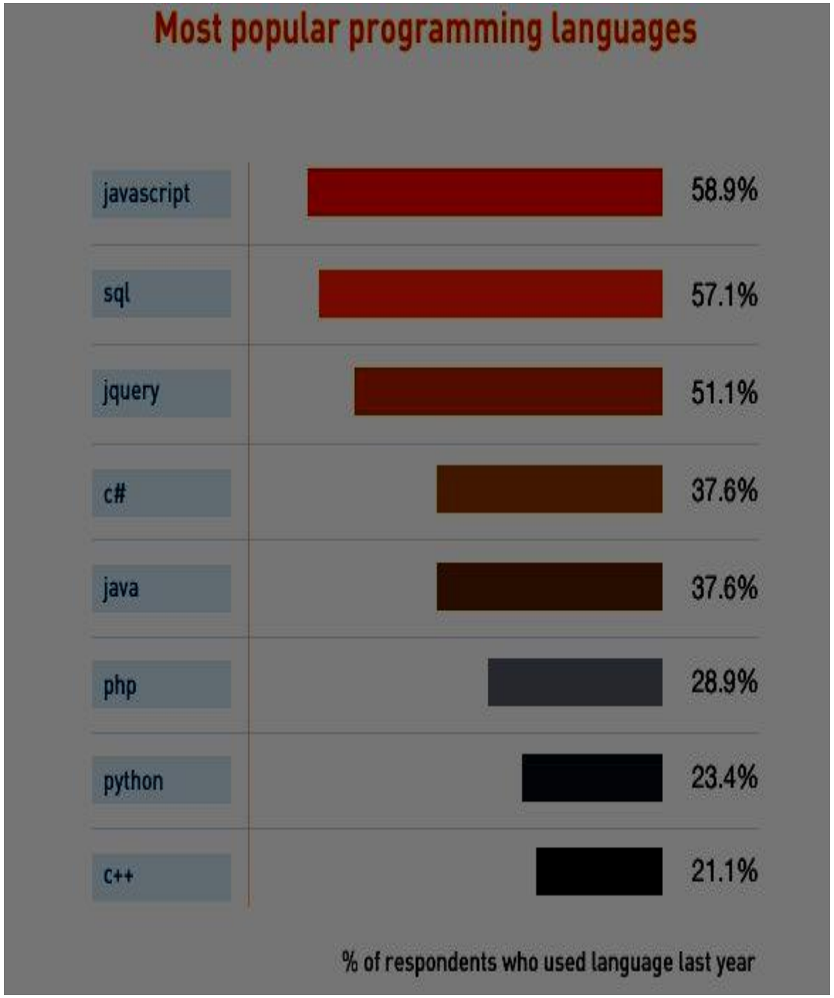


modulecounts.com

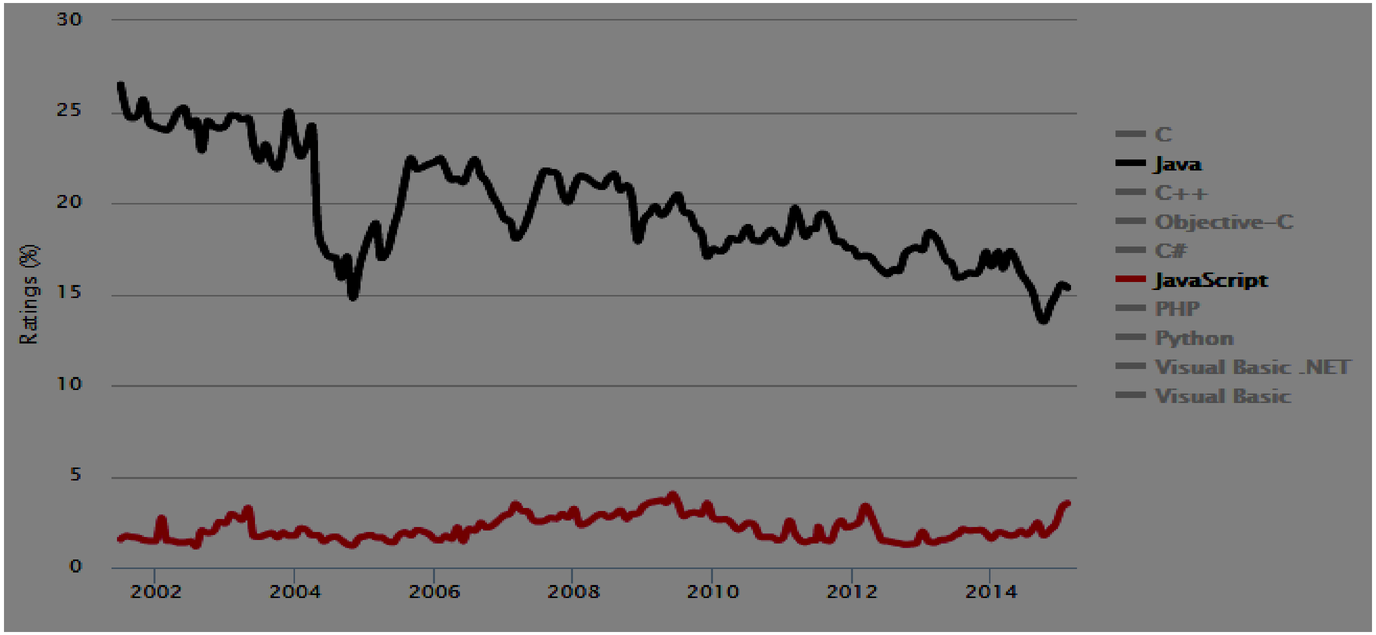


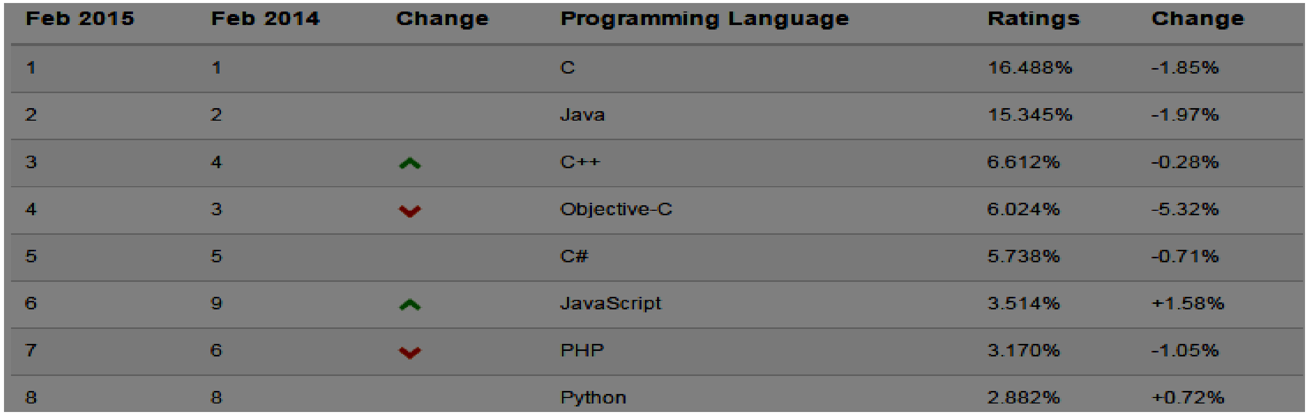
	Feb 12	Feb 13	Feb 14	Feb 15	Feb 16	Feb 17	Feb 18	Avg Growth
npm (node.js)	125055	125282	125466	125646	125887	126165	126382	221/day
Maven Central (Java)	97449	97493	97586	97632	97658	97834	97959	85/day

StackOverflow User Survey



Tiobe Community Programming Index

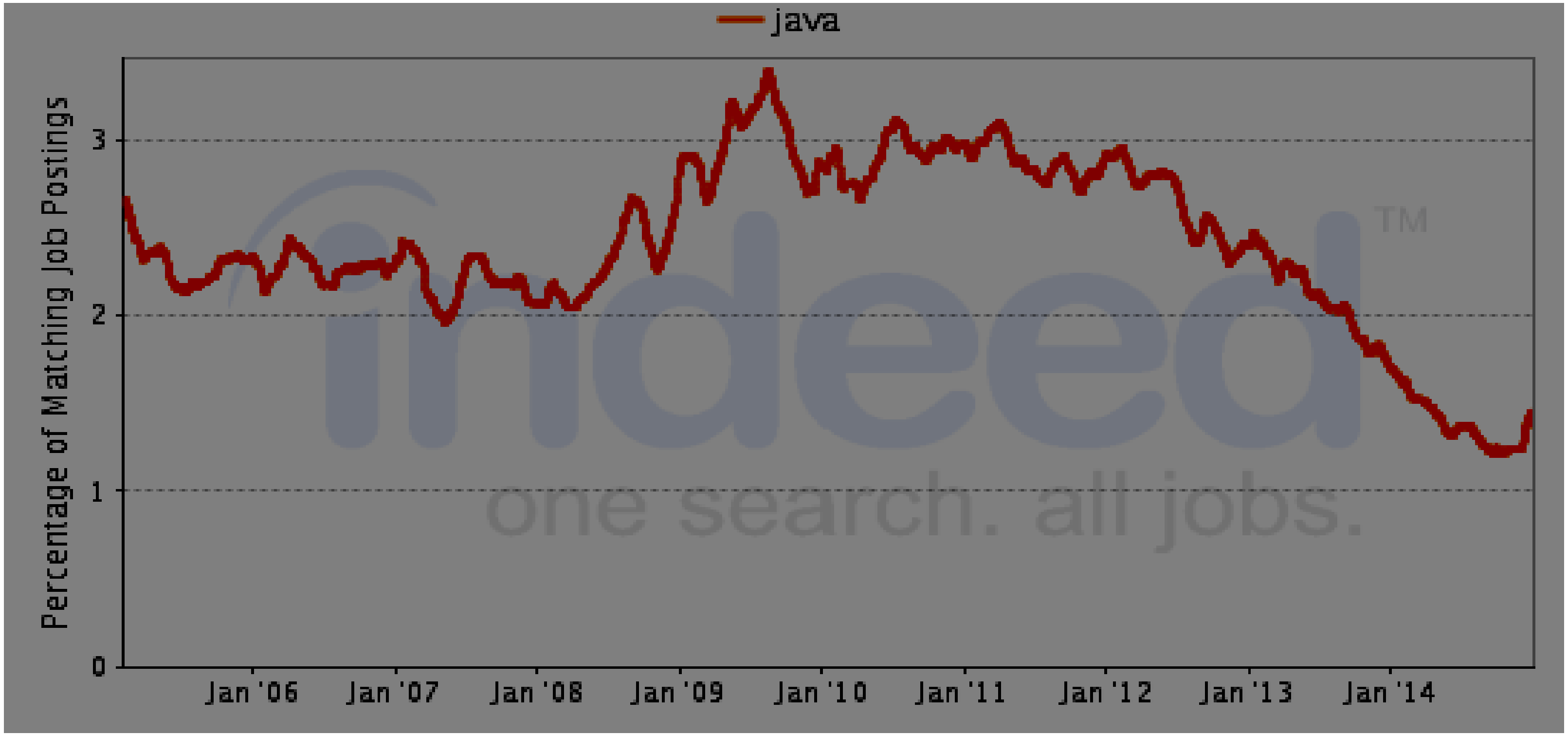




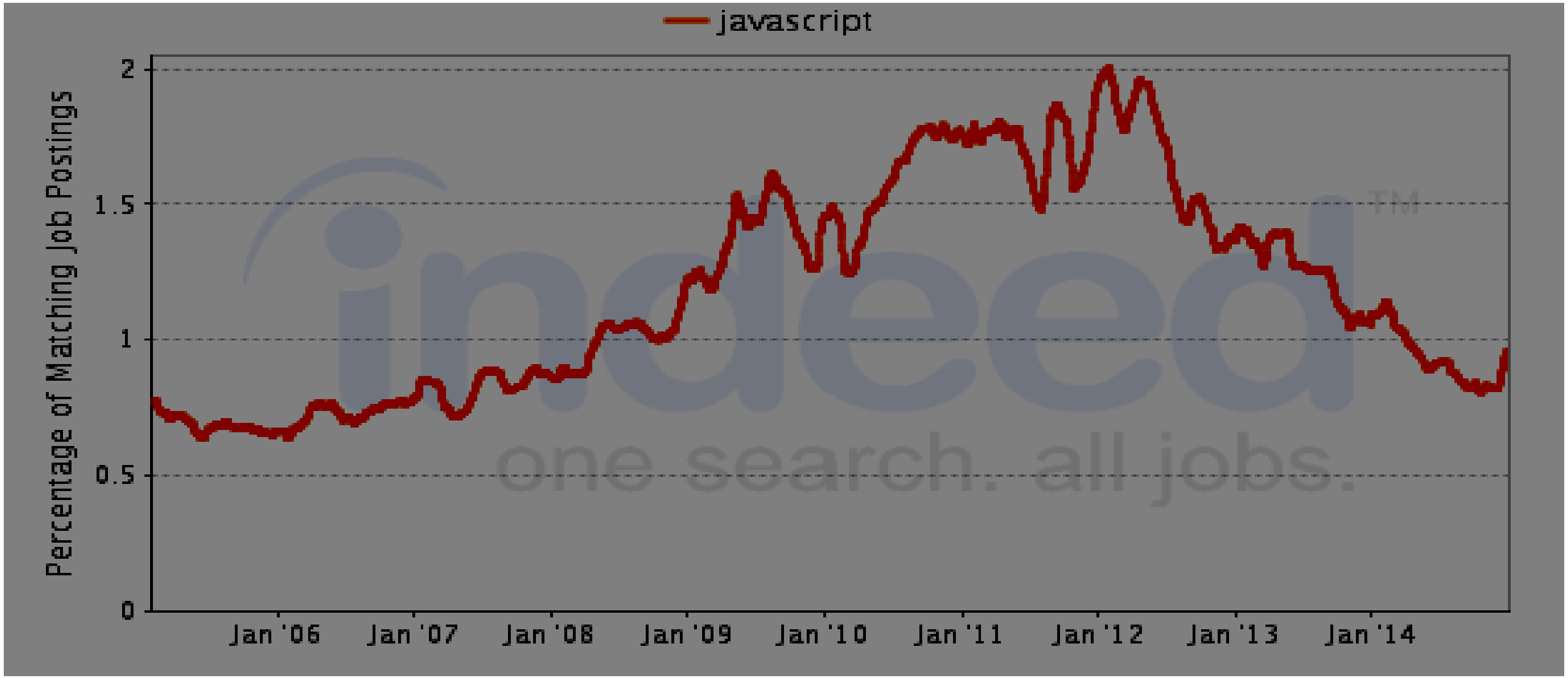
Feb 2015	Feb 2014	Change	Programming Language	Ratings	Change
1	1		C	16.488%	-1.85%
2	2		Java	15.345%	-1.97%
3	4	▲	C++	6.612%	-0.28%
4	3	▼	Objective-C	6.024%	-5.32%
5	5		C#	5.738%	-0.71%
6	9	▲	JavaScript	3.514%	+1.58%
7	6	▼	PHP	3.170%	-1.05%
8	8		Python	2.882%	+0.72%

Ratings based on the number of skilled engineers, courses and third party vendors.

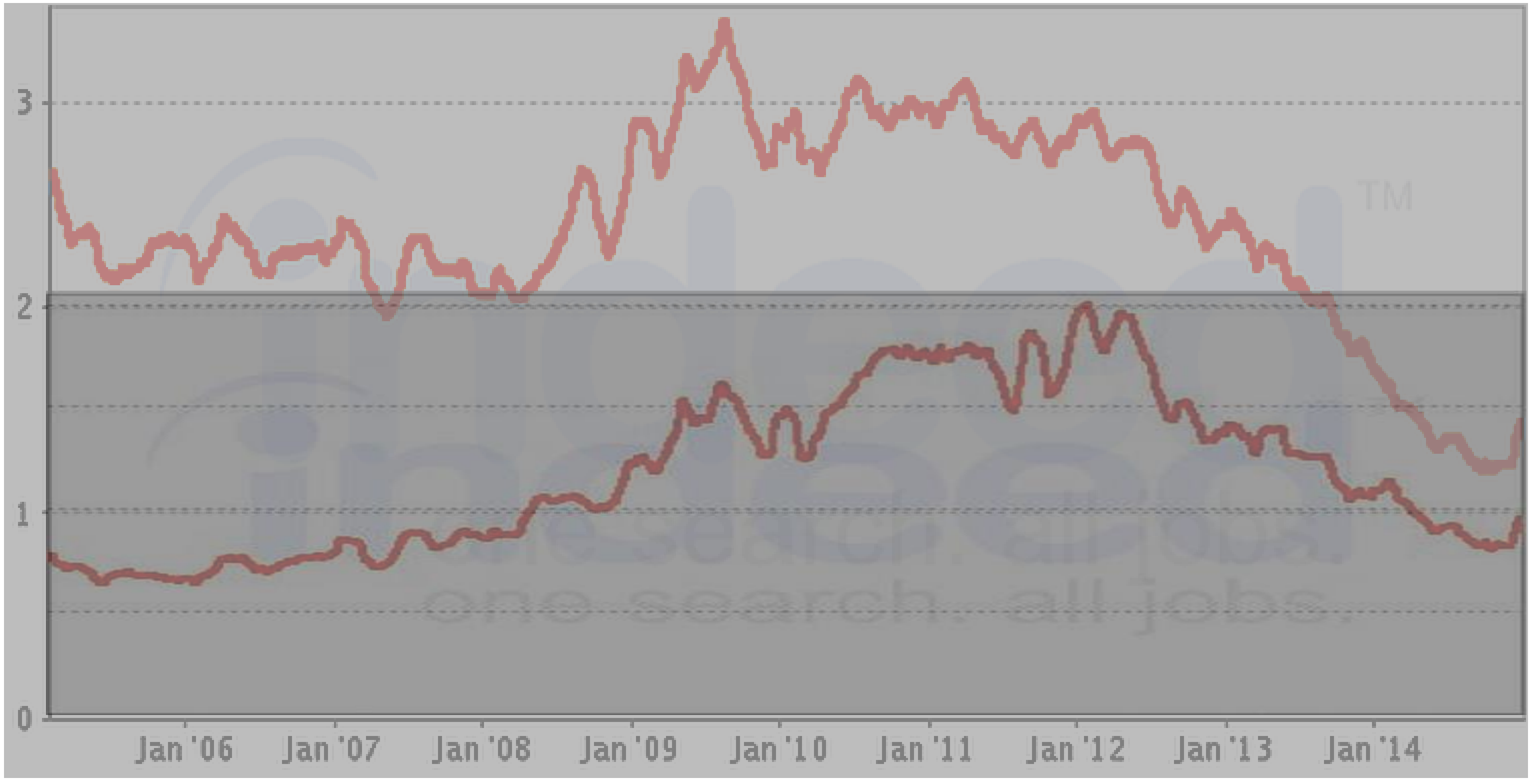
Indeed.com Job Trends: Java



Indeed.com Job Trends: JavaScript



Indeed.com Job Trends



Language Adoption

- JavaScript has a large developer base

-#1 on GitHub with 45% more active repositories than Java
-#1 on modulecounts.com with 29% more NPM modules than Maven
-#1 used language by StackOverflow survey responders
-#6 language on the Tiobe index

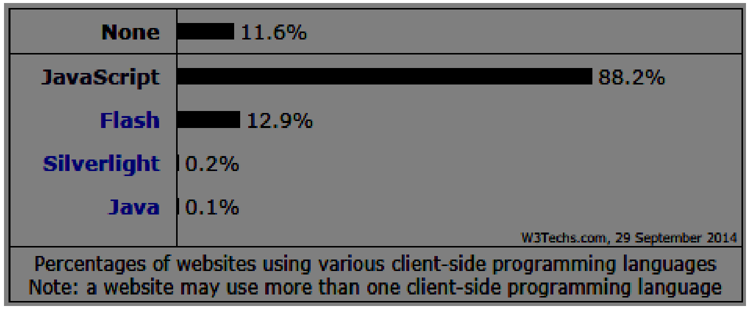
- Java remains hugely relevant, particularly on the server

-#2 on GitHub with 52% more active repositories than the next language
-#3 on modulecounts with 73.8% more modules than the next language
-#2 language on the Tiobe index
-#1 on indeed.com for developer jobs

Deployment Modes

Usage in the browser

- JavaScript is ubiquitous in the browser
  - Supported in every browser
  - Full integration with HTML and CSS



W3Techs.com, 29 September 2014

Percentages of websites using various client-side programming languages
Note: a website may use more than one client-side programming language

- JavaScript is not affected by negative publicity....

CERT | Software Engineering Institute | Carnegie Mellon University

Homeland Security
Sponsored by the DHS Office of Cybersecurity and Communications

***Unless** it is **absolutely necessary** to run Java **in web browsers, disable it** as described below, even after updating to 7u11. This will help mitigate other Java vulnerabilities that may be discovered in the future.*

US-CERT
UNITED STATES COMPUTER EMERGENCY READINESS TEAM

*This and previous Java vulnerabilities have been widely targeted by attackers, and new Java vulnerabilities are likely to be discovered. **To defend against** this and **future** Java **vulnerabilities**, **consider disabling Java in web browsers**…*

Usage on the server

- Java has a long history on the server
  - JPE launched in 1998

- Java has rich platform support:
  - Linux x86, Linux POWER, zLinux
  - Windows, Mac OS, Solaris, AIX, z/OS

- JavaScript is a nascent language on the server
  - Limited platform support – although its growing
  - No API support to interact with the OS
- Part of the browser security model
  - Frameworks like Node.js have changed that.

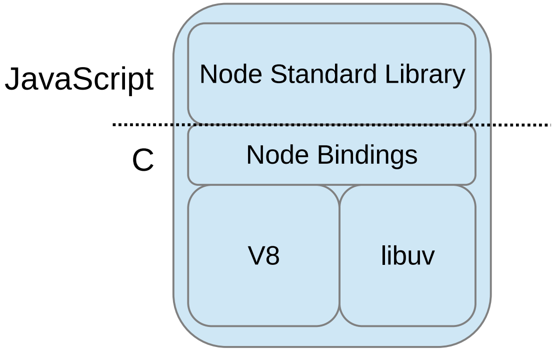
Server Side JavaScript: Node.js

- Single Threaded Event based JavaScript framework

–Uses non-blocking asynchronous I/O

- Wraps the Chrome V8 JavaScript engine with I/O interfaces

–Libuv provides interaction with OS/system



- Designed to build scalable network applications

–Suited for real time delivery of data to distributed client

- Available on a growing set of platforms

-Windows, Linux x86, Linux ARM, Mac OS, Solaris

-Linux POWER, zLinux, AIX

Async I/O Model

Typical approach to I/O

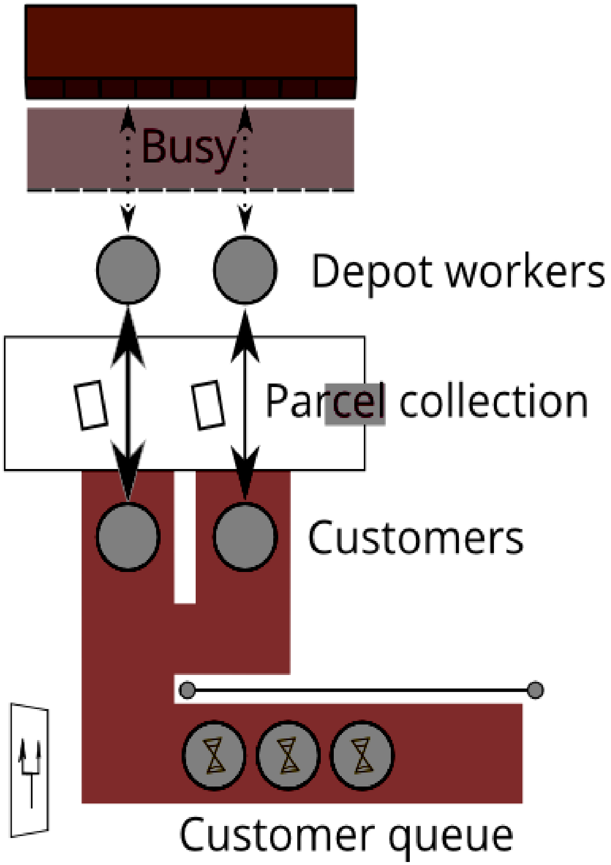
- One thread (or process) per connection
  - Each thread waits on a response
  - Scalability determined by the number of threads

- Each thread:
  - consumes memory
  - is relatively idle

- Number of concurrent customers determined by number of depot workers

- Additional customers wait in a queue with no response

Parcel collection depot



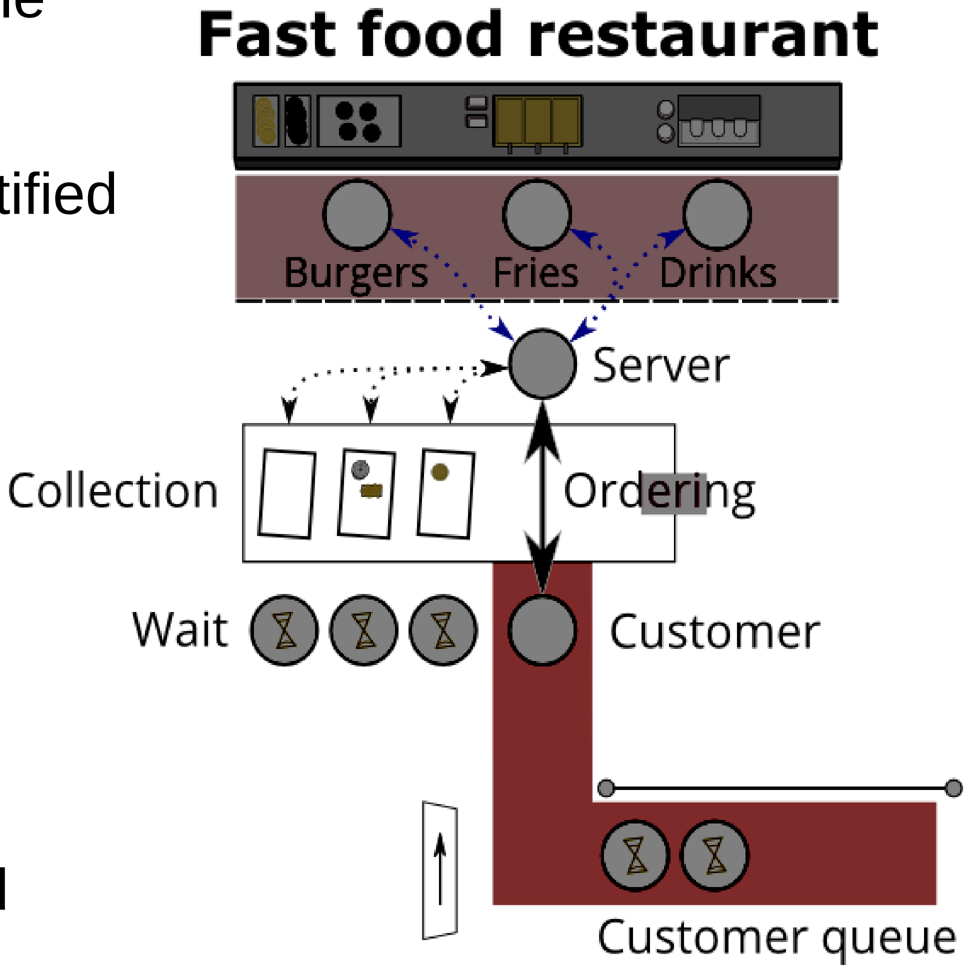
Asycnhronous Non-Blocking I/O

- One thread multiplexes for multiple requests
  - -No waiting for a response
  - -Handles return from I/O when notified

- Scalability determined by:
  - -CPU usage
  - -“Back end” responsiveness

- Number of concurrent customers determined by how fast the food Server can work

- Or until the kitchen gets slammed



Drawbacks of Asynchronous I/O

- Tasks must execute quickly to avoid blocking the event queue
-Analogous to work done under a lock
-Stick to the right jobs, eg, I/O
-Delegate CPU bound tasks to back end processes

- Easy to run out of memory
-No direct bound on amount of parallel work
-Holding state for each piece or work means unbounded memory usage

JavaScript and Asynchronous I/O

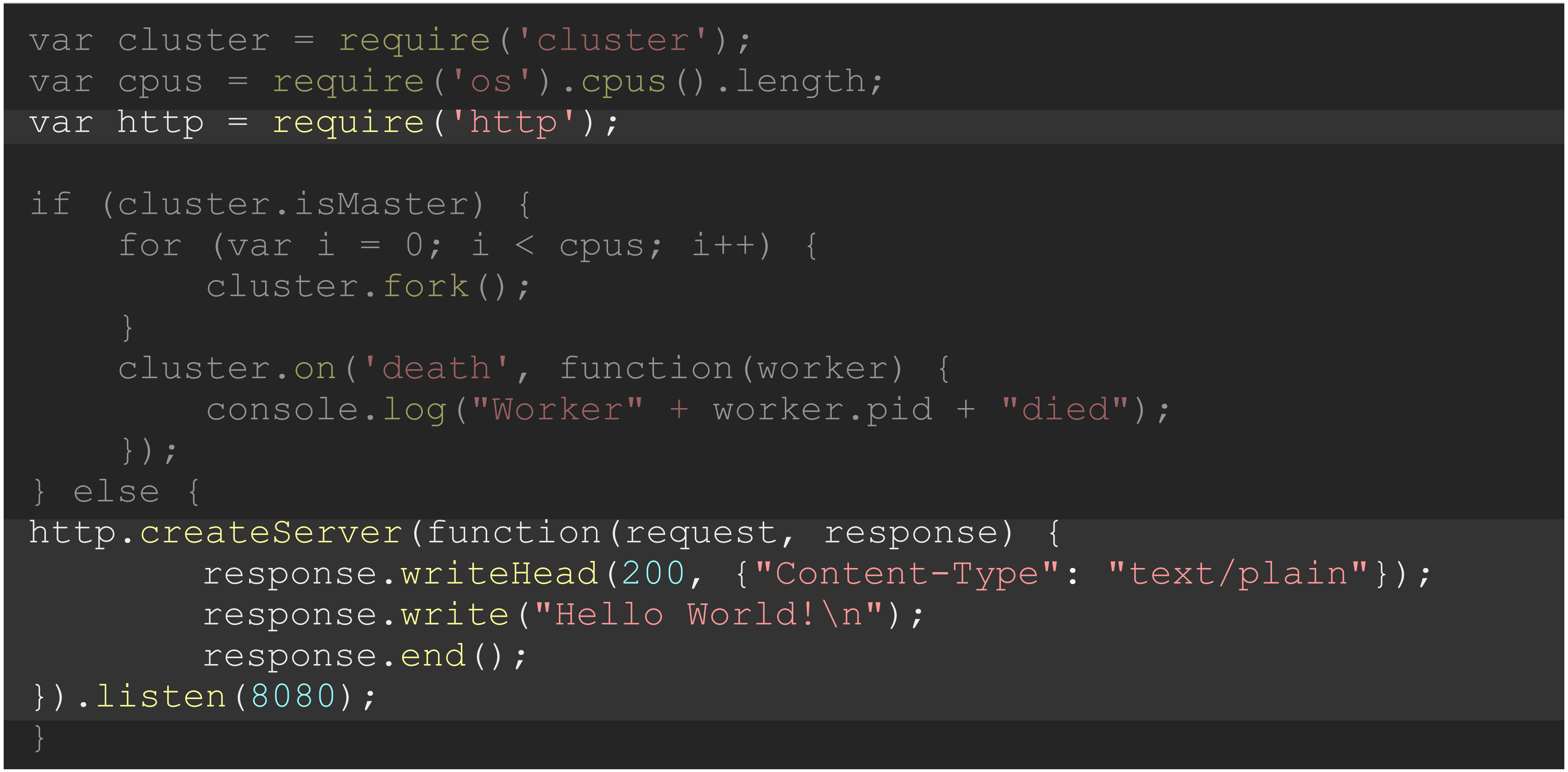
- JavaScript is already event based in the browser
-eg. onClick and onMouseOver events

- First class functions and closures fit well with events
-Easy to create and pass function callbacks
-Easy to execute callbacks in the context of the event

- Node.js execution is based on an event loop
-Asynchronous I/O built in from the ground up

- Node.js execution uses a single thread
-No need to worry about locking or shared data
-Most machines are now multi-CPU, so cluster capabilities are provided

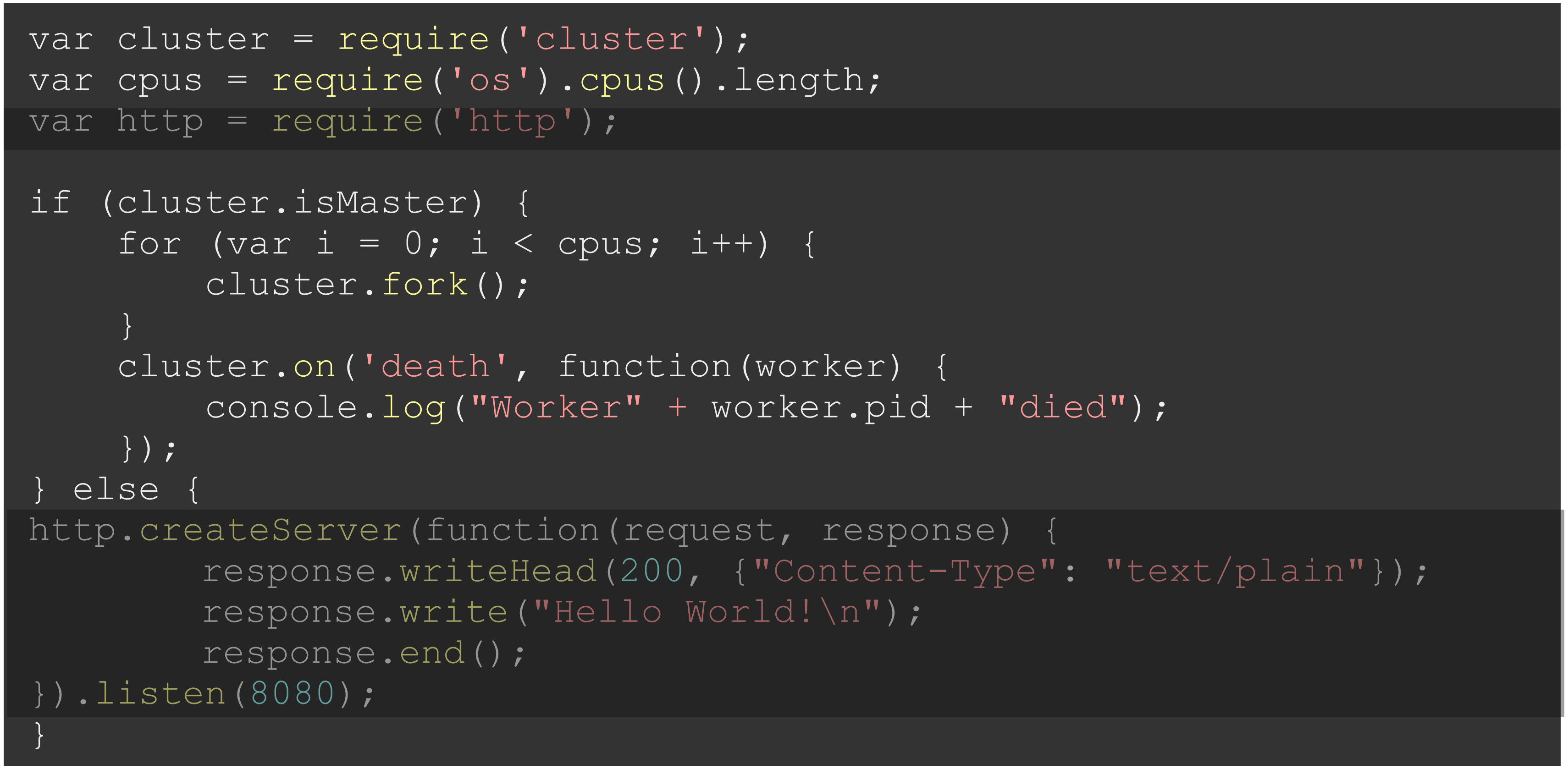
HTTP Server Example



```
var cluster = require('cluster');
var cpus = require('os').cpus().length;
var http = require('http');

if (cluster.isMaster) {
    for (var i = 0; i < cpus; i++) {
        cluster.fork();
    }
    cluster.on('death', function(worker) {
        console.log("Worker" + worker.pid + "died");
    });
} else {
http.createServer(function(request, response) {
        response.writeHead(200, {"Content-Type": "text/plain"});
        response.write("Hello World!\n");
        response.end();
}).listen(8080);
}
```

HTTP Server Example with Clustering



```
var cluster = require('cluster');
var cpus = require('os').cpus().length;
var http = require('http');

if (cluster.isMaster) {
    for (var i = 0; i < cpus; i++) {
        cluster.fork();
    }
    cluster.on('death', function(worker) {
        console.log("Worker" + worker.pid + "died");
    });
} else {
http.createServer(function(request, response) {
        response.writeHead(200, {"Content-Type": "text/plain"});
        response.write("Hello World!\n");
        response.end();
}).listen(8080);
}
```

JavaScript and Asynchronous I/O

- Very little time spent with events on the Event Loop

- Provides good scalability, so should provide great performance for IO bound apps

- Like WebApplications...

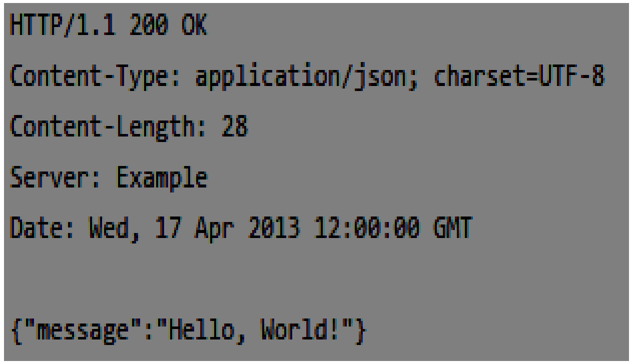
WebApp Performance

JSON Serialization

- JSON serialization of a newly instantiated object

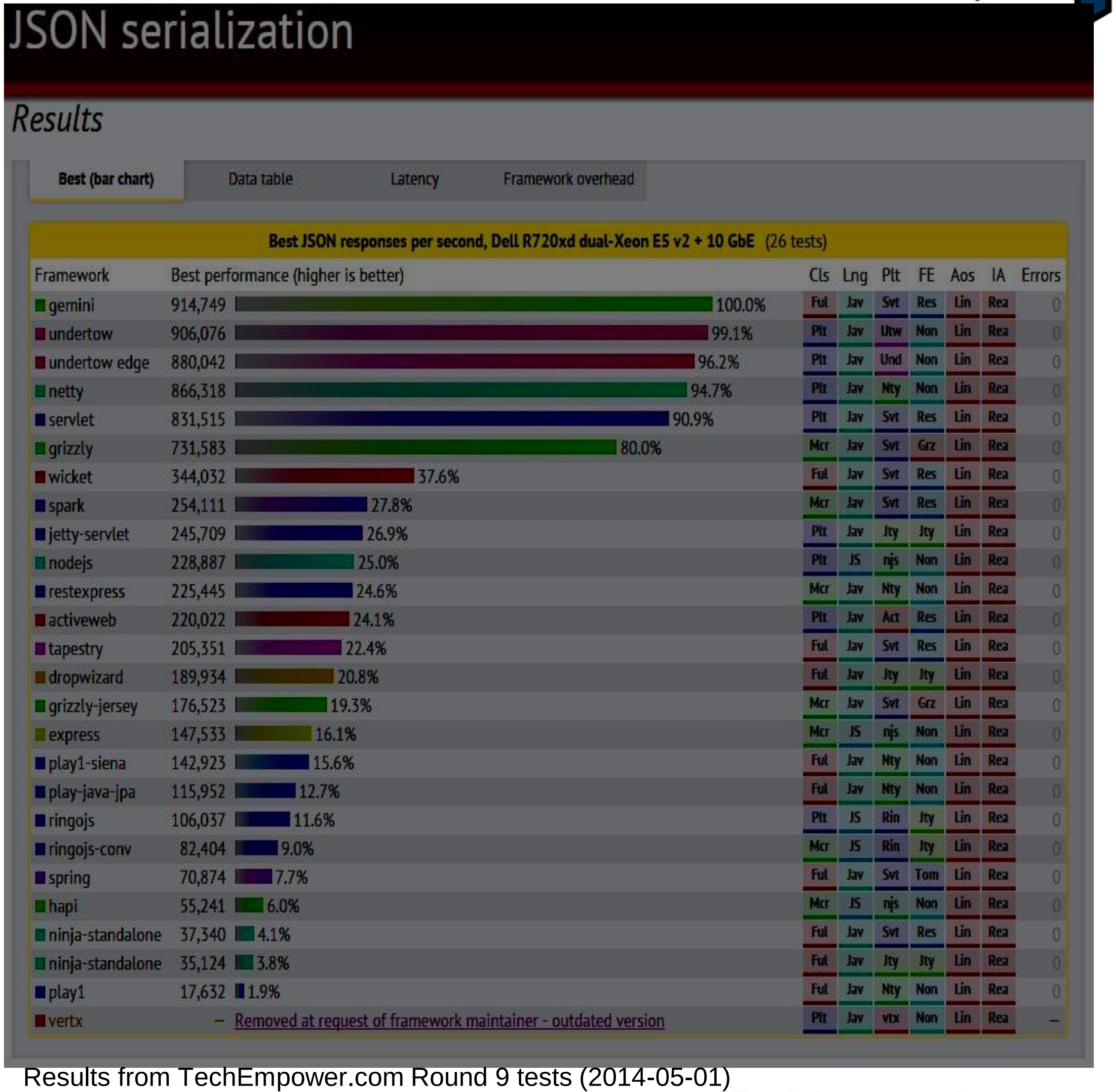
- Maps
-Key of *message*
-Value of *Hello, World!*

- Example response:



```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 28
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT

{"message":"Hello, World!"}
```



JSON serialization

Results

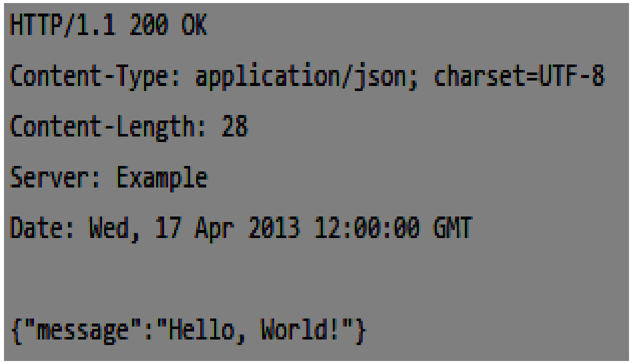
Best JSON responses per second, Dell R720xd dual-Xeon E5 v2 + 10 GbE (26 tests)

Framework	Best performance (higher is better)		Cls	Lng	Plt	FE	Aos	IA	Errors
gemini	914,749	100.0%	Ful	Jav	Svt	Res	Lin	Rea	0
undertow	906,076	99.1%	Plt	Jav	Utw	Non	Lin	Rea	0
undertow edge	880,042	96.2%	Plt	Jav	Und	Non	Lin	Rea	0
netty	866,318	94.7%	Plt	Jav	Nty	Non	Lin	Rea	0
servlet	831,515	90.9%	Plt	Jav	Svt	Res	Lin	Rea	0
grizzly	731,583	80.0%	Mcr	Jav	Svt	Grz	Lin	Rea	0
wicket	344,032	37.6%	Ful	Jav	Svt	Res	Lin	Rea	0
spark	254,111	27.8%	Mcr	Jav	Svt	Res	Lin	Rea	0
jetty-servlet	245,709	26.9%	Plt	Jav	Jty	Jty	Lin	Rea	0
nodejs	228,887	25.0%	Plt	JS	njs	Non	Lin	Rea	0
restexpress	225,445	24.6%	Mcr	Jav	Nty	Non	Lin	Rea	0
activeweb	220,022	24.1%	Plt	Jav	Act	Res	Lin	Rea	0
tapestry	205,351	22.4%	Ful	Jav	Svt	Res	Lin	Rea	0
dropwizard	189,934	20.8%	Ful	Jav	Jty	Jty	Lin	Rea	0
grizzly-jersey	176,523	19.3%	Mcr	Jav	Svt	Grz	Lin	Rea	0
express	147,533	16.1%	Mcr	JS	njs	Non	Lin	Rea	0
play1-siena	142,923	15.6%	Ful	Jav	Nty	Non	Lin	Rea	0
play-java-jpa	115,952	12.7%	Ful	Jav	Nty	Non	Lin	Rea	0
ringojs	106,037	11.6%	Plt	JS	Rin	Jty	Lin	Rea	0
ringojs-conv	82,404	9.0%	Mcr	JS	Rin	Jty	Lin	Rea	0
spring	70,874	7.7%	Ful	Jav	Svt	Tom	Lin	Rea	0
hapi	55,241	6.0%	Mcr	JS	njs	Non	Lin	Rea	0
ninja-standalone	37,340	4.1%	Ful	Jav	Svt	Res	Lin	Rea	0
ninja-standalone	35,124	3.8%	Ful	Jav	Jty	Jty	Lin	Rea	0
play1	17,632	1.9%	Ful	Jav	Nty	Non	Lin	Rea	0
vertx	–	Removed at request of framework maintainer - outdated version	Plt	Jav	vtx	Non	Lin	Rea	–

Results from TechEmpower.com Round 9 tests (2014-05-01)

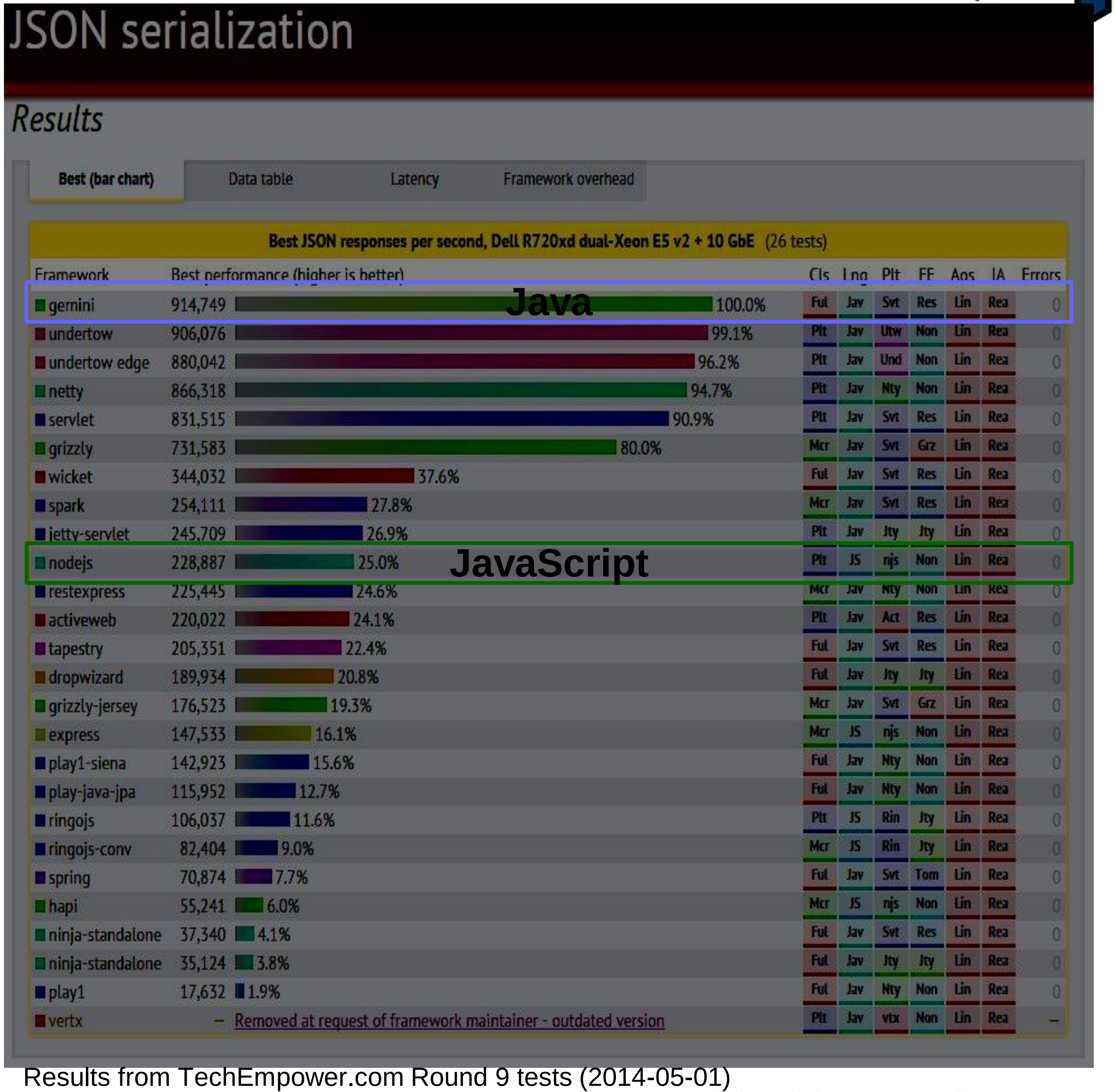
JSON Serialization

- JSON serialization of a newly instantiated object
- Maps
  - Key of *message*
  - Value of *Hello, World!*
- Example response:



```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 28
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT

{"message":"Hello, World!"}
```



JSON serialization

Results

Best JSON responses per second, Dell R720xd dual-Xeon E5 v2 + 10 GbE (26 tests)

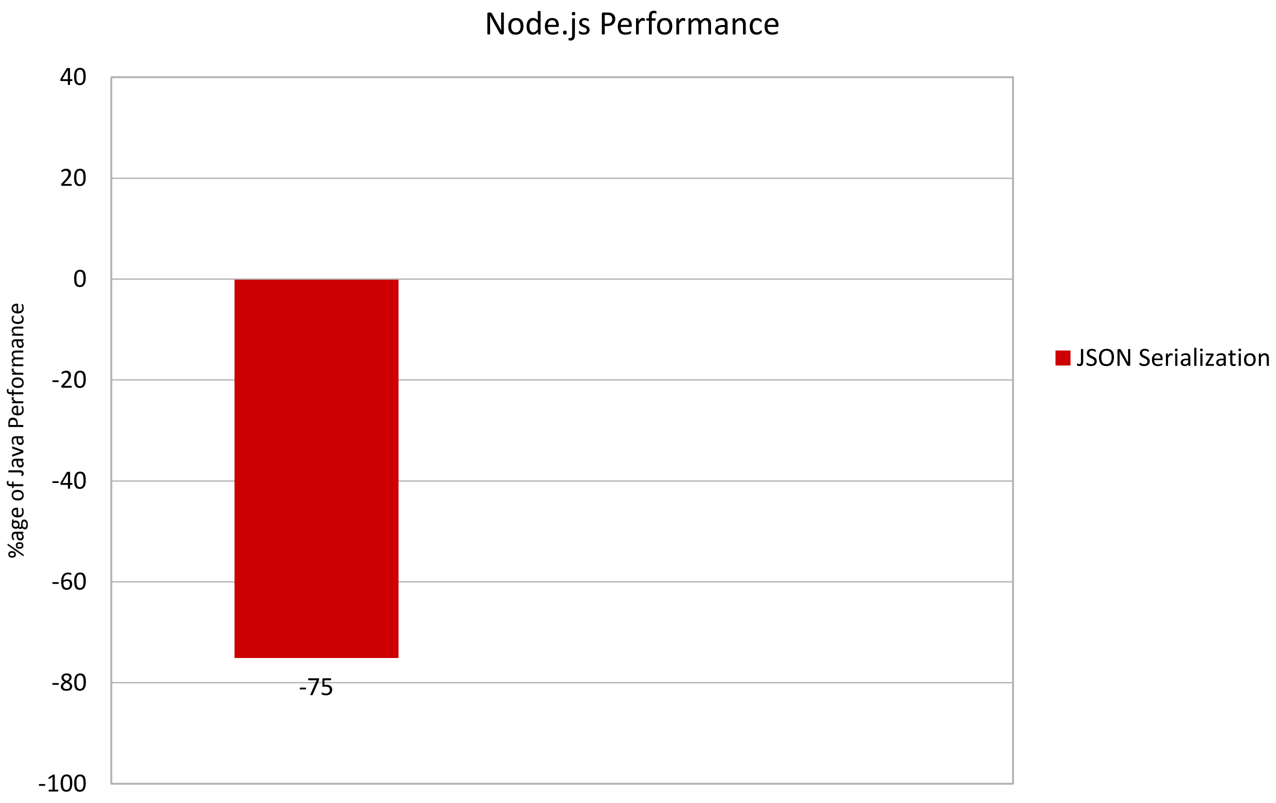
Framework	Best performance (higher is better)		Cls	Lng	Plt	FE	Aos	IA	Errors
gemini	914,749	100.0%	Ful	Jav	Svt	Res	Lin	Rea	0
undertow	906,076	99.1%	Plt	Jav	Utw	Non	Lin	Rea	0
undertow edge	880,042	96.2%	Plt	Jav	Und	Non	Lin	Rea	0
netty	866,318	94.7%	Plt	Jav	Nty	Non	Lin	Rea	0
servlet	831,515	90.9%	Plt	Jav	Svt	Res	Lin	Rea	0
grizzly	731,583	80.0%	Mcr	Jav	Svt	Grz	Lin	Rea	0
wicket	344,032	37.6%	Ful	Jav	Svt	Res	Lin	Rea	0
spark	254,111	27.8%	Mcr	Jav	Svt	Res	Lin	Rea	0
jetty-servlet	245,709	26.9%	Plt	Jav	Jty	Jty	Lin	Rea	0
nodejs	228,887	25.0%	Plt	JS	njs	Non	Lin	Rea	0
restexpress	225,445	24.6%	Mcr	Jav	Nty	Non	Lin	Rea	0
activeweb	220,022	24.1%	Plt	Jav	Act	Res	Lin	Rea	0
tapestry	205,351	22.4%	Ful	Jav	Svt	Res	Lin	Rea	0
dropwizard	189,934	20.8%	Ful	Jav	Jty	Jty	Lin	Rea	0
grizzly-jersey	176,523	19.3%	Mcr	Jav	Svt	Grz	Lin	Rea	0
express	147,533	16.1%	Mcr	JS	njs	Non	Lin	Rea	0
play1-siena	142,923	15.6%	Ful	Jav	Nty	Non	Lin	Rea	0
play-java-jpa	115,952	12.7%	Ful	Jav	Nty	Non	Lin	Rea	0
ringojs	106,037	11.6%	Plt	JS	Rin	Jty	Lin	Rea	0
ringojs-conv	82,404	9.0%	Mcr	JS	Rin	Jty	Lin	Rea	0
spring	70,874	7.7%	Ful	Jav	Svt	Tom	Lin	Rea	0
hapi	55,241	6.0%	Mcr	JS	njs	Non	Lin	Rea	0
ninja-standalone	37,340	4.1%	Ful	Jav	Svt	Res	Lin	Rea	0
ninja-standalone	35,124	3.8%	Ful	Jav	Jty	Jty	Lin	Rea	0
play1	17,632	1.9%	Ful	Jav	Nty	Non	Lin	Rea	0
vertx	–	Removed at request of framework maintainer - outdated version	Plt	Jav	vtx	Non	Lin	Rea	–

Java

JavaScript

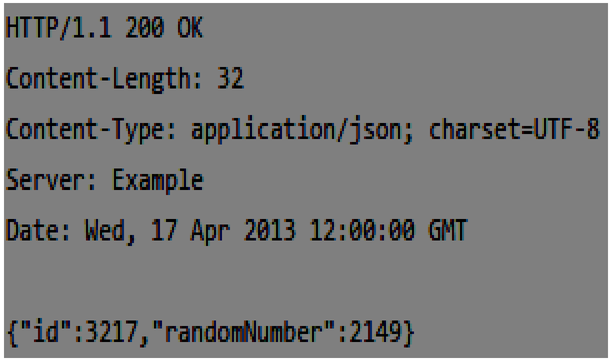
Results from TechEmpower.com Round 9 tests (2014-05-01)

JavaScript WebApp Performance



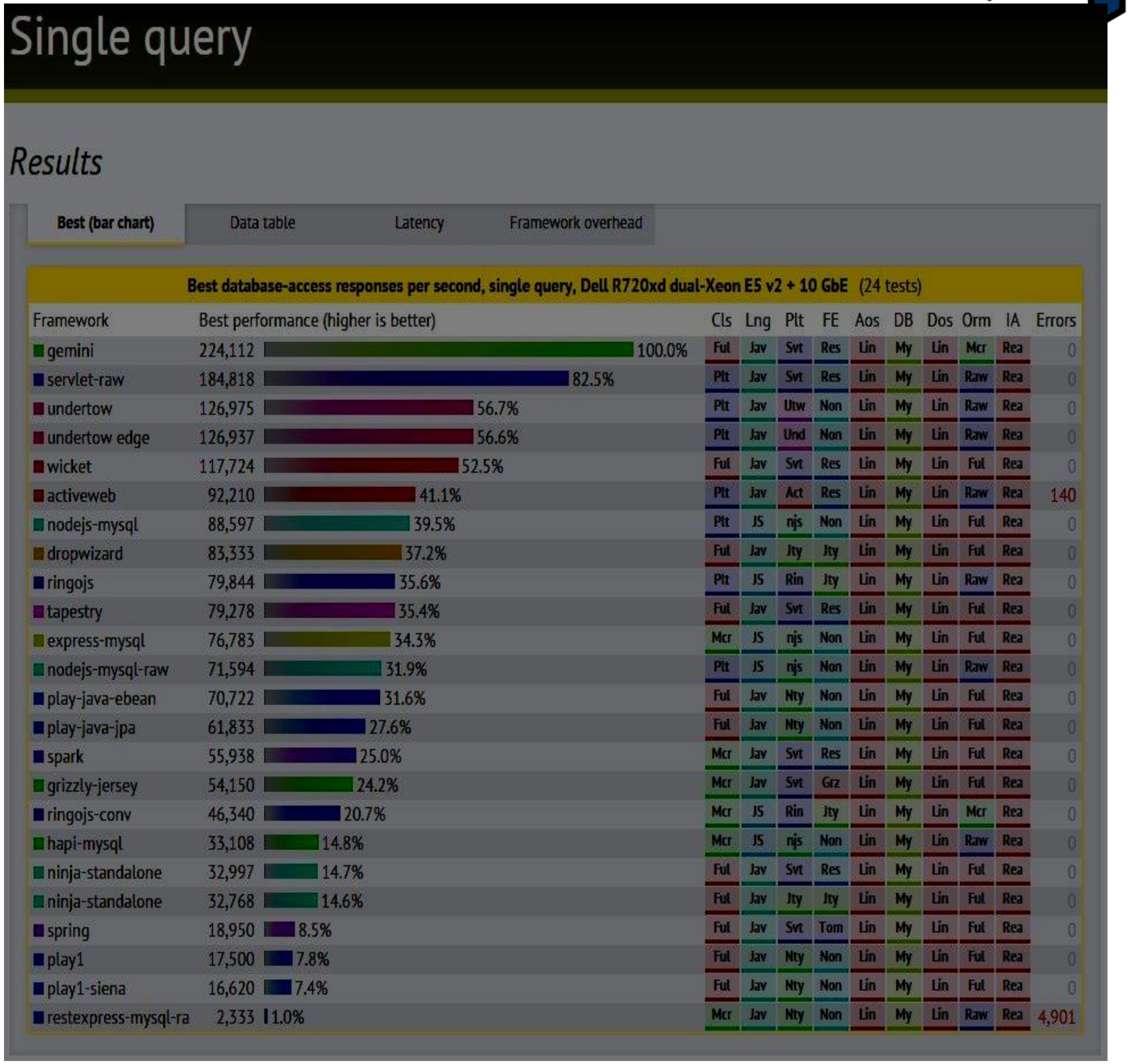
Single Query

- Fetches single row from simple database table
- Row serialized as JSON
- Example response:



```
HTTP/1.1 200 OK
Content-Length: 32
Content-Type: application/json; charset=UTF-8
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT

{"id":3217,"randomNumber":2149}
```



Single query

Results

Best database-access responses per second, single query, Dell R720xd dual-Xeon E5 v2 + 10 GbE (24 tests)

Framework	Best performance (higher is better)		Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA	Errors
gemini	224,112	100.0%	Ful	Jav	Svt	Res	Lin	My	Lin	Mcr	Rea	0
servlet-raw	184,818	82.5%	Plt	Jav	Svt	Res	Lin	My	Lin	Raw	Rea	0
undertow	126,975	56.7%	Plt	Jav	Utw	Non	Lin	My	Lin	Raw	Rea	0
undertow edge	126,937	56.6%	Plt	Jav	Und	Non	Lin	My	Lin	Raw	Rea	0
wicket	117,724	52.5%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
activeweb	92,210	41.1%	Plt	Jav	Act	Res	Lin	My	Lin	Raw	Rea	140
nodejs-mysql	88,597	39.5%	Plt	JS	njs	Non	Lin	My	Lin	Ful	Rea	0
dropwizard	83,333	37.2%	Ful	Jav	Jty	Jty	Lin	My	Lin	Ful	Rea	0
ringojs	79,844	35.6%	Plt	JS	Rin	Jty	Lin	My	Lin	Raw	Rea	0
tapestry	79,278	35.4%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
express-mysql	76,783	34.3%	Mcr	JS	njs	Non	Lin	My	Lin	Ful	Rea	0
nodejs-mysql-raw	71,594	31.9%	Plt	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
play-java-ebean	70,722	31.6%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
play-java-jpa	61,833	27.6%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
spark	55,938	25.0%	Mcr	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
grizzly-jersey	54,150	24.2%	Mcr	Jav	Svt	Grz	Lin	My	Lin	Ful	Rea	0
ringojs-conv	46,340	20.7%	Mcr	JS	Rin	Jty	Lin	My	Lin	Mcr	Rea	0
hapi-mysql	33,108	14.8%	Mcr	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
ninja-standalone	32,997	14.7%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
ninja-standalone	32,768	14.6%	Ful	Jav	Jty	Jty	Lin	My	Lin	Ful	Rea	0
spring	18,950	8.5%	Ful	Jav	Svt	Tom	Lin	My	Lin	Ful	Rea	0
play1	17,500	7.8%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
play1-siena	16,620	7.4%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
restexpress-mysql-ra	2,333	1.0%	Mcr	Jav	Nty	Non	Lin	My	Lin	Raw	Rea	4,901

Results from TechEmpower.com Round 9 tests (2014-05-01)

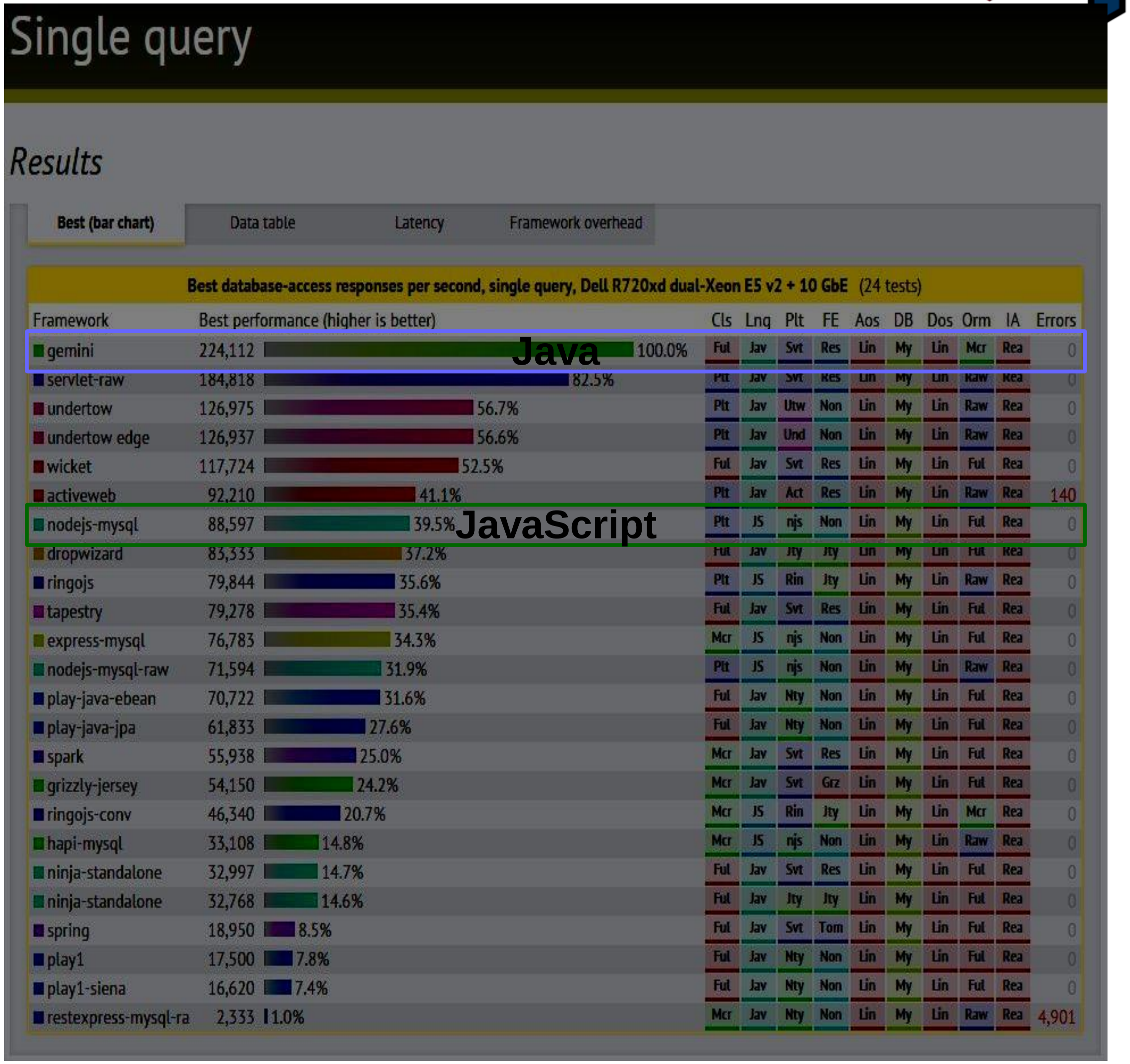
Single Query

- Fetches single row from simple database table
- Row serialized as JSON

- Example response:

```
HTTP/1.1 200 OK
Content-Length: 32
Content-Type: application/json; charset=UTF-8
Server: Example
Date: Wed, 17 Apr 2013 12:00:00 GMT

{"id":3217,"randomNumber":2149}
```



Single query

Results

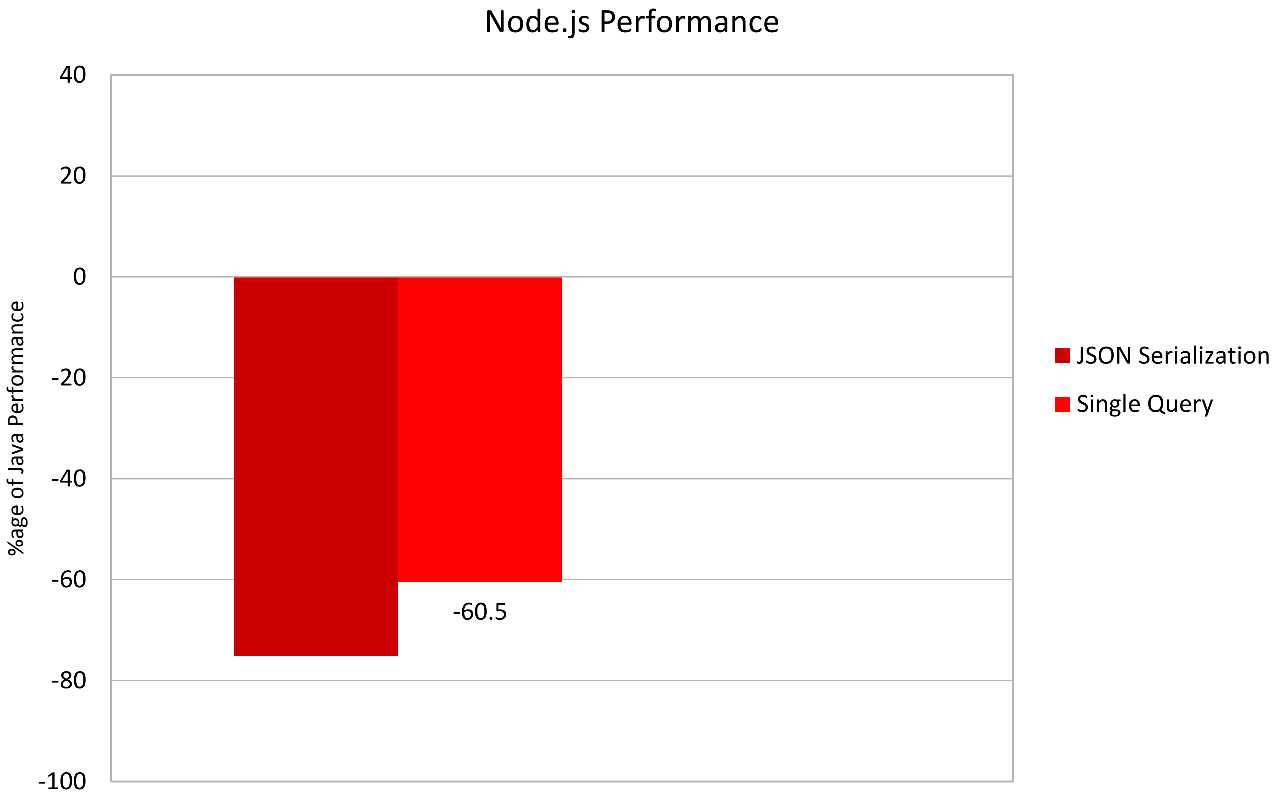
Best (bar chart) | Data table | Latency | Framework overhead

Best database-access responses per second, single query, Dell R720xd dual-Xeon E5 v2 + 10 GbE (24 tests)

Framework	Best performance (higher is better)	Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA	Errors
gemini	224,112 100.0%	Ful	Jav	Svt	Res	Lin	My	Lin	Mcr	Rea	0
servlet-raw	184,818 82.5%	Plt	Jav	Svt	Res	Lin	My	Lin	Raw	Rea	0
undertow	126,975 56.7%	Plt	Jav	Utw	Non	Lin	My	Lin	Raw	Rea	0
undertow edge	126,937 56.6%	Plt	Jav	Und	Non	Lin	My	Lin	Raw	Rea	0
wicket	117,724 52.5%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
activeweb	92,210 41.1%	Plt	Jav	Act	Res	Lin	My	Lin	Raw	Rea	140
nodejs-mysql	88,597 39.5%	Plt	JS	njs	Non	Lin	My	Lin	Ful	Rea	0
dropwizard	83,333 37.2%	Ful	Jav	Jty	Jty	Lin	My	Lin	Ful	Rea	0
ringojs	79,844 35.6%	Plt	JS	Rin	Jty	Lin	My	Lin	Raw	Rea	0
tapestry	79,278 35.4%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
express-mysql	76,783 34.3%	Mcr	JS	njs	Non	Lin	My	Lin	Ful	Rea	0
nodejs-mysql-raw	71,594 31.9%	Plt	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
play-java-ebean	70,722 31.6%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
play-java-jpa	61,833 27.6%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
spark	55,938 25.0%	Mcr	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
grizzly-jersey	54,150 24.2%	Mcr	Jav	Svt	Grz	Lin	My	Lin	Ful	Rea	0
ringojs-conv	46,340 20.7%	Mcr	JS	Rin	Jty	Lin	My	Lin	Mcr	Rea	0
hapi-mysql	33,108 14.8%	Mcr	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
ninja-standalone	32,997 14.7%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
ninja-standalone	32,768 14.6%	Ful	Jav	Jty	Jty	Lin	My	Lin	Ful	Rea	0
spring	18,950 8.5%	Ful	Jav	Svt	Tom	Lin	My	Lin	Ful	Rea	0
play1	17,500 7.8%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
play1-siena	16,620 7.4%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
restexpress-mysql-ra	2,333 1.0%	Mcr	Jav	Nty	Non	Lin	My	Lin	Raw	Rea	4,901

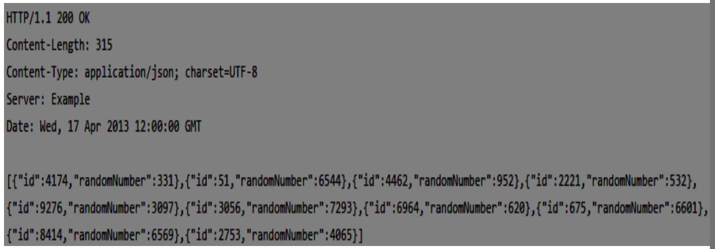
Results from TechEmpower.com Round 9 tests (2014-05-01)

JavaScript WebApp Performance



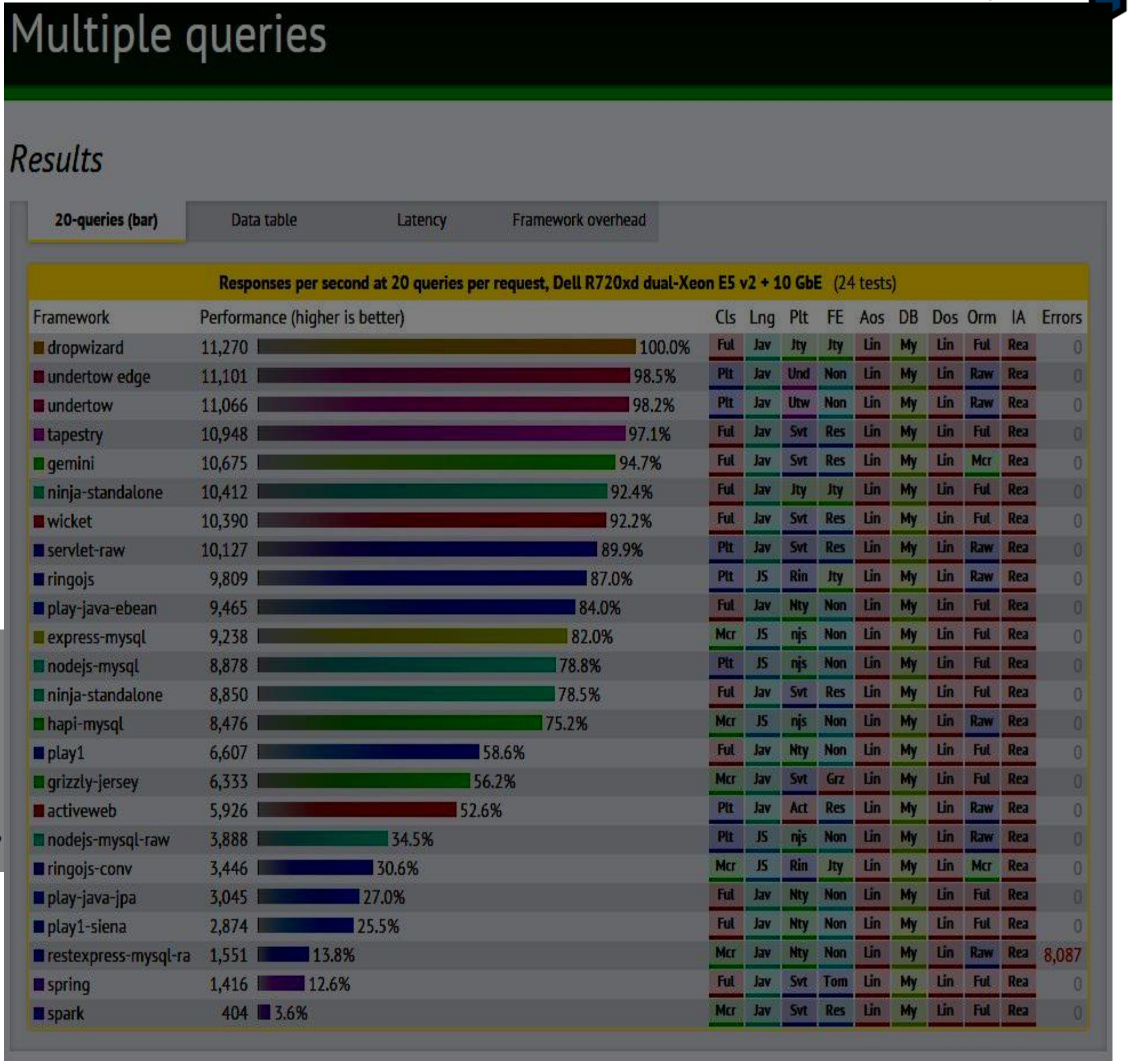
Multiple Queries

- Fetches multiple rows from a simple database table
- Rows serialized as JSON
- Example response:



```
HTTP/1.1 200 OK
Content-Length: 315
Content-Type: application/json; charset=UTF-8
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}]
```



Multiple queries

Results

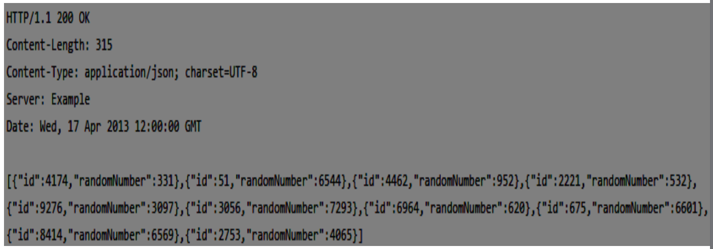
Responses per second at 20 queries per request, Dell R720xd dual-Xeon E5 v2 + 10 GbE (24 tests)

Framework	Performance (higher is better)		Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA	Errors
dropwizard	11,270	100.0%	Ful	Jav	Jty	Jty	Lin	My	Lin	Ful	Rea	0
undertow edge	11,101	98.5%	Plt	Jav	Und	Non	Lin	My	Lin	Raw	Rea	0
undertow	11,066	98.2%	Plt	Jav	Utw	Non	Lin	My	Lin	Raw	Rea	0
tapestry	10,948	97.1%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
gemini	10,675	94.7%	Ful	Jav	Svt	Res	Lin	My	Lin	Mcr	Rea	0
ninja-standalone	10,412	92.4%	Ful	Jav	Jty	Jty	Lin	My	Lin	Ful	Rea	0
wicket	10,390	92.2%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
servlet-raw	10,127	89.9%	Plt	Jav	Svt	Res	Lin	My	Lin	Raw	Rea	0
ringojs	9,809	87.0%	Plt	JS	Rin	Jty	Lin	My	Lin	Raw	Rea	0
play-java-ebean	9,465	84.0%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
express-mysql	9,238	82.0%	Mcr	JS	njs	Non	Lin	My	Lin	Ful	Rea	0
nodejs-mysql	8,878	78.8%	Plt	JS	njs	Non	Lin	My	Lin	Ful	Rea	0
ninja-standalone	8,850	78.5%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
hapi-mysql	8,476	75.2%	Mcr	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
play1	6,607	58.6%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
grizzly-jersey	6,333	56.2%	Mcr	Jav	Svt	Grz	Lin	My	Lin	Ful	Rea	0
activeweb	5,926	52.6%	Plt	Jav	Act	Res	Lin	My	Lin	Raw	Rea	0
nodejs-mysql-raw	3,888	34.5%	Plt	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
ringojs-conv	3,446	30.6%	Mcr	JS	Rin	Jty	Lin	My	Lin	Mcr	Rea	0
play-java-jpa	3,045	27.0%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
play1-siena	2,874	25.5%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
restexpress-mysql-ra	1,551	13.8%	Mcr	Jav	Nty	Non	Lin	My	Lin	Raw	Rea	8,087
spring	1,416	12.6%	Ful	Jav	Svt	Tom	Lin	My	Lin	Ful	Rea	0
spark	404	3.6%	Mcr	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0

Results from TechEmpower.com Round 9 tests (2014-05-01)

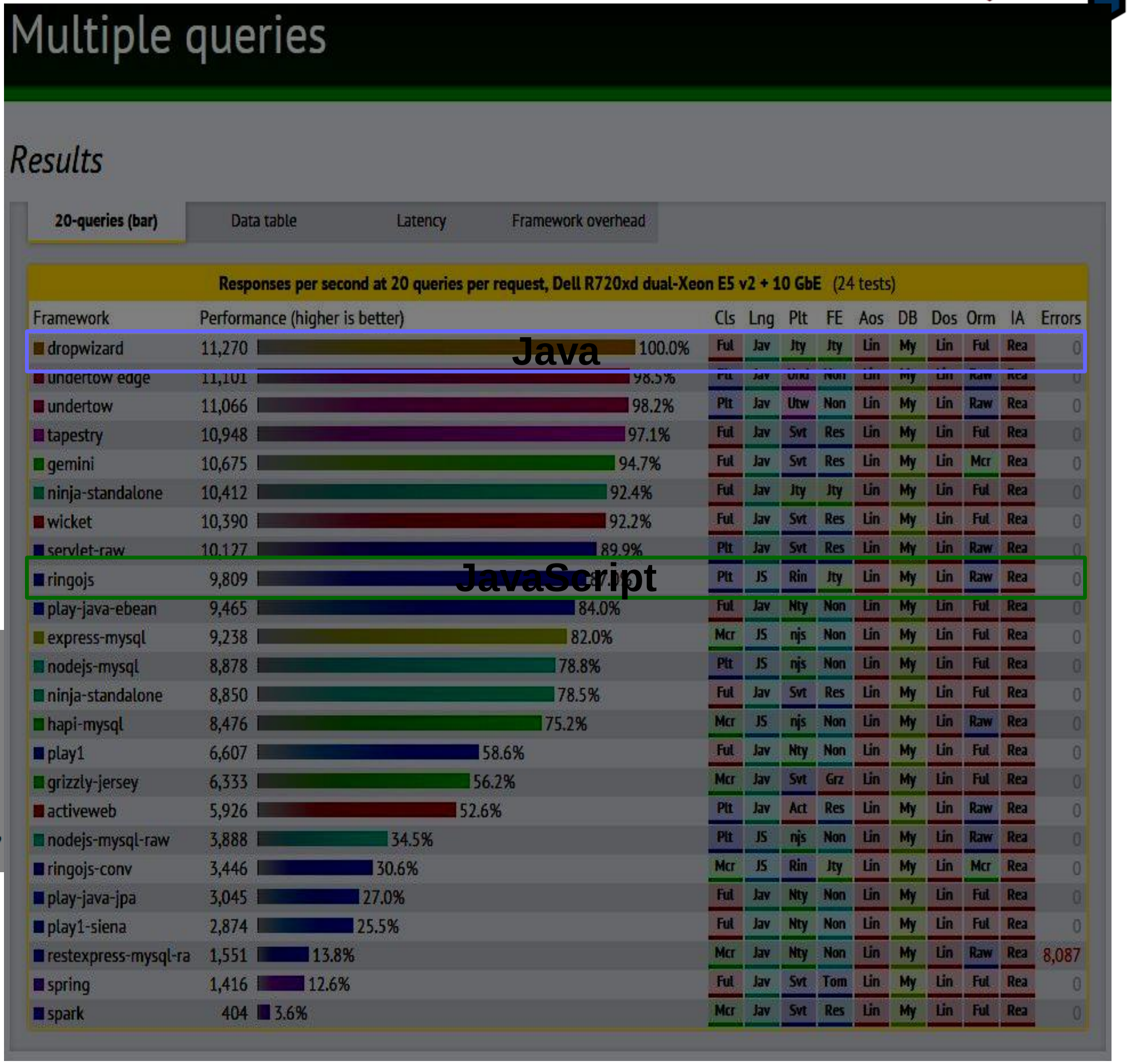
Multiple Queries

- Fetches multiple rows from a simple database table
- Rows serialized as JSON
- Example response:



```
HTTP/1.1 200 OK
Content-Length: 315
Content-Type: application/json; charset=UTF-8
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}]
```



Multiple queries

Results

Responses per second at 20 queries per request, Dell R720xd dual-Xeon E5 v2 + 10 GbE (24 tests)

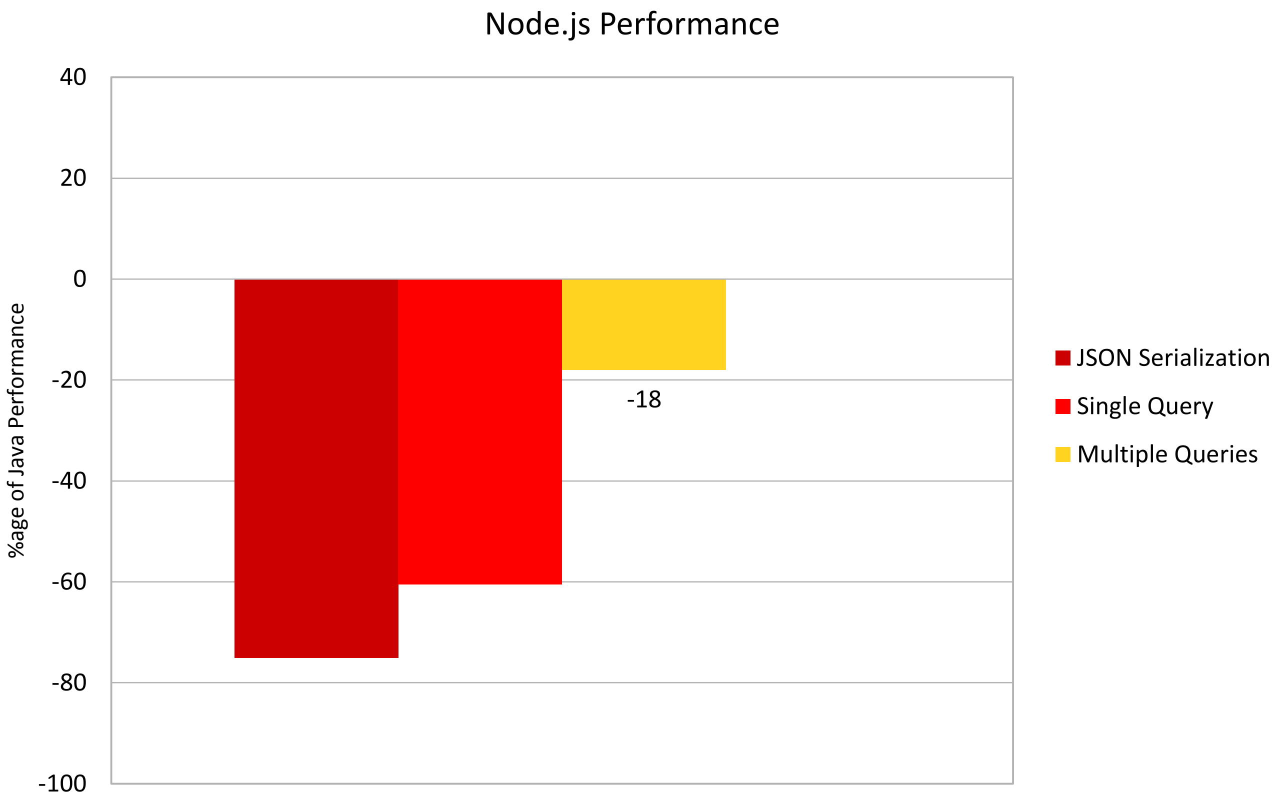
Framework	Performance (higher is better)		Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA	Errors
dropwizard	11,270	100.0%	Ful	Jav	Jty	Jty	Lin	My	Lin	Ful	Rea	0
undertow edge	11,101	98.5%	Plt	Jav	Und	Non	Lin	My	Lin	Raw	Rea	0
undertow	11,066	98.2%	Plt	Jav	Utw	Non	Lin	My	Lin	Raw	Rea	0
tapestry	10,948	97.1%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
gemini	10,675	94.7%	Ful	Jav	Svt	Res	Lin	My	Lin	Mcr	Rea	0
ninja-standalone	10,412	92.4%	Ful	Jav	Jty	Jty	Lin	My	Lin	Ful	Rea	0
wicket	10,390	92.2%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
servlet-raw	10,127	89.9%	Plt	Jav	Svt	Res	Lin	My	Lin	Raw	Rea	0
ringojs	9,809	87.0%	Plt	JS	Rin	Jty	Lin	My	Lin	Raw	Rea	0
play-java-ebean	9,465	84.0%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
express-mysql	9,238	82.0%	Mcr	JS	njs	Non	Lin	My	Lin	Ful	Rea	0
nodejs-mysql	8,878	78.8%	Plt	JS	njs	Non	Lin	My	Lin	Ful	Rea	0
ninja-standalone	8,850	78.5%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0
hapi-mysql	8,476	75.2%	Mcr	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
play1	6,607	58.6%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
grizzly-jersey	6,333	56.2%	Mcr	Jav	Svt	Grz	Lin	My	Lin	Ful	Rea	0
activeweb	5,926	52.6%	Plt	Jav	Act	Res	Lin	My	Lin	Raw	Rea	0
nodejs-mysql-raw	3,888	34.5%	Plt	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
ringojs-conv	3,446	30.6%	Mcr	JS	Rin	Jty	Lin	My	Lin	Mcr	Rea	0
play-java-jpa	3,045	27.0%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
play1-siena	2,874	25.5%	Ful	Jav	Nty	Non	Lin	My	Lin	Ful	Rea	0
restexpress-mysql-ra	1,551	13.8%	Mcr	Jav	Nty	Non	Lin	My	Lin	Raw	Rea	8,087
spring	1,416	12.6%	Ful	Jav	Svt	Tom	Lin	My	Lin	Ful	Rea	0
spark	404	3.6%	Mcr	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0

Java

JavaScript

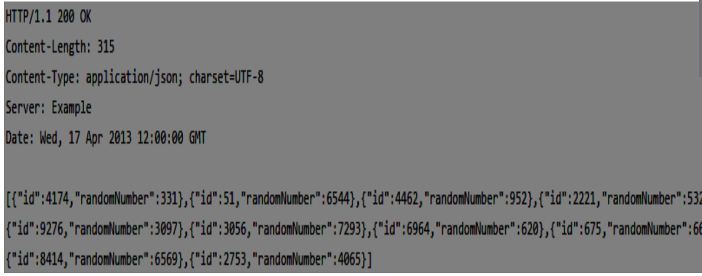
Results from TechEmpower.com Round 9 tests (2014-05-01)

JavaScript WebApp Performance



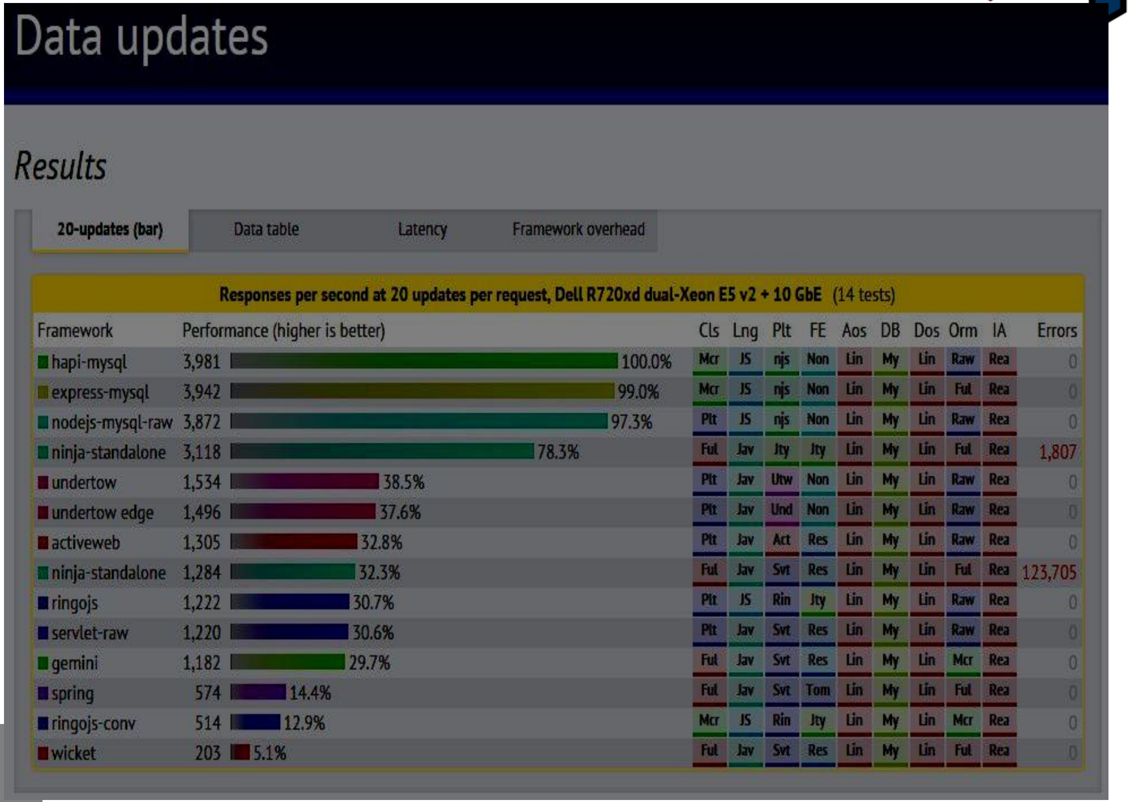
Data Updates

- Fetches multiple rows from a simple database table
- Converts rows to objects and modifies one attribute of each object
- Updates each associated row and serializes as JSON
- Example Response:



```
HTTP/1.1 200 OK
Content-Length: 315
Content-Type: application/json; charset=UTF-8
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}]
```



Data updates

Results

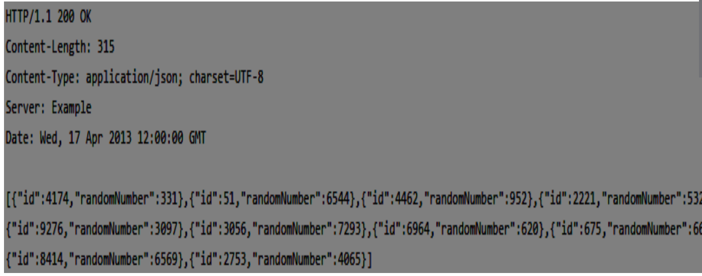
Responses per second at 20 updates per request, Dell R720xd dual-Xeon E5 v2 + 10 GbE (14 tests)

Framework	Performance (higher is better)		Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA	Errors
hapi-mysql	3,981	100.0%	Mcr	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
express-mysql	3,942	99.0%	Mcr	JS	njs	Non	Lin	My	Lin	Ful	Rea	0
nodejs-mysql-raw	3,872	97.3%	Plt	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
ninja-standalone	3,118	78.3%	Ful	Jav	Jty	Jty	Lin	My	Lin	Ful	Rea	1,807
undertow	1,534	38.5%	Plt	Jav	Utw	Non	Lin	My	Lin	Raw	Rea	0
undertow edge	1,496	37.6%	Plt	Jav	Und	Non	Lin	My	Lin	Raw	Rea	0
activeweb	1,305	32.8%	Plt	Jav	Act	Res	Lin	My	Lin	Raw	Rea	0
ninja-standalone	1,284	32.3%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	123,705
ringojs	1,222	30.7%	Plt	JS	Rin	Jty	Lin	My	Lin	Raw	Rea	0
servlet-raw	1,220	30.6%	Plt	Jav	Svt	Res	Lin	My	Lin	Raw	Rea	0
gemini	1,182	29.7%	Ful	Jav	Svt	Res	Lin	My	Lin	Mcr	Rea	0
spring	574	14.4%	Ful	Jav	Svt	Tom	Lin	My	Lin	Ful	Rea	0
ringojs-conv	514	12.9%	Mcr	JS	Rin	Jty	Lin	My	Lin	Mcr	Rea	0
wicket	203	5.1%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0

Results from TechEmpower.com Round 9 tests (2014-05-01)

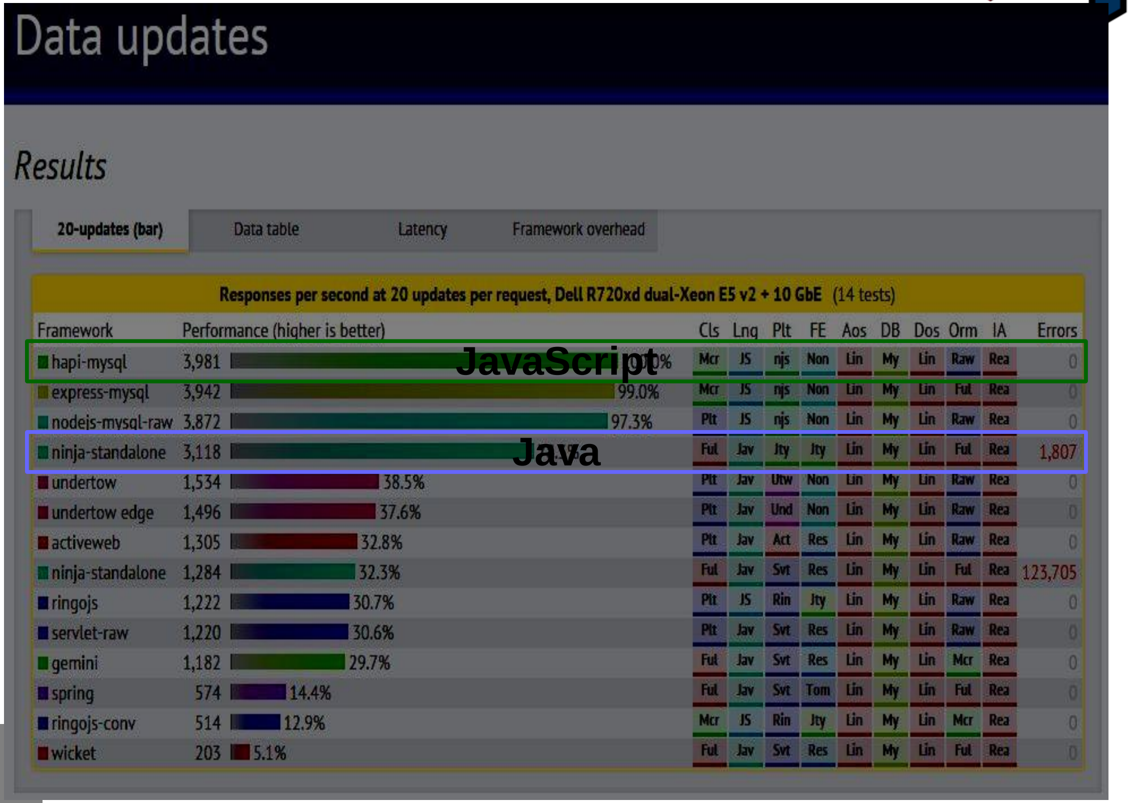
Data Updates

- Fetches multiple rows from a simple database table
- Converts rows to objects and modifies one attribute of each object
- Updates each associated row and serializes as JSON
- Example Response:



```
HTTP/1.1 200 OK
Content-Length: 315
Content-Type: application/json; charset=UTF-8
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}]
```



Data updates

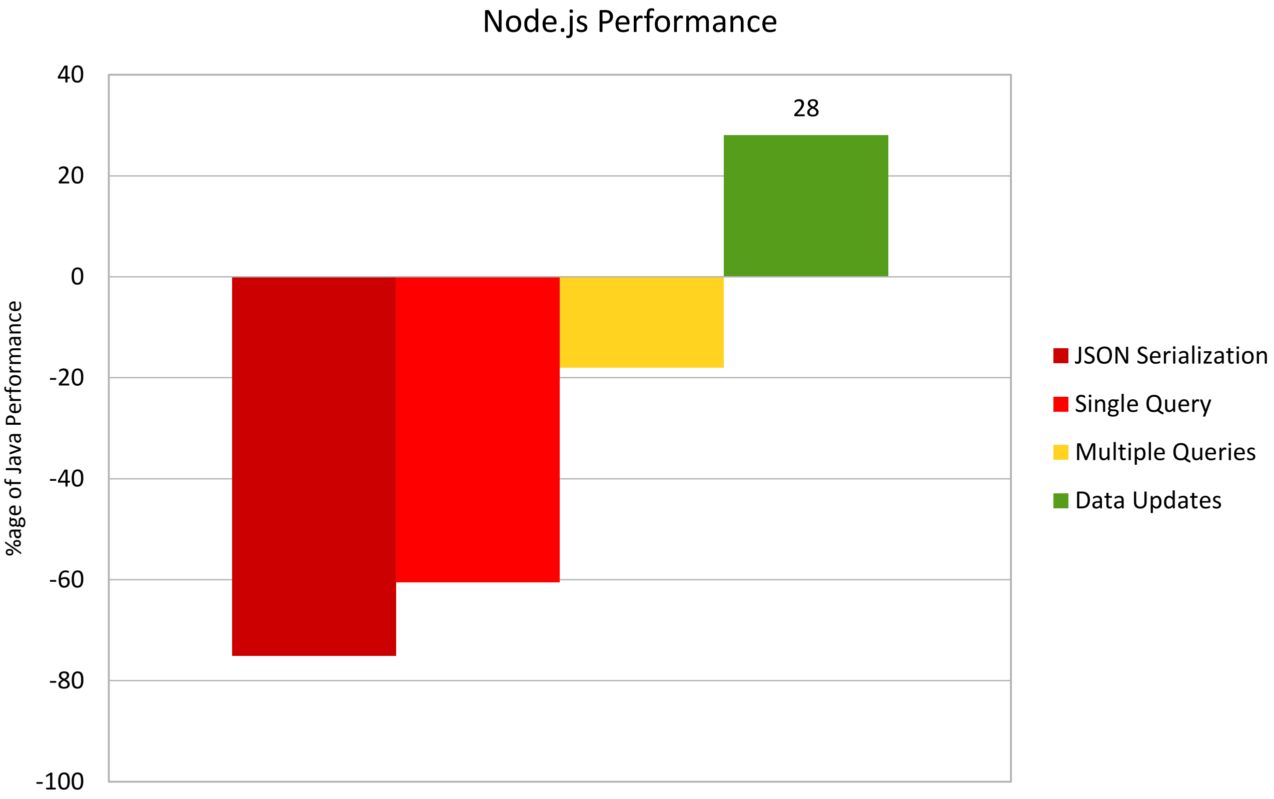
Results

Responses per second at 20 updates per request, Dell R720xd dual-Xeon E5 v2 + 10 GbE (14 tests)

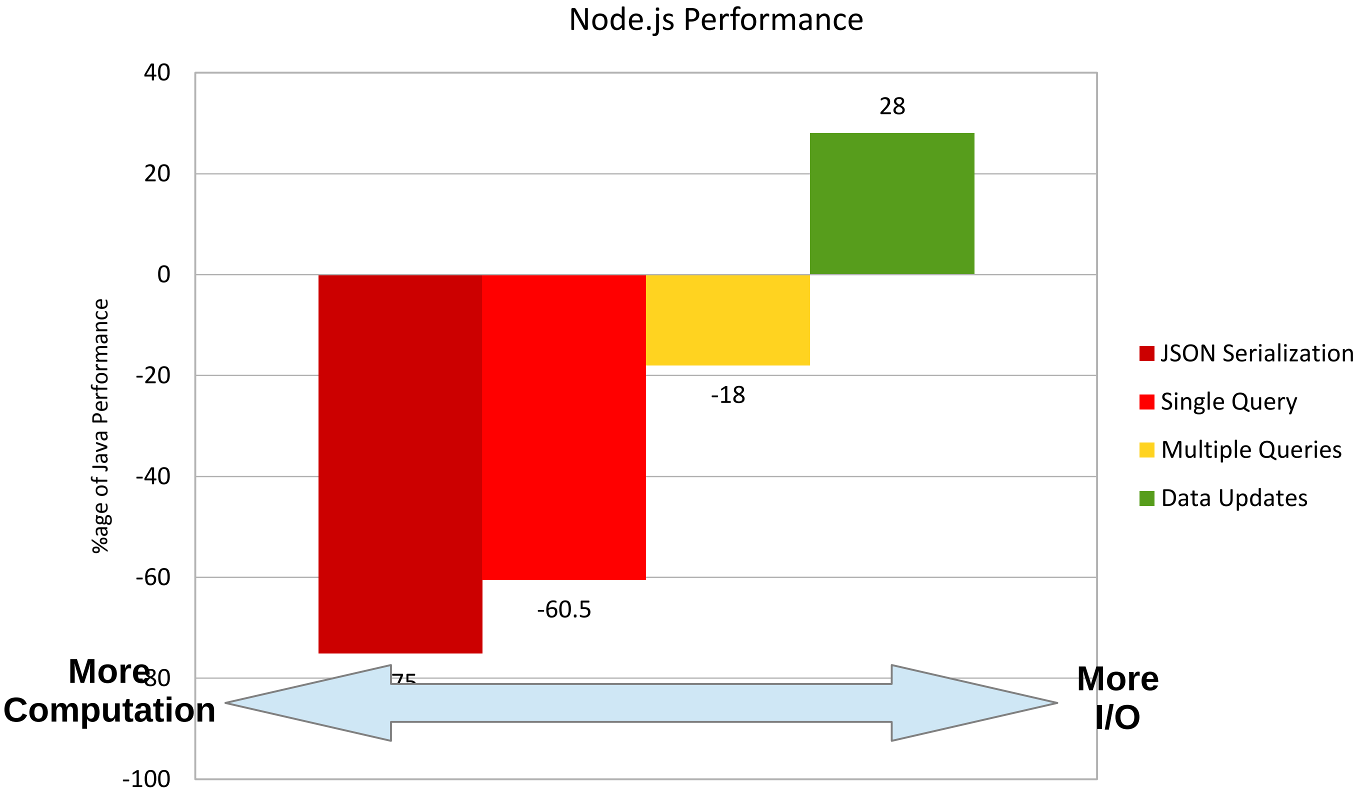
Framework	Performance (higher is better)		Cls	Lng	Plt	FE	Aos	DB	Dos	Orm	IA	Errors
hapi-mysql	3,981	100.0%	Mcr	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
express-mysql	3,942	99.0%	Mcr	JS	njs	Non	Lin	My	Lin	Ful	Rea	0
nodejs-mysql-raw	3,872	97.3%	Plt	JS	njs	Non	Lin	My	Lin	Raw	Rea	0
ninja-standalone	3,118	[illegible]	Ful	Jav	Jty	Jty	Lin	My	Lin	Ful	Rea	1,807
undertow	1,534	38.5%	Plt	Jav	Utw	Non	Lin	My	Lin	Raw	Rea	0
undertow edge	1,496	37.6%	Plt	Jav	Und	Non	Lin	My	Lin	Raw	Rea	0
activeweb	1,305	32.8%	Plt	Jav	Act	Res	Lin	My	Lin	Raw	Rea	0
ninja-standalone	1,284	32.3%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	123,705
ringojs	1,222	30.7%	Plt	JS	Rin	Jty	Lin	My	Lin	Raw	Rea	0
servlet-raw	1,220	30.6%	Plt	Jav	Svt	Res	Lin	My	Lin	Raw	Rea	0
gemini	1,182	29.7%	Ful	Jav	Svt	Res	Lin	My	Lin	Mcr	Rea	0
spring	574	14.4%	Ful	Jav	Svt	Tom	Lin	My	Lin	Ful	Rea	0
ringojs-conv	514	12.9%	Mcr	JS	Rin	Jty	Lin	My	Lin	Mcr	Rea	0
wicket	203	5.1%	Ful	Jav	Svt	Res	Lin	My	Lin	Ful	Rea	0

Results from TechEmpower.com Round 9 tests (2014-05-01)

JavaScript WebApp Performance



JavaScript WebApp Performance

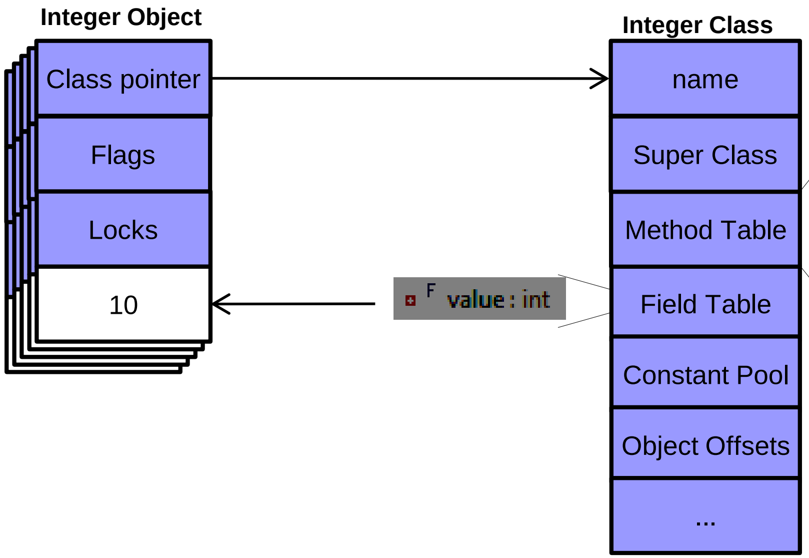


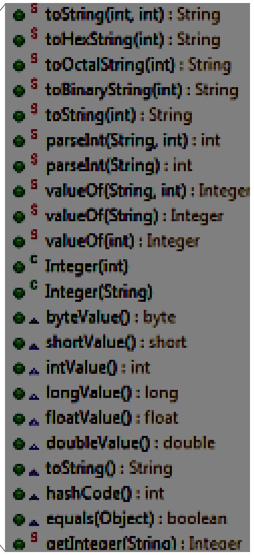
- Computation speed is (much) slower than Java
- I/O speed is higher than Java

Under the Hood

Object Representation: Java

- Java objects are ***fixed*** in size and shape



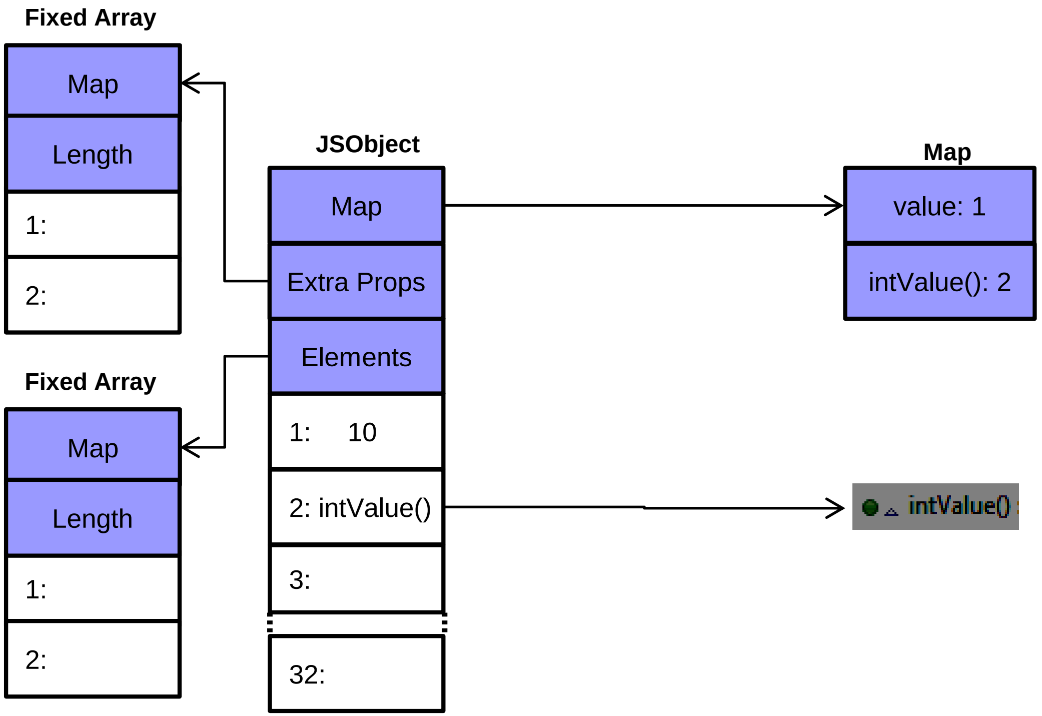


```
toString(int, int) : String
toHexString(int) : String
toOctalString(int) : String
toBinaryString(int) : String
toString(int) : String
parseInt(String, int) : int
parseInt(String) : int
valueOf(String, int) : Integer
valueOf(String) : Integer
valueOf(int) : Integer
Integer(int)
Integer(String)
byteValue() : byte
shortValue() : short
intValue() : int
longValue() : long
floatValue() : float
doubleValue() : double
toString() : String
hashCode() : int
equals(Object) : boolean
getInteger(String) : Integer
```

- Values associated with objects are fixed and typed (known what and where it is)
- Methods associated with objects are fixed and typed (parameters and return types)

Object Representation: JavaScript

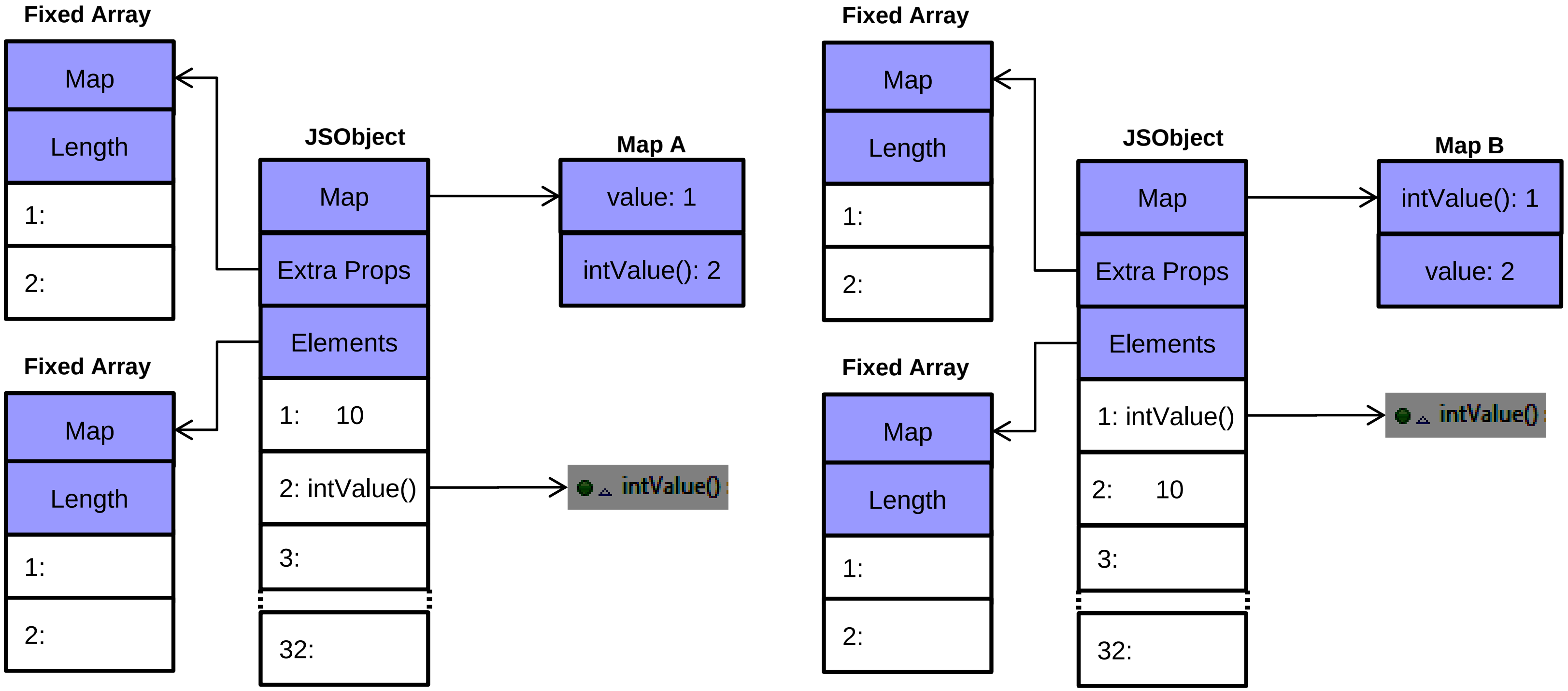
- JavaScript objects are ***dynamic*** in size and shape



- Values associated with objects are dynamic and un-typed
- Methods associated with objects are dynamic and un-typed
  - 32 “slots” exist for method and values with overflow arrays if this is not enough
  - Every “slot” is 64bits as any type of data could be stored there

Object Representation: JavaScript

- Order in which methods and fields are added matters



- Objects are equivalent and equal, but have different Maps and layouts

JIT Compilation

- Functions are stored in JavaScript objects as fields
  - No fixed set of methods for an object

- Objects are not typed, so data much be checked to determine how to handle it

eg. the '+' operator:

- number + number → addition
- string involved? → concatenation
- objects involved? → convert to primitives then addition or concatenation

eg. property load:

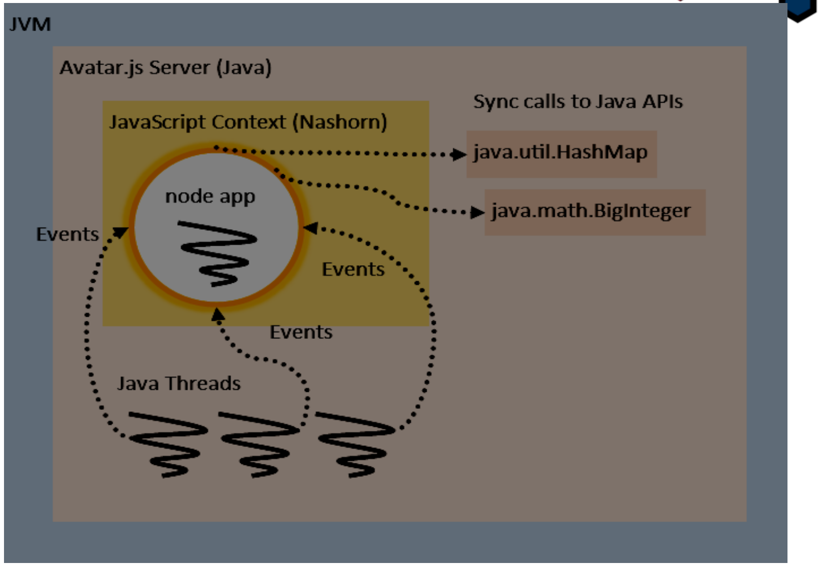
- Load prototype object
- Load getter method
- Load callback function

- Therefore not possible to determine what instructions to use just from the source code

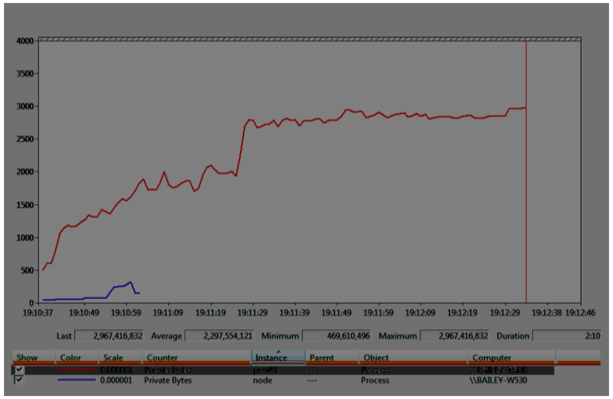
JavaScript on the JVM?

Nashorn and Avatar.js

- Nashorn JavaScript engine delivered in JDK8
  - Utilizes new JVM level features for performance
- Avatar.js provides Node.js support on Nashorn



- Results of “Octane” JavaScript benchmark*:
  - Node.js is 4.8x faster
  - Avatar.js is >10x larger



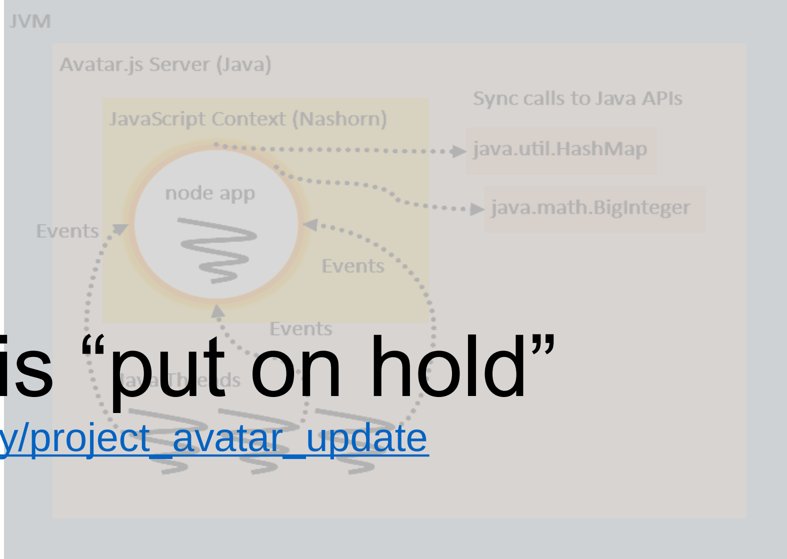
Last 2,967,416,832 Average 2,297,554,121 Minimum 469,610,496 Maximum 2,967,416,832 Duration 2:10

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
☑	🟥	0.000001	Private Bytes	java#3	---	Process	\\BAILEY-W530
☑	🟦	0.000001	Private Bytes	node	---	Process	\\BAILEY-W530

* Using Java 8 pre-u20

Nashorn and Avatar.js

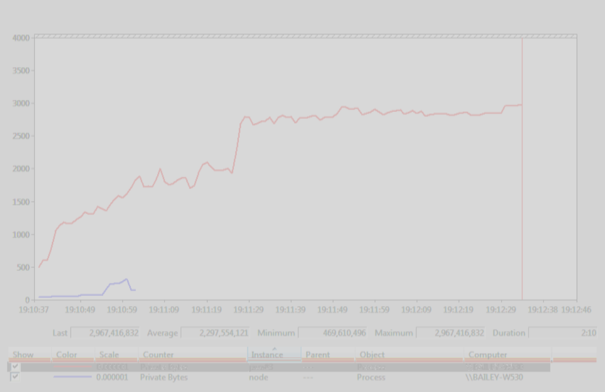
- Nashorn JavaScript engine delivered in JDK8
  - Utilizes new JVM level features for performance
- Avatar.js provides Node.js support on Nashorn



Feb 12th, 2015: Avatar is “put on hold”

https://blogs.oracle.com/theaquarium/entry/project_avatar_update

- Results of “Octane” JavaScript benchmark*:
  - Node.js is 4.8x faster
  - Avatar.js is >10x larger



Last 2,967,416,832 Average 2,297,554,121 Minimum 469,610,496 Maximum 2,967,416,832 Duration 2:10

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
☑	🟥	[illegible]	Private Bytes	[illegible]	---	Process	[illegible]
☑	🟦	0.000001	Private Bytes	node	---	Process	\\BAILEY-W530

* Using Java 8 pre-u40

Enterprise Deployments

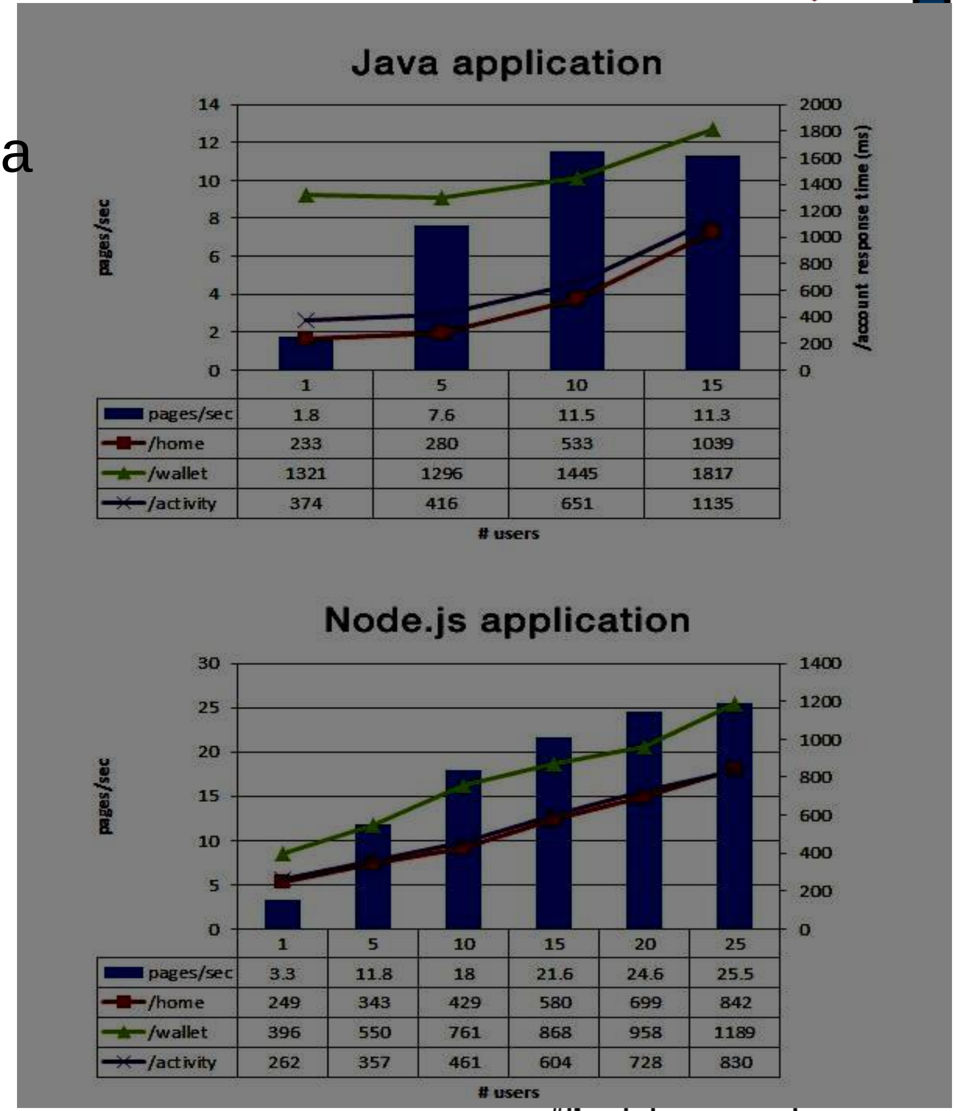
The PayPal Story

PayPal and “Account Overview” Project

- 2013: PayPal evaluates use of Node.js for “Account Overview”
  - Implementation done in both Java and Node.js to compare

- Node.js implementation
  - 50% less development effort
  - 33% fewer lines of code
  - 40% fewer files
  - ~35% faster request response

- **Note:** legacy Java frameworks involved.....



Java application

	1	5	10	15
pages/sec	1.8	7.6	11.5	11.3
/home	233	280	533	1039
/wallet	1321	1296	1445	1817
/activity	374	416	651	1135

Node.js application

	1	5	10	15	20	25
pages/sec	3.3	11.8	18	21.6	24.6	25.5
/home	249	343	429	580	699	842
/wallet	396	550	761	868	958	1189
/activity	262	357	461	604	728	830

The WalMart Story

WalMart experiences Node.js memory leak

- 2013: Eran Hammer (WalMart) discovers 200+MB/day leak
  - Increasing memory usage at 200+MB/day per server

- Application improvements by Eran reduces leak to 8MB/day
  - Lots of progress made
  - But required months of investigation effort

- Identified remaining leak related to HTTP Client Requests
  - Unable to make further progress....

- Node.js runtime development team required to resolve issue
  - 5 core runtime developers/engineers
  - 3 weeks of effort

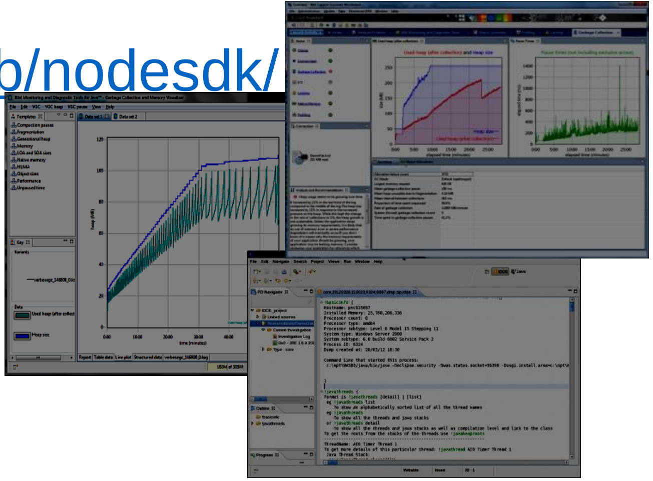
IBM and Node.js

IBM and Node.js

- Node.js Foundation Founding Member
  - Alongside Joyent, Linux Foundation, Microsoft, PayPal and Fidelity

- IBM SDK for Node.js v1.2
  - Open source ports of Google V8 JavaScript engine
    - Support for POWER and zLinux
  - Runtimes available for all platforms to provide consistency
    - AIX, Linux (Intel, POWER, System z, Windows, Mac OS X)
  - http://www.ibm.com/developerworks/web/nodesdk/

- IBM Monitoring and Diagnostics Tools
  - Live monitoring: Health Center
  - GC log analysis: GCMV
  - Dump analysis: IDDE



Summary

- JavaScript has a large amount of interest and is growing
  - Web applications with code sharing between server and browser
  - Async IO and event loop makes it easy to write scalable applications
  - Rich set of APIs available via the npm module ecosystem

- Dynamic nature makes development easier, but introduces challenges
  - Errors typically found during compilation are found at runtime
  - JIT compilation loves certainty, which is removed

- Additional “enterprise-grade” capabilities needed
  - Monitoring/Diagnostics, Security, Internationalization, etc
  - IBM contributing via the newly announce Node Foundation

Notices and Disclaimers

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS document is distributed "AS IS" without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and Disclaimers (con't)

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

- IBM, the IBM logo, ibm.com, Bluemix, Blueworks Live, CICS, Clearcase, DOORS®, Enterprise Document Management System™, Global Business Services ®, Global Technology Services ®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, SoDA, SPSS, StoredIQ, Tivoli®, Trusteer®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Thank You

Your Feedback is Important!

Access the InterConnect 2015 Conference CONNECT Attendee Portal to complete your session surveys from your smartphone, laptop or conference kiosk.

InterConnect2015

The Premier Cloud & Mobile Conference

InterConnect2015