#ibminterconnect

Java and Other Languages

Chris Bailey: STSM, IBM Runtime Monitoring

IBM

InterConnect2015

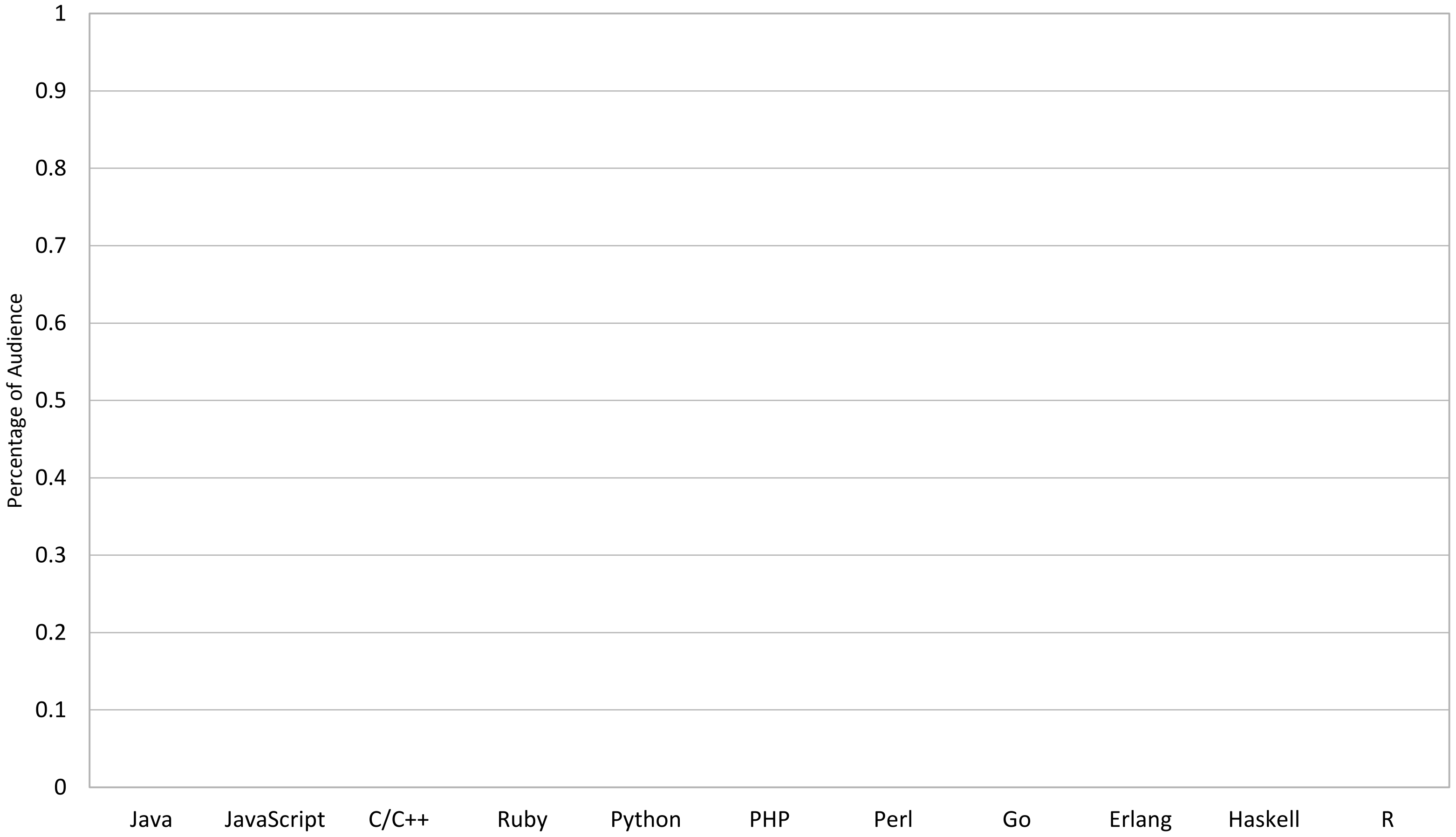
The Premier Cloud & Mobile Conference

February 22 – 26

MGM Grand & Mandalay Bay | Las Vegas, Nevada

A Quick Survey

What languages do you use?

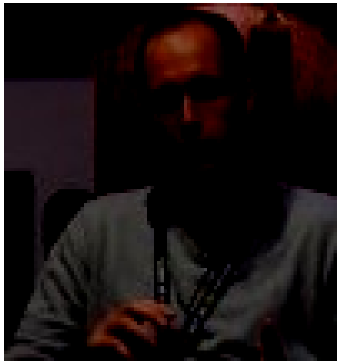


Introduction to the Speakers

- **Chris Baley**

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:

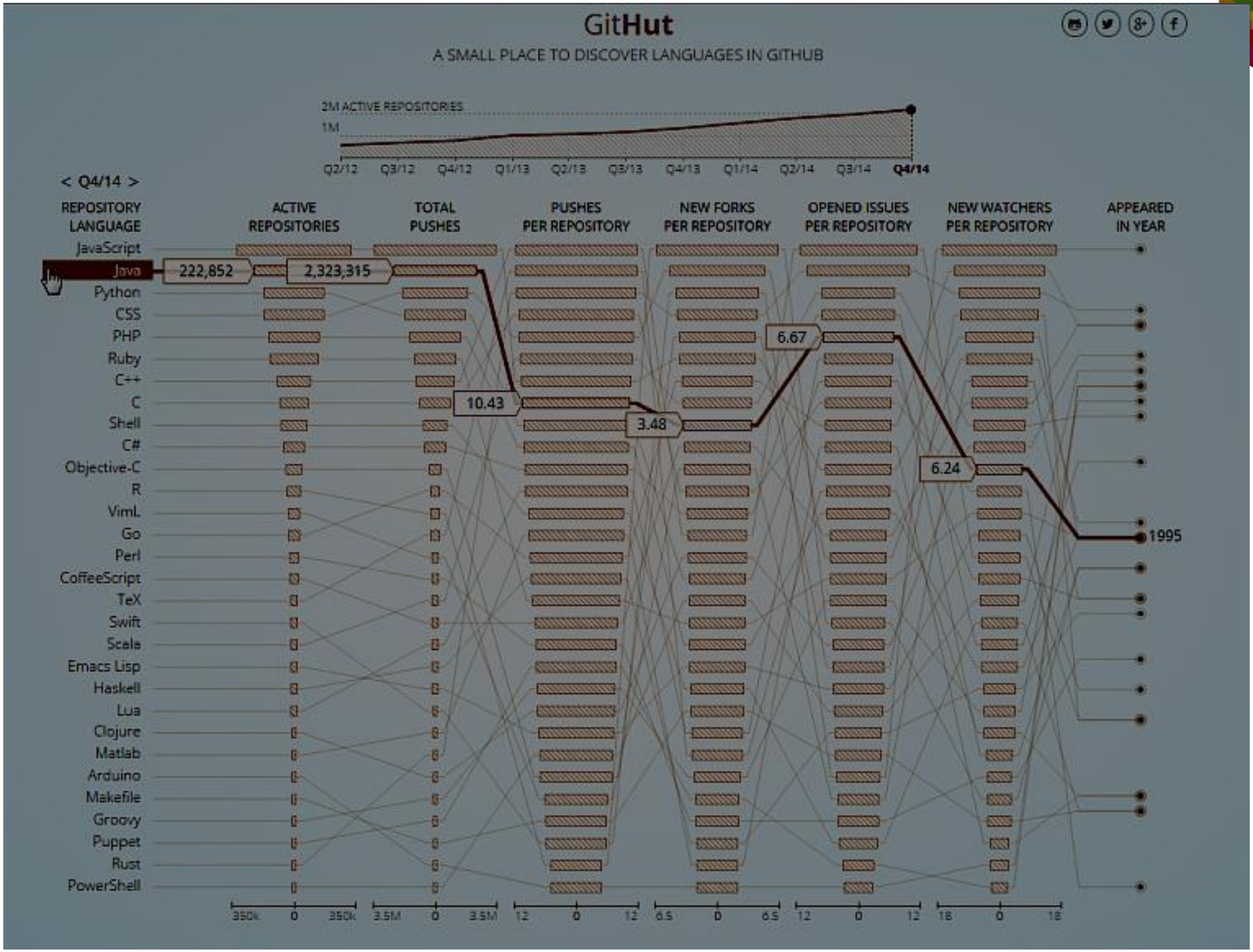
-Java monitoring, diagnostics and troubleshooting
-Java integration into the cloud
-JavaScript monitoring, diagnostics and troubleshooting

- My contact information:

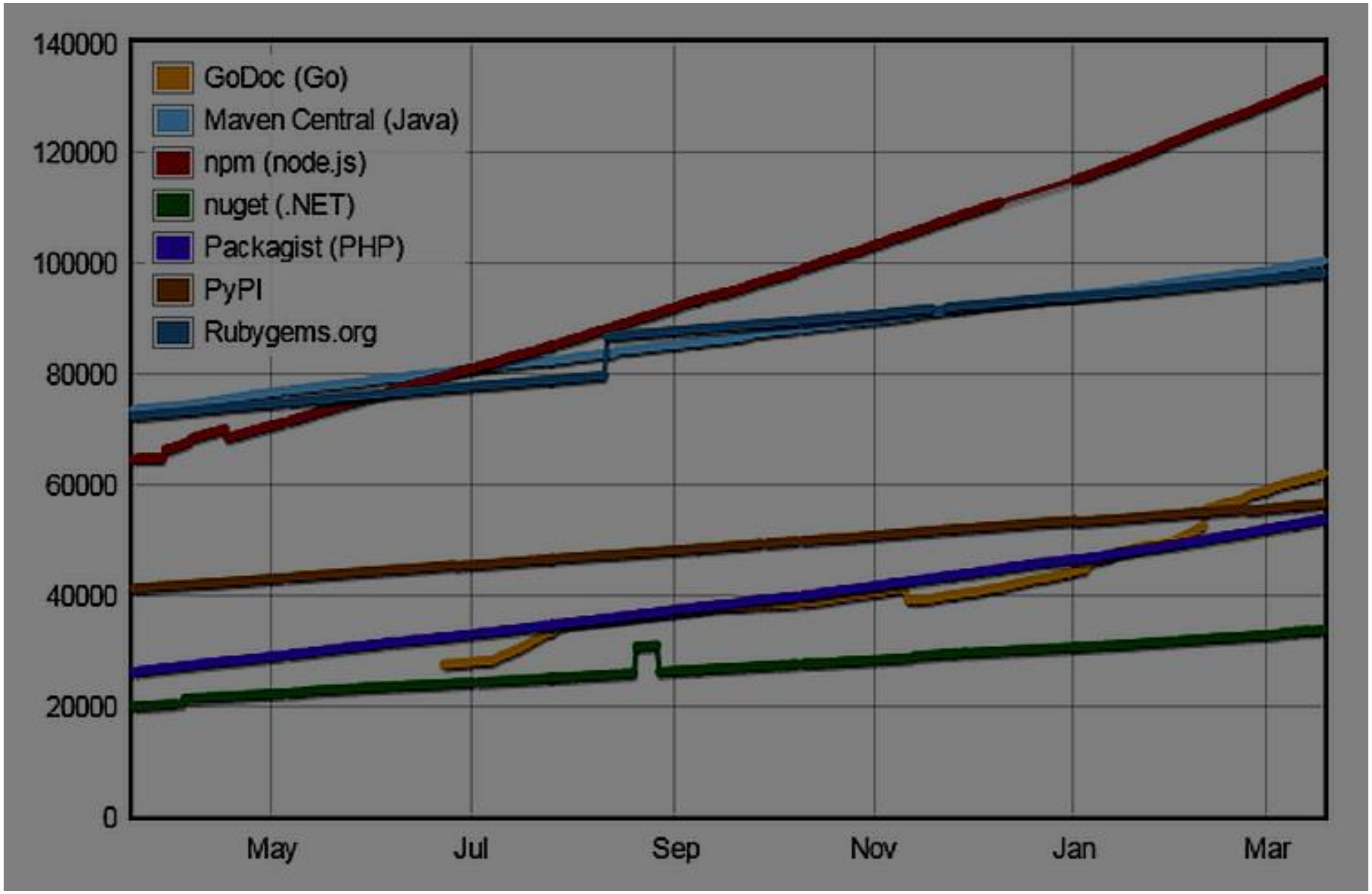
-baileyc@uk.ibm.com
-http://www.linkedin.com/in/chrisbaileyibm
-http://www.slideshare.net/cnbailey/
-@Chris__Bailey

Language Adoption

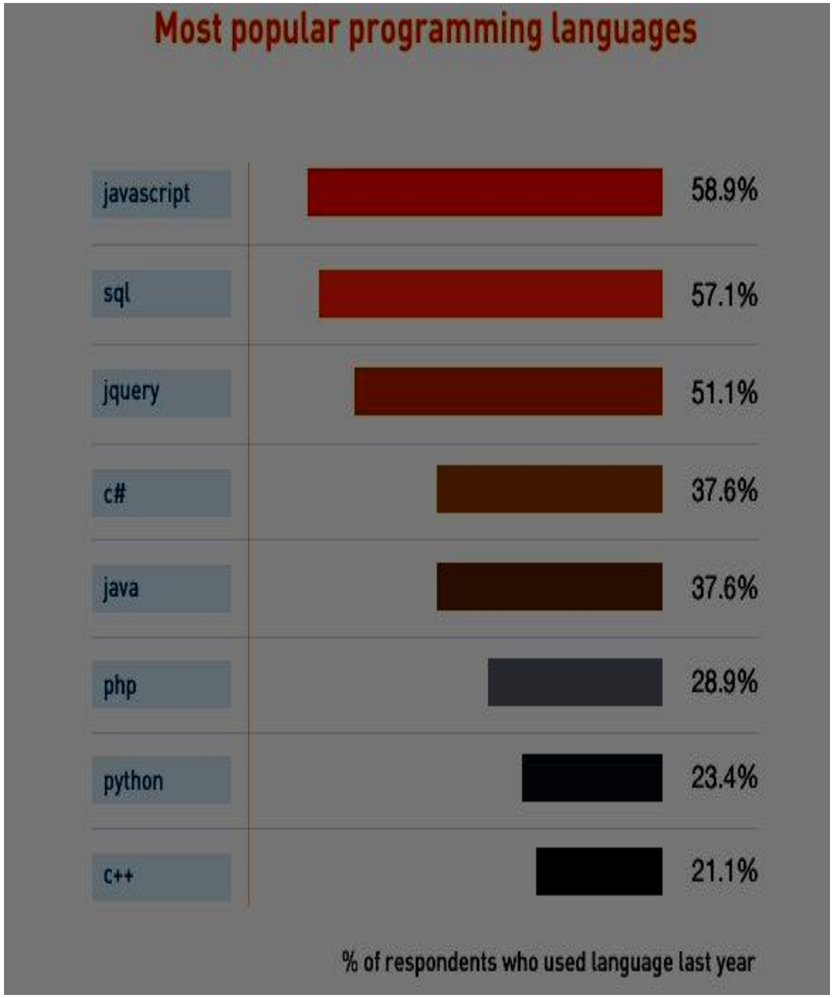
GitHub Adoption: Java



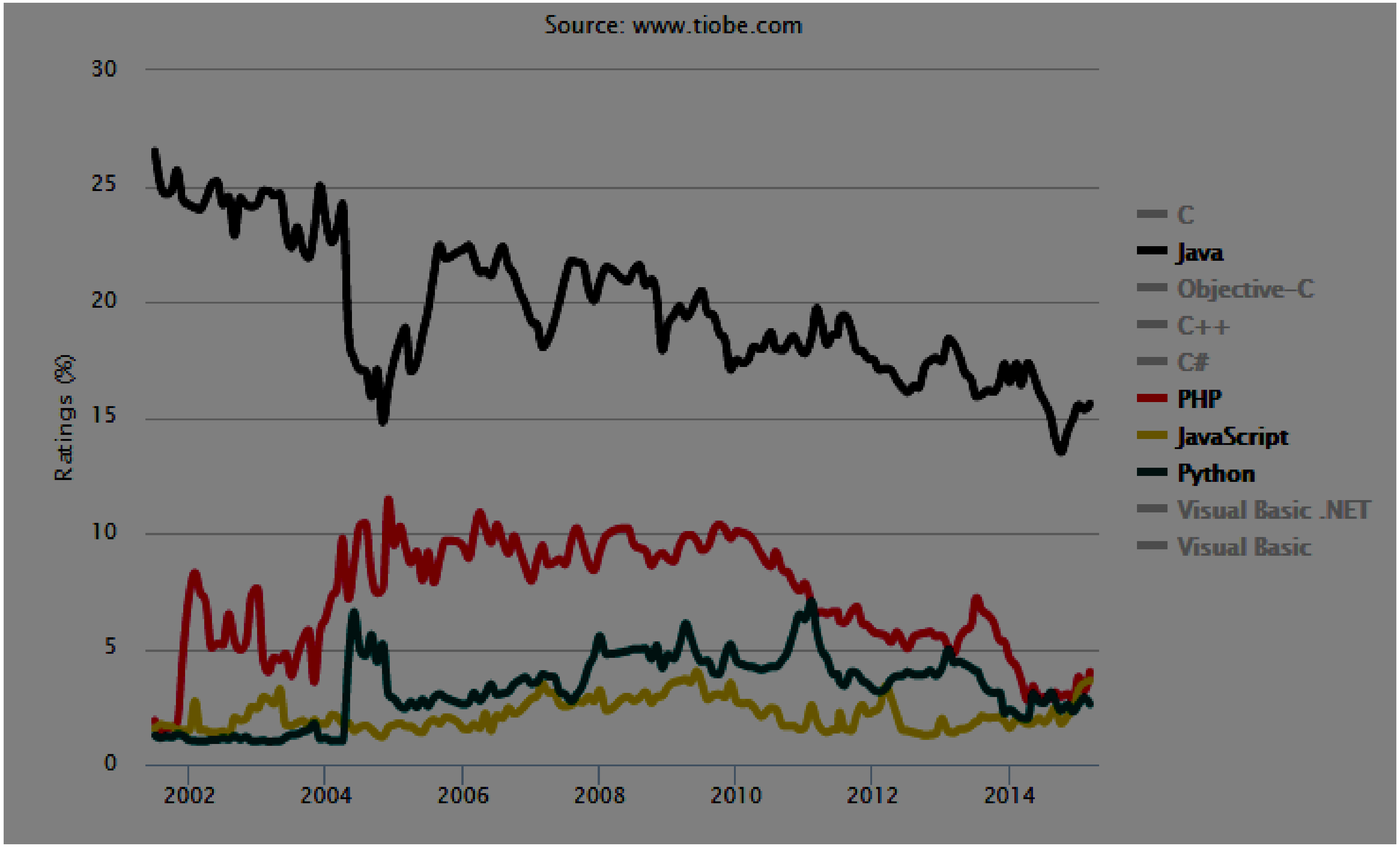
modulecounts.com



StackOverflow User Survey



Tiobe Community Programming Index

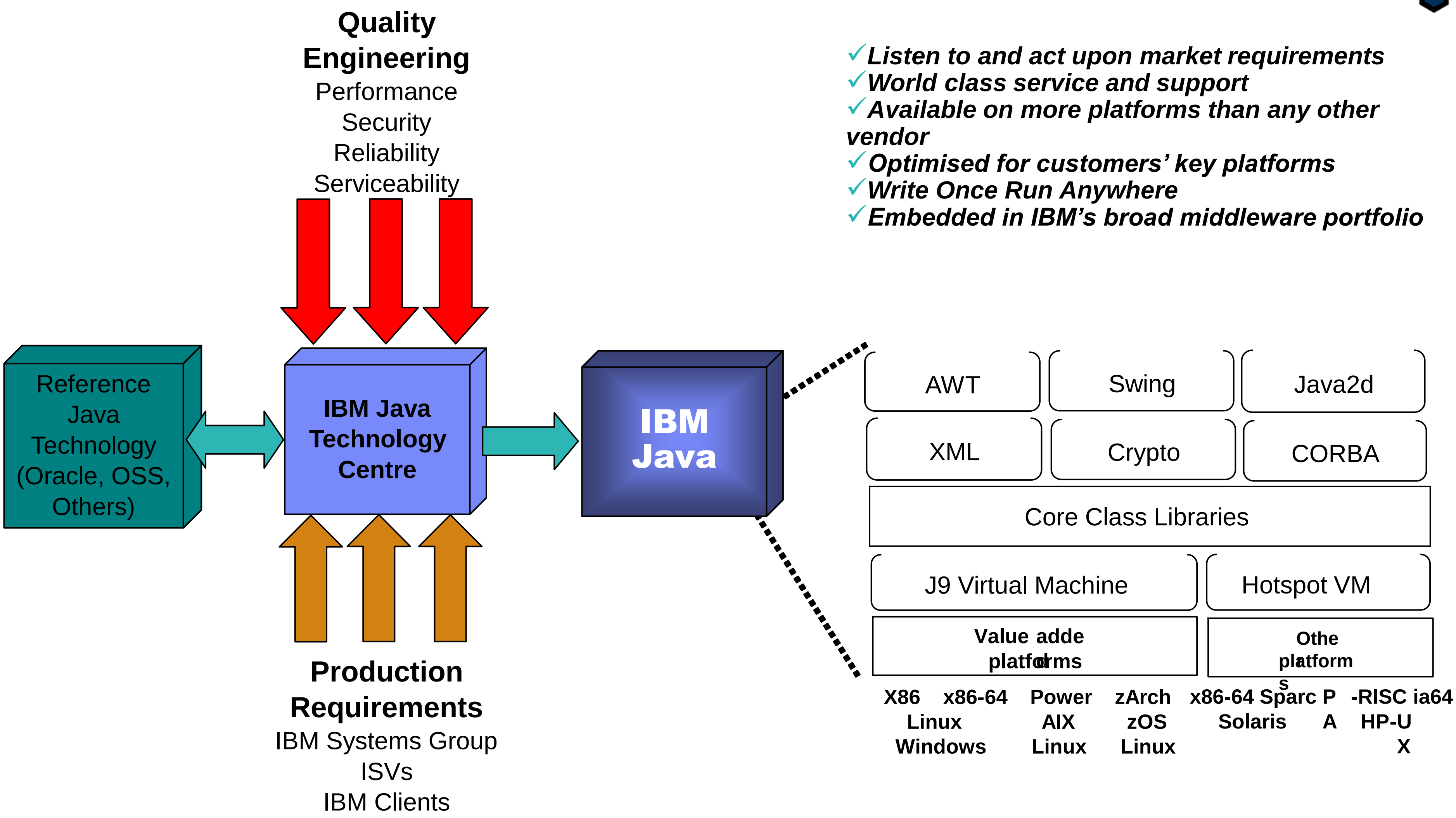


Ratings based on the number of skilled engineers, courses and third party vendors.

Java and Other Languages

IBM SDK for Java

- ✓ ***Listen to and act upon market requirements***
- ✓ ***World class service and support***
- ✓ ***Available on more platforms than any other vendor***
- ✓ ***Optimised for customers' key platforms***
- ✓ ***Write Once Run Anywhere***
- ✓ ***Embedded in IBM's broad middleware portfolio***



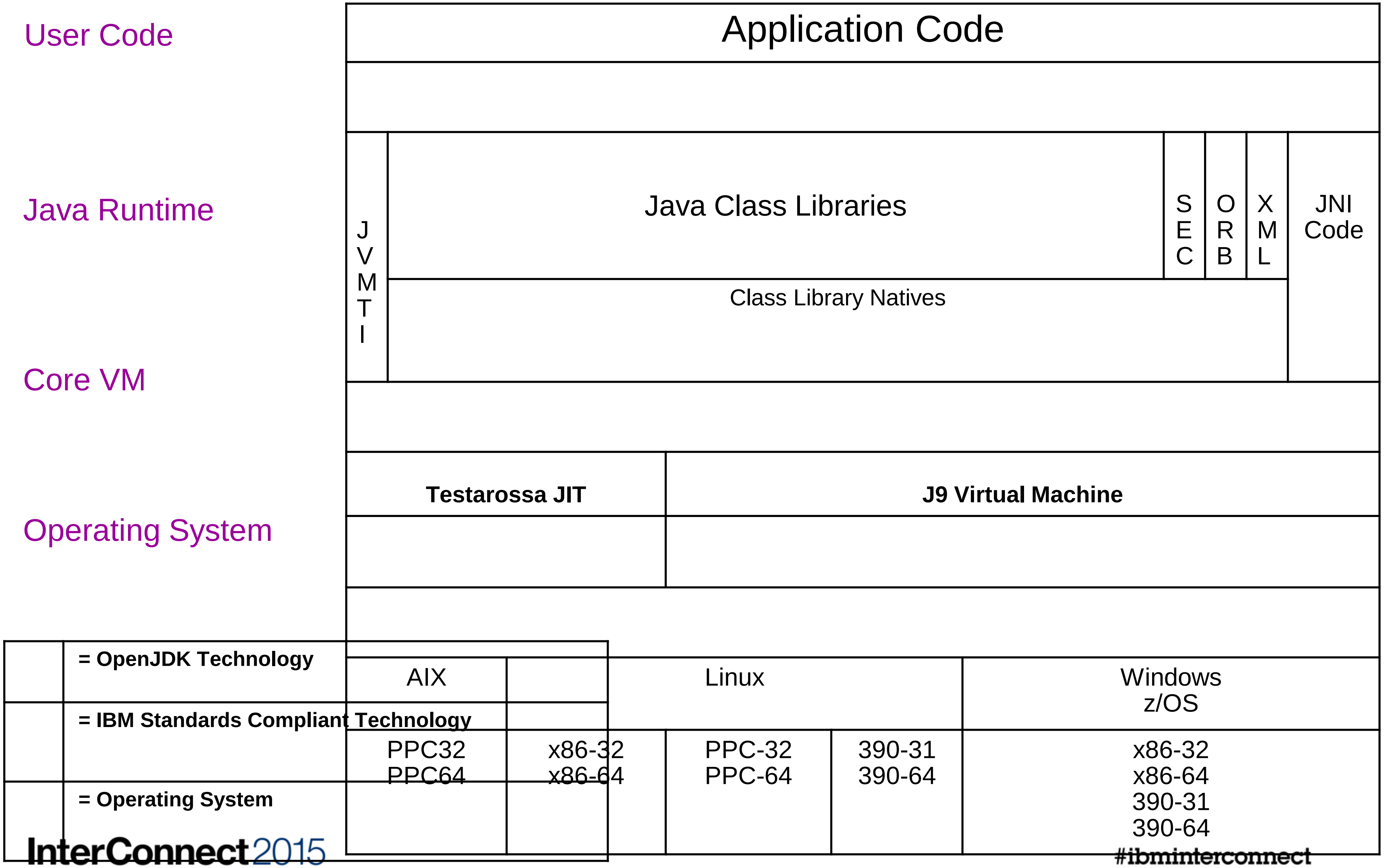
Relationship between IBM, Oracle and OpenJDKs

User Code

Java Runtime

Core VM

Operating System



Application Code

J V M T I

Java Class Libraries

S E C

O R B

X M L

JNI Code

Class Library Natives

Testarossa JIT

J9 Virtual Machine

AIX

Linux

Windows
z/OS

PPC32
PPC64

x86-32
x86-64

PPC-32
PPC-64

390-31
390-64

x86-32
x86-64
390-31
390-64

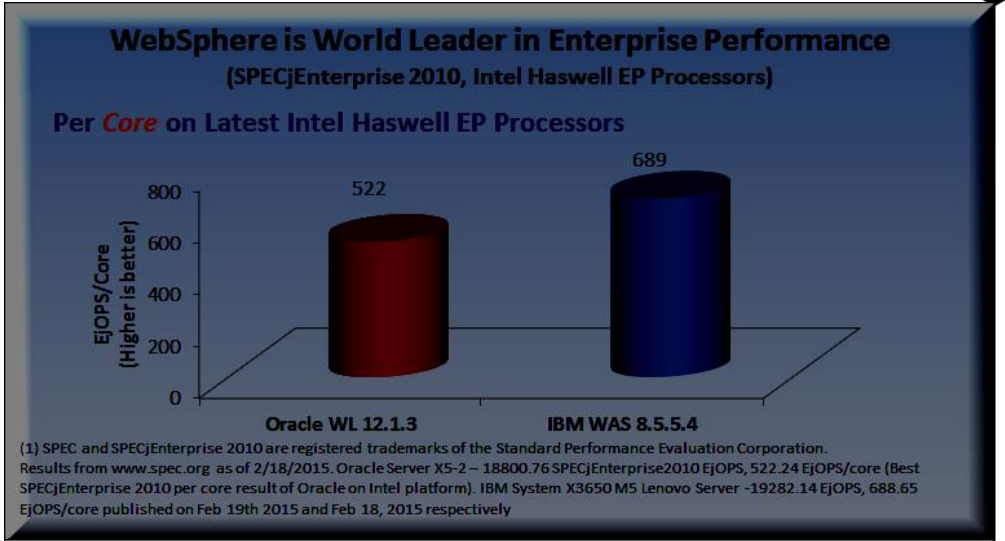
= OpenJDK Technology

= IBM Standards Compliant Technology

= Operating System

IBM SDK for Java in WAS 8.5.5

- WAS 8.5.5.4 outperforms WL 12.1.3 by 32% on per core basis and retains Industry leadership on SPECjEnterprise 2010 Benchmark results published on latest Intel Haswell EP Processors



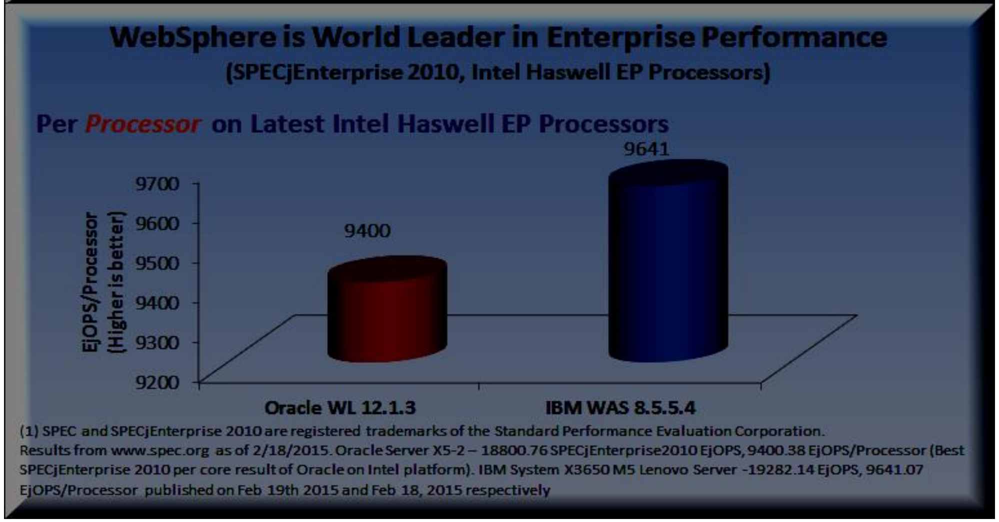
WebSphere is World Leader in Enterprise Performance

(SPECjEnterprise 2010, Intel Haswell EP Processors)

Per Core on Latest Intel Haswell EP Processors

(1) SPEC and SPECjEnterprise 2010 are registered trademarks of the Standard Performance Evaluation Corporation. Results from www.spec.org as of 2/18/2015. Oracle Server X5-2 – 18800.76 SPECjEnterprise2010 EjOPS, 522.24 EjOPS/core (Best SPECjEnterprise 2010 per core result of Oracle on Intel platform). IBM System X3650 M5 Lenovo Server -19282.14 EjOPS, 688.65 EjOPS/core published on Feb 19th 2015 and Feb 18, 2015 respectively

- WAS leads on per Processor performance as well beating WL 12.1.3 on the latest Intel Haswell EP processors as per results published on SPEC



WebSphere is World Leader in Enterprise Performance

(SPECjEnterprise 2010, Intel Haswell EP Processors)

Per Processor on Latest Intel Haswell EP Processors

(1) SPEC and SPECjEnterprise 2010 are registered trademarks of the Standard Performance Evaluation Corporation. Results from www.spec.org as of 2/18/2015. Oracle Server X5-2 – 18800.76 SPECjEnterprise2010 EjOPS, 9400.38 EjOPS/Processor (Best SPECjEnterprise 2010 per core result of Oracle on Intel platform). IBM System X3650 M5 Lenovo Server -19282.14 EjOPS, 9641.07 EjOPS/Processor published on Feb 19th 2015 and Feb 18, 2015 respectively

GCMV

Offline memory monitoring capability for:

- Garbage Collection (GC)
- Operating System memory usage

Provides ability to:

- Monitor process and system memory usage
- Identify application memory usage and leaks
- Analyse and tune GC performance

Visualization provided via Eclipse Client UI

- Available from Eclipse Marketplace
- Available from IBM Support Assistant

Headless/batch mode

- Run automated analysis

Health Center

Eclipse UI for performance monitoring

- Available from Eclipse Marketplace
- Available from Liberty Repository
- Available from IBM Support Assistant

Provides insight into Operating System

- Environment, CPU, Memory

Provides insight into Runtimes:

- GC, Class loading, Memory usage, Threads

Provides insight into Runtime view of application:

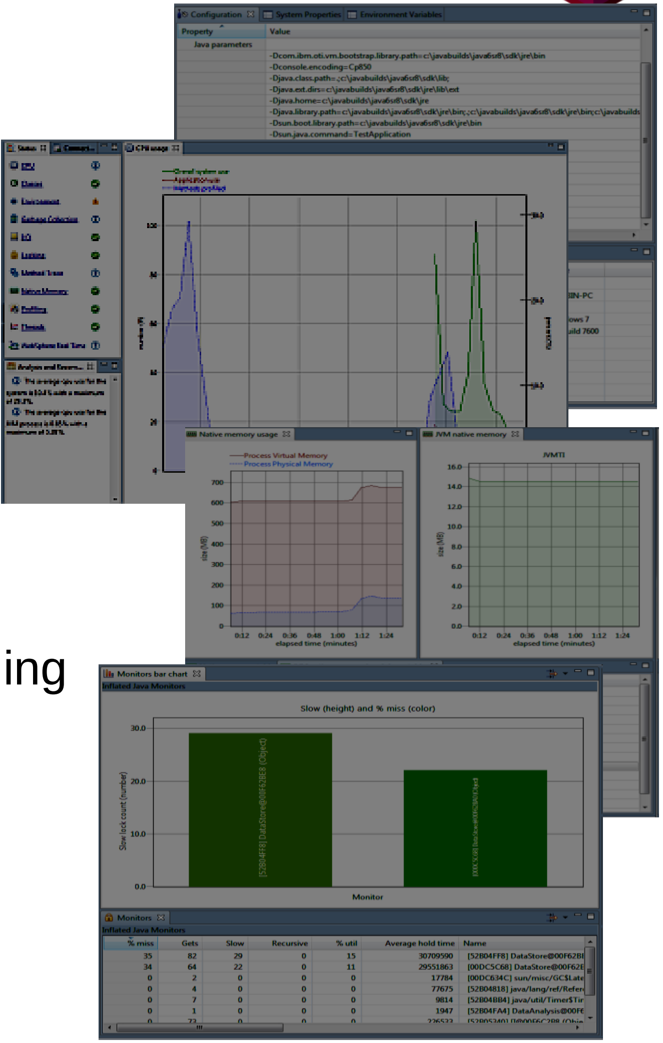
- Method profiling, Allocation profiling, Lock profiling

Provides diagnostic control

- Dump request, trace control, profiling control

Provides API to create custom tools

- Full JavaDoc in Knowledge Centre



Property	Value
Java parameters	
	-Dcom.ibm.oti.vm.bootstrap.library.path=c:\javabuilds\java6sr8\sdk\jre\bin
	-Dconsole.encoding=Cp850
	-Djava.class.path=.;c:\javabuilds\java6sr8\sdk\lib;
	-Djava.ext.dirs=c:\javabuilds\java6sr8\sdk\jre\lib\ext
	-Djava.home=c:\javabuilds\java6sr8\sdk\jre
	-Djava.library.path=c:\javabuilds\java6sr8\sdk\jre\bin;.;c:\javabuilds\java6sr8\sdk\jre\bin;c:\javabuilds
	-Dsun.boot.library.path=c:\javabuilds\java6sr8\sdk\jre\bin
	-Dsun.java.command=TestApplication

Inflated Java Monitors

% miss	Gets	Slow	Recursive	% util	Average hold time	Name
35	82	29	0	15	30709590	[52B04FF8] DataStore@00F62BI
34	64	22	0	11	29551863	[00DC5C68] DataStore@00F62E
0	2	0	0	0	17784	[00DC634C] sun/misc/GC$Late
0	4	0	0	0	77675	[52B04818] java/lang/ref/Refer
0	7	0	0	0	9814	[52B04BB4] java/util/Timer$Tir
0	1	0	0	0	1947	[52B04FA4] DataAnalysis@00F6
0	73	0	0	0	226533	[52B05340] [I@00F6C2B8 (Obje

IDDE

GUI based, cross platform, dump debugger

Provides ability to:

- Analyse crash dumps for root cause
- View field and values store in objects
- Understand the state of your application

Visualization provided via Eclipse Client UI

- Available from Eclipse Marketplace
- Available from IBM Support Assistant

Supports analysis of dumps created on:

- AIX, Linux, Windows or z/OS

Provides local or remote analysis using server/client

Memory Analyzer

Application Memory Analysis Capability

- Analyses dump files

Provides ability to:

- Identify application memory leaks
- Optimize memory usage

Visualization provided via Eclipse Client UI

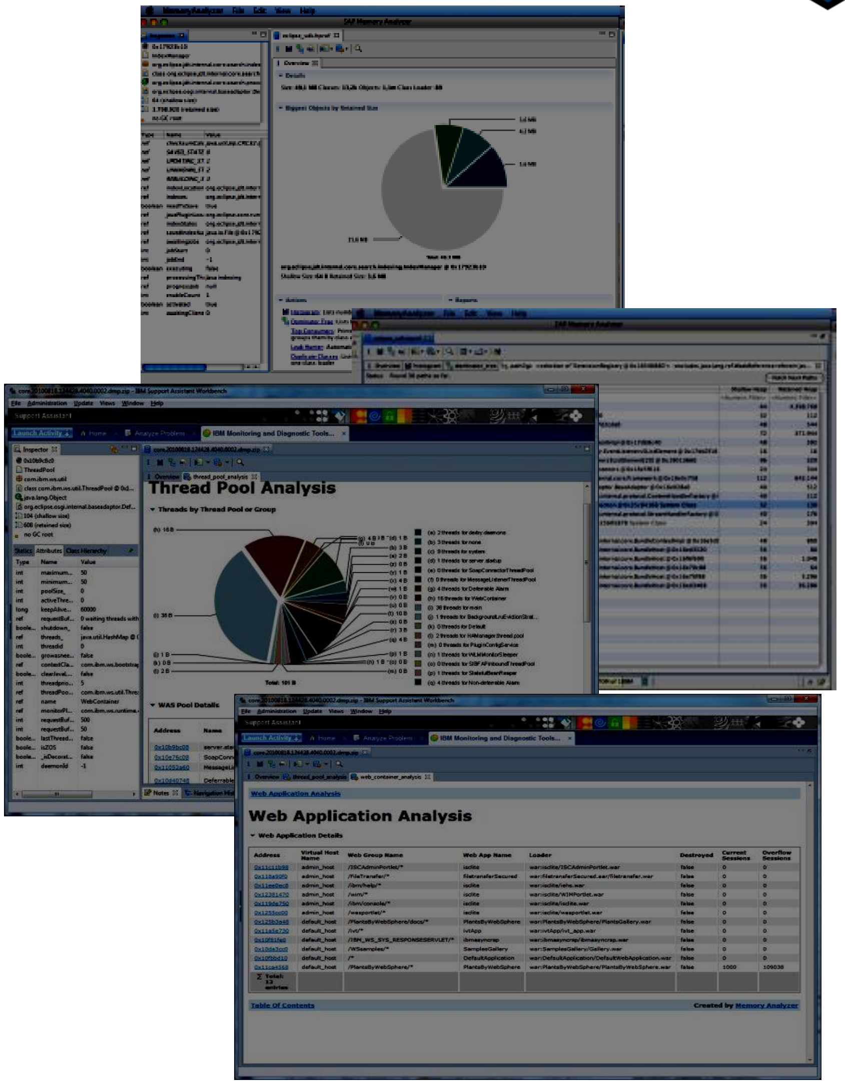
- Available from Eclipse Marketplace
- Available from IBM Support Assistant

Vizualization provided by Web UI

- Available in IBM Support Assistant

Headless/batch mode

- Run automated analysis



Thread Pool Analysis

Threads by Thread Pool or Group

(h) 16 B
(i) 38 B
(j) 1 B
(k) 0 B
(l) 2 B
Total: 101 B
(a) 2 threads for derby.daemons
(b) 3 threads for none
(c) 9 threads for system
(d) 1 threads for server.startup
(e) 0 threads for SoapConnectorThreadPool
(f) 0 threads for MessageListenerThreadPool
(g) 4 threads for Deferrable Alarm
(h) 16 threads for WebContainer
(i) 38 threads for main
(j) 1 threads for BackgroundLruEvictionStrat...
(k) 0 threads for Default
(l) 2 threads for HAManager.thread.pool
(m) 0 threads for PluginConfigService
(n) 1 threads for WLMMonitorSleeper
(o) 0 threads for SIBFAPInboundThreadPool
(p) 1 threads for StatefulBeanReaper
(q) 4 threads for Non-deferrable Alarm

WAS Pool Details

Web Application Analysis

Web Application Details

Address	Virtual Host Name	Web Group Name	Web App Name	Loader	Destroyed	Current Sessions	Overflow Sessions
0x11c11b98	admin_host	/ISCAdminPortlet/*	isclite	war:isclite/ISCAdminPortlet.war	false	0	0
0x118a90f0	admin_host	/FileTransfer/*	filetransferSecured	war:filetransferSecured.ear/filetransfer.war	false	0	0
0x11ee0ec8	admin_host	/ibm/help/*	isclite	war:isclite/iehs.war	false	0	0
0x12381470	admin_host	/wim/*	isclite	war:isclite/WIMPortlet.war	false	0	0
0x119de750	admin_host	/ibm/console/*	isclite	war:isclite/isclite.war	false	0	0
0x1255cc00	admin_host	/wasportlet/*	isclite	war:isclite/wasportlet.war	false	0	0
0x125b3a48	default_host	/PlantsByWebSphere/docs/*	PlantsByWebSphere	war:PlantsByWebSphere/PlantsGallery.war	false	0	0
0x11a5e730	default_host	/ivt/*	ivtApp	war:ivtApp/ivt_app.war	false	0	0
0x10f81fe0	default_host	/IBM_WS_SYS_RESPONSESERVLET/*	ibmasyncrsp	war:ibmasyncrsp/ibmasyncrsp.war	false	0	0
0x10da3cc0	default_host	/WSsamples/*	SamplesGallery	war:SamplesGallery/Gallery.war	false	0	0
0x10fbbd10	default_host	/*	DefaultApplication	war:DefaultApplication/DefaultWebApplication.war	false	0	0
0x11ca4568	default_host	/PlantsByWebSphere/*	PlantsByWebSphere	war:PlantsByWebSphere/PlantsByWebSphere.war	false	1000	109038
Σ Total: 12 entries							

Table Of Contents

Created by Memory Analyzer

Other Languages on the JVM

Many Languages exist on the JVM

Incremental improvements to the JVM support this:

– Java 7 and JSR292 (invokeDynamic)

- new bytecode that executes a given method directly
- provides the ability at runtime to rewire what method that is.
- decouples method lookup and method dispatch

– Enables JIT optimizations to handle dynamic types

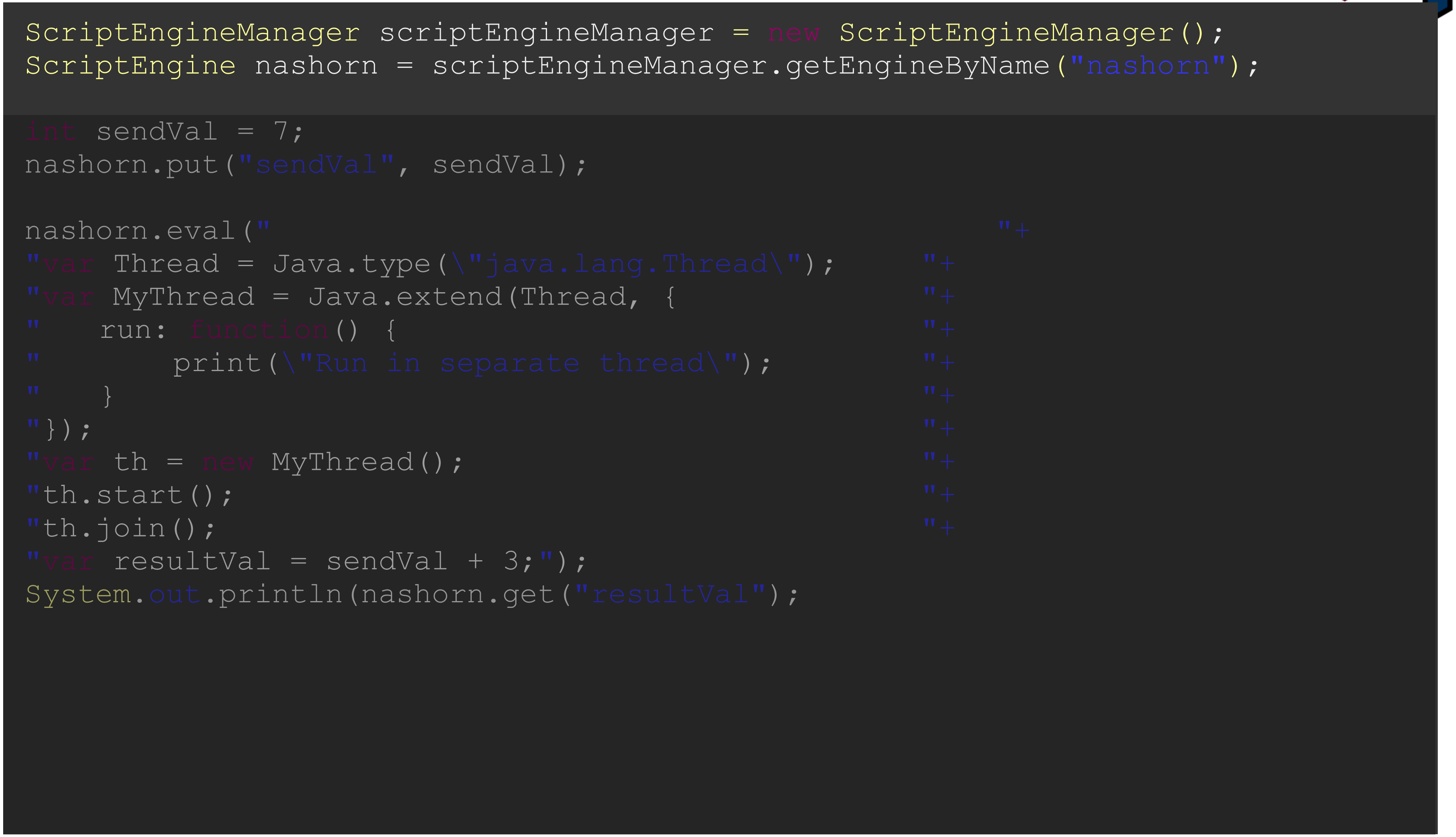
Case Study: JavaScript

- Rhino was bundled in JDK6
  - Based on the Mozilla Rhino engine
  - Allowed JavaScript to be embedded in Java
  - Allowed JavaScript to call out to Java
  - Provided `jrunscript` command line utility

- Nashorn replaces JavaScript in JDK8
  - Supports full ECMAScript 5.1 specification
  - Exploits invokedynamic for 2x to 10x better performance than Rhino
  - Provides `jjs` command line utility

- Avatar.js builds on Nashorn to provide Node.js support
- Allow Node.js applications to run on the JVM

Case Study: JavaScript

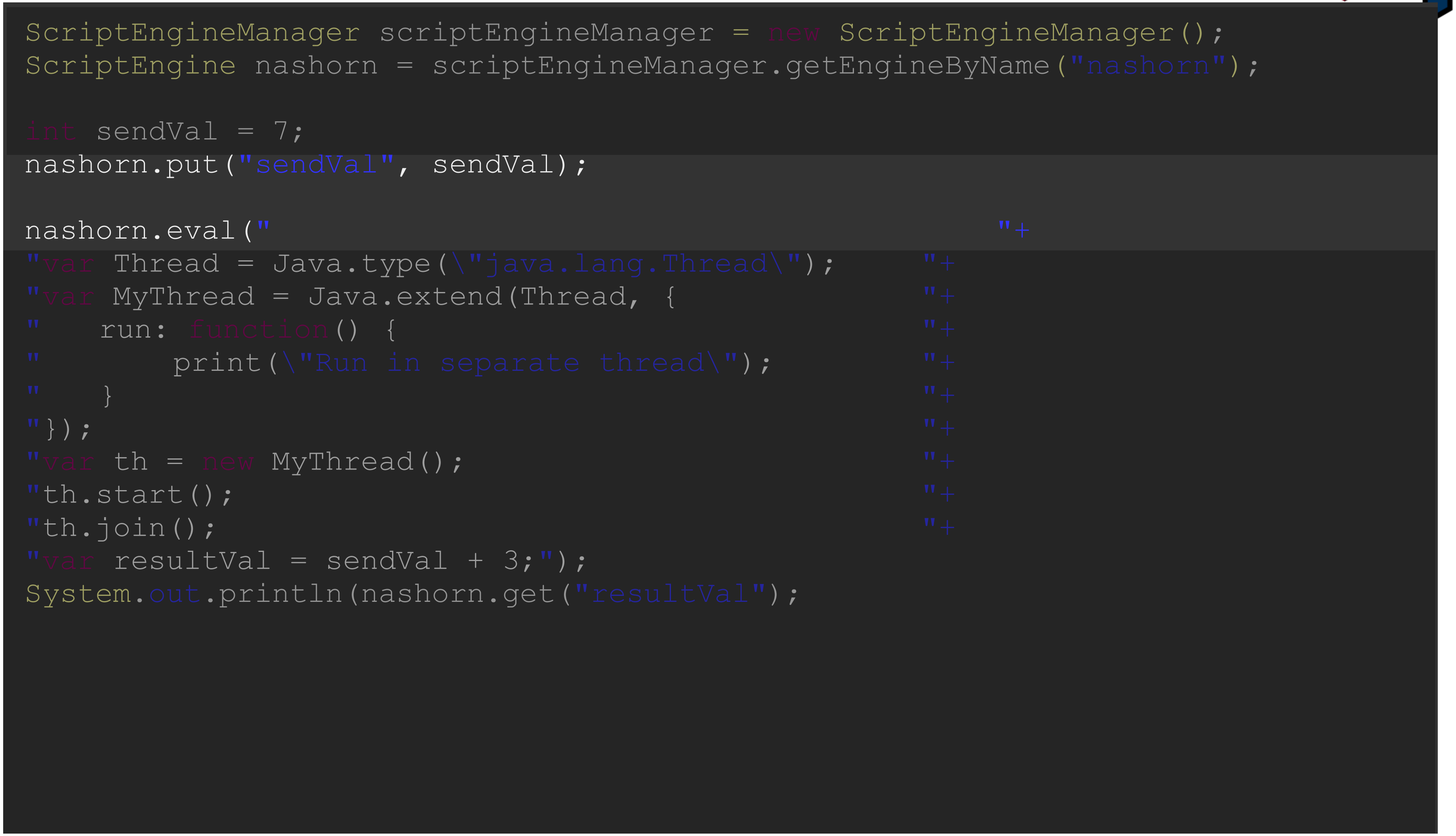


```
ScriptEngineManager scriptEngineManager = new ScriptEngineManager();
ScriptEngine nashorn = scriptEngineManager.getEngineByName("nashorn");

int sendVal = 7;
nashorn.put("sendVal", sendVal);

nashorn.eval("                                         "+
"var Thread = Java.type(\"java.lang.Thread\");     "+
"var MyThread = Java.extend(Thread, {              "+
"   run: function() {                              "+
"       print(\"Run in separate thread\");         "+
"    }                                             "+
"});                                               "+
"var th = new MyThread();                          "+
"th.start();                                       "+
"th.join();                                        "+
"var resultVal = sendVal + 3;");
System.out.println(nashorn.get("resultVal"));
```

Case Study: JavaScript

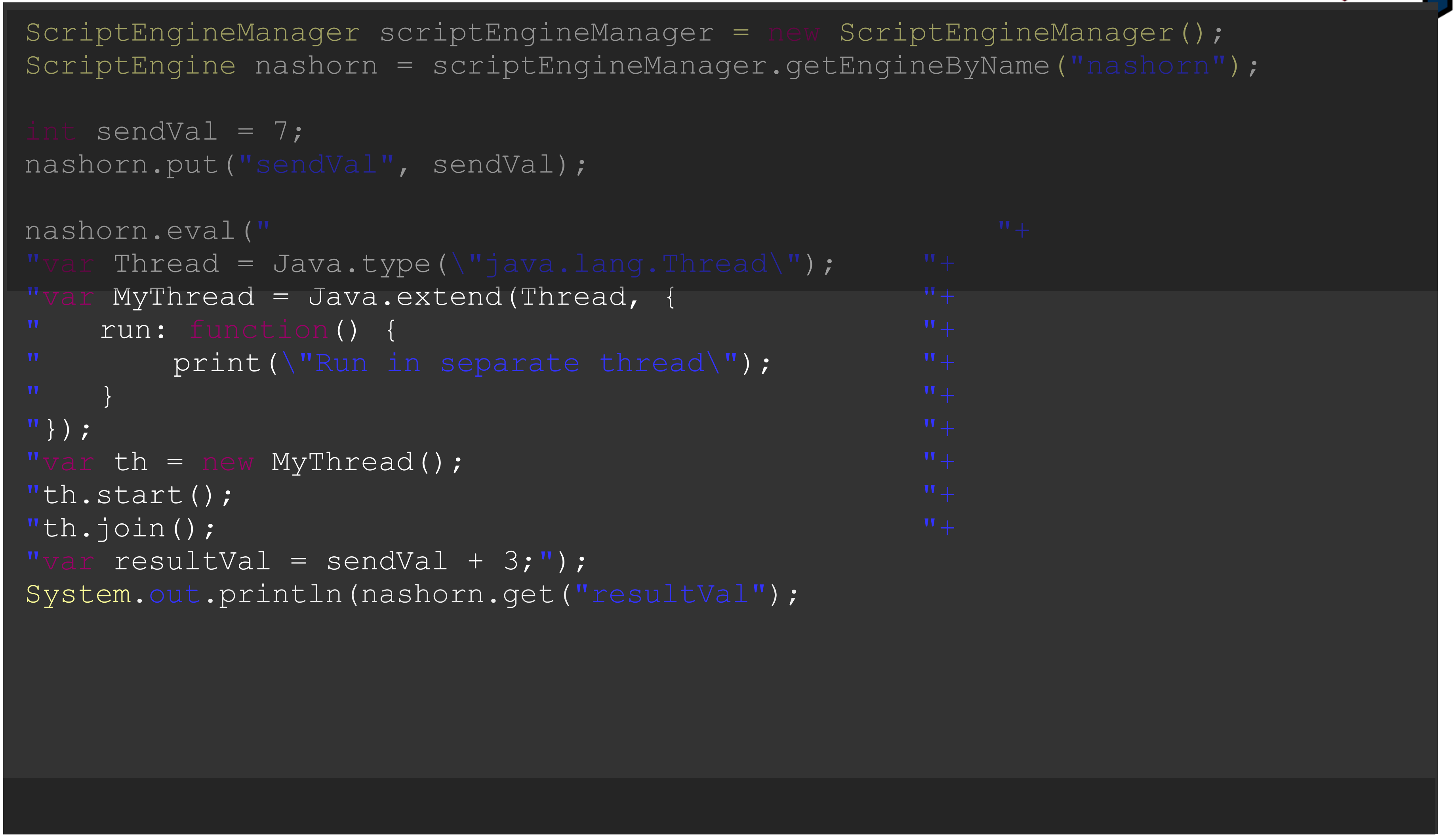


```
ScriptEngineManager scriptEngineManager = new ScriptEngineManager();
ScriptEngine nashorn = scriptEngineManager.getEngineByName("nashorn");

int sendVal = 7;
nashorn.put("sendVal", sendVal);

nashorn.eval("                                              "+
"var Thread = Java.type(\"java.lang.Thread\");          "+
"var MyThread = Java.extend(Thread, {                   "+
"    run: function() {                                  "+
"        print(\"Run in separate thread\");             "+
"    }                                                  "+
"});                                                    "+
"var th = new MyThread();                               "+
"th.start();                                            "+
"th.join();                                             "+
"var resultVal = sendVal + 3;");
System.out.println(nashorn.get("resultVal"));
```

Case Study: JavaScript

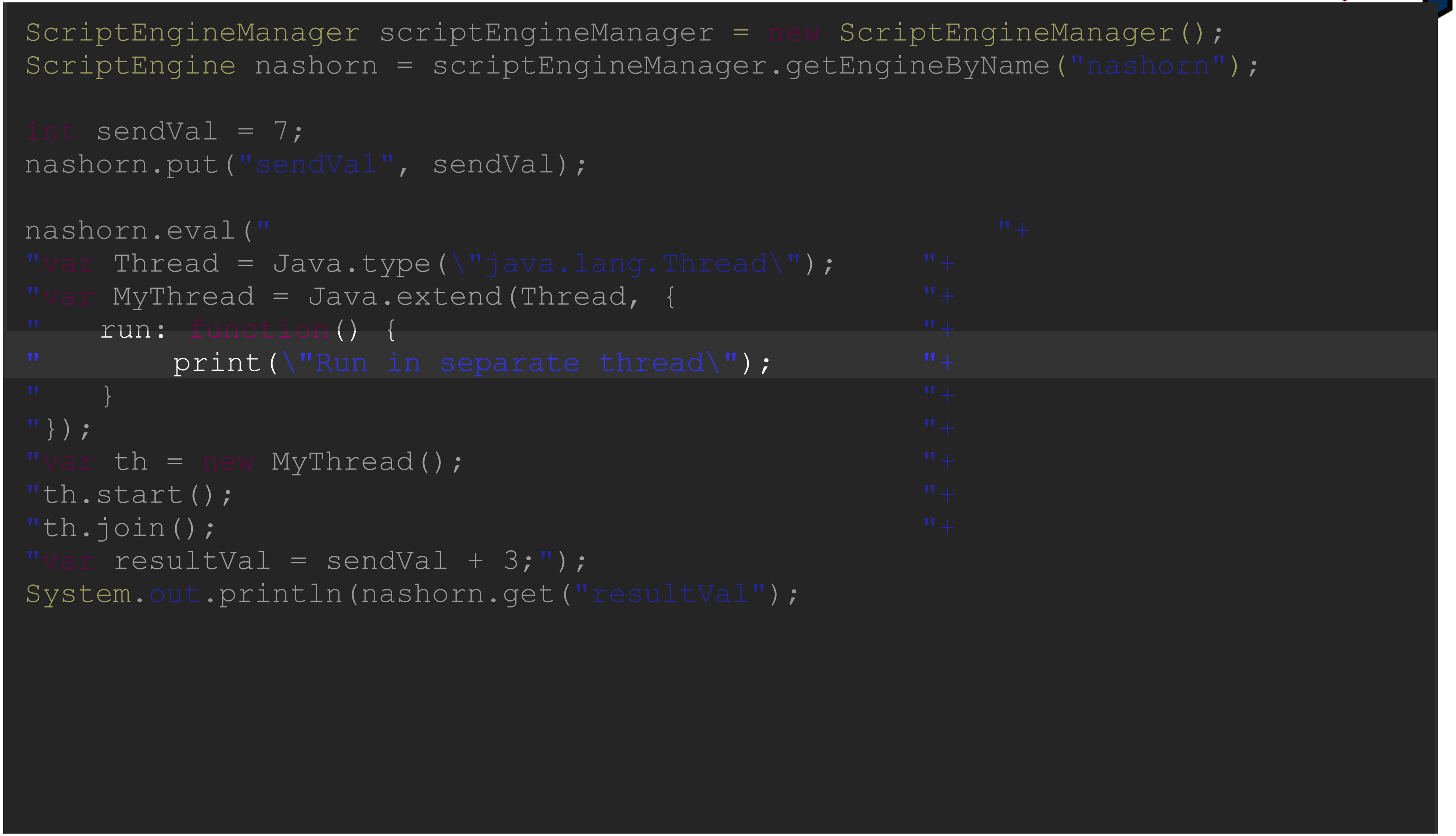


```
ScriptEngineManager scriptEngineManager = new ScriptEngineManager();
ScriptEngine nashorn = scriptEngineManager.getEngineByName("nashorn");

int sendVal = 7;
nashorn.put("sendVal", sendVal);

nashorn.eval("                                         "+
"var Thread = Java.type(\"java.lang.Thread\");     "+
"var MyThread = Java.extend(Thread, {              "+
"   run: function() {                              "+
"       print(\"Run in separate thread\");         "+
"   }                                              "+
"});                                               "+
"var th = new MyThread();                          "+
"th.start();                                       "+
"th.join();                                        "+
"var resultVal = sendVal + 3;");
System.out.println(nashorn.get("resultVal"));
```

Case Study: JavaScript



```
ScriptEngineManager scriptEngineManager = new ScriptEngineManager();
ScriptEngine nashorn = scriptEngineManager.getEngineByName("nashorn");

int sendVal = 7;
nashorn.put("sendVal", sendVal);

nashorn.eval("                                         "+
"var Thread = Java.type(\"java.lang.Thread\");     "+
"var MyThread = Java.extend(Thread, {              "+
"   run: function() {                              "+
"       print(\"Run in separate thread\");         "+
"   }                                              "+
"});                                               "+
"var th = new MyThread();                          "+
"th.start();                                       "+
"th.join();                                        "+
"var resultVal = sendVal + 3;");
System.out.println(nashorn.get("resultVal"));
```

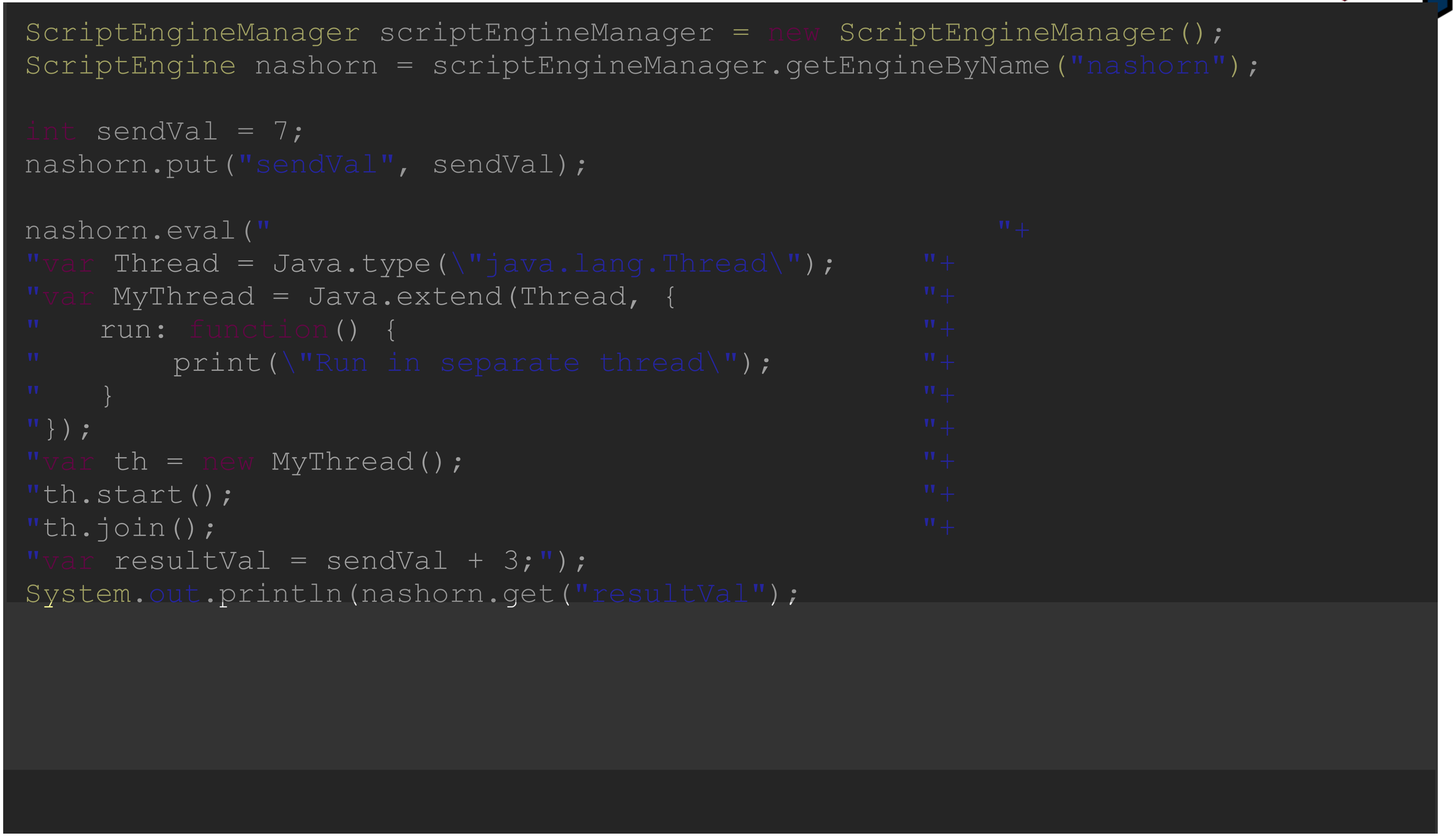
Case Study: JavaScript

```
ScriptEngineManager scriptEngineManager = new ScriptEngineManager();
ScriptEngine nashorn = scriptEngineManager.getEngineByName("nashorn");

int sendVal = 7;
nashorn.put("sendVal", sendVal);

nashorn.eval("                                         "+
"var Thread = Java.type(\"java.lang.Thread\");     "+
"var MyThread = Java.extend(Thread, {              "+
"   run: function() {                              "+
"       print(\"Run in separate thread\");         "+
"   }                                              "+
"});                                               "+
"var th = new MyThread();                          "+
"th.start();                                       "+
"th.join();                                        "+
"var resultVal = sendVal + 3;");
System.out.println(nashorn.get("resultVal"));
```

Case Study: JavaScript



```
ScriptEngineManager scriptEngineManager = new ScriptEngineManager();
ScriptEngine nashorn = scriptEngineManager.getEngineByName("nashorn");

int sendVal = 7;
nashorn.put("sendVal", sendVal);

nashorn.eval("                                         "+
"var Thread = Java.type(\"java.lang.Thread\");     "+
"var MyThread = Java.extend(Thread, {              "+
"   run: function() {                              "+
"       print(\"Run in separate thread\");         "+
"   }                                              "+
"});                                               "+
"var th = new MyThread();                          "+
"th.start();                                       "+
"th.join();                                        "+
"var resultVal = sendVal + 3;");
System.out.println(nashorn.get("resultVal");
```

Case Study: JavaScript

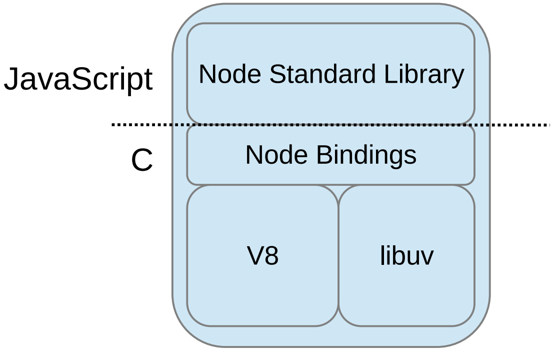
```
ScriptEngineManager scriptEngineManager = new ScriptEngineManager();
ScriptEngine nashorn = scriptEngineManager.getEngineByName("nashorn");

int sendVal = 7;
nashorn.put("sendVal", sendVal);

nashorn.eval("                                         "+
"var Thread = Java.type(\"java.lang.Thread\");     "+
"var MyThread = Java.extend(Thread, {              "+
"   run: function() {                              "+
"       print(\"Run in separate thread\");         "+
"   }                                              "+
"});                                               "+
"var th = new MyThread();                          "+
"th.start();                                       "+
"th.join();                                        "+
"var resultVal = sendVal + 3;");
System.out.println(nashorn.get("resultVal"));
```

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

Case Study: JavaScript and Avatar.js

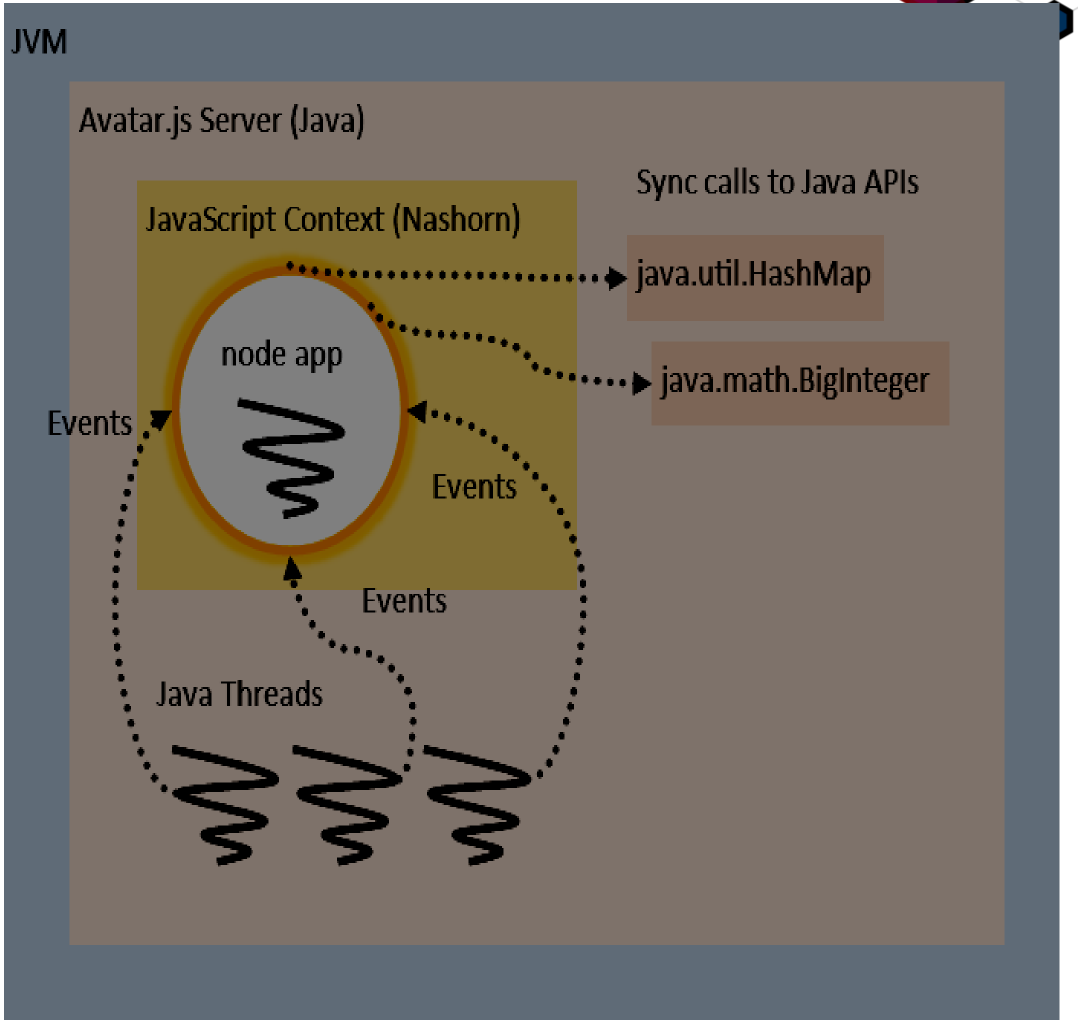
- Support for Node.js on Nashorn

- Binary builds available from Maven
  - Avatar-js.jar
  - Avatar-js library (64bit)

- Number of common NPM modules are supported

- NPM required to dependencies

- Issues for native NPMs
  - No native V8 APIs



From https://avatar-js.java.net/

Case Study: Avatar.js vs Node.js

- Running Octane r33
  - The “SpecJVM98” of the JavaScript world
  - Actually a JavaScript benchmark rather than a Node.js benchmark
  - So more comparing Nashorn to V8

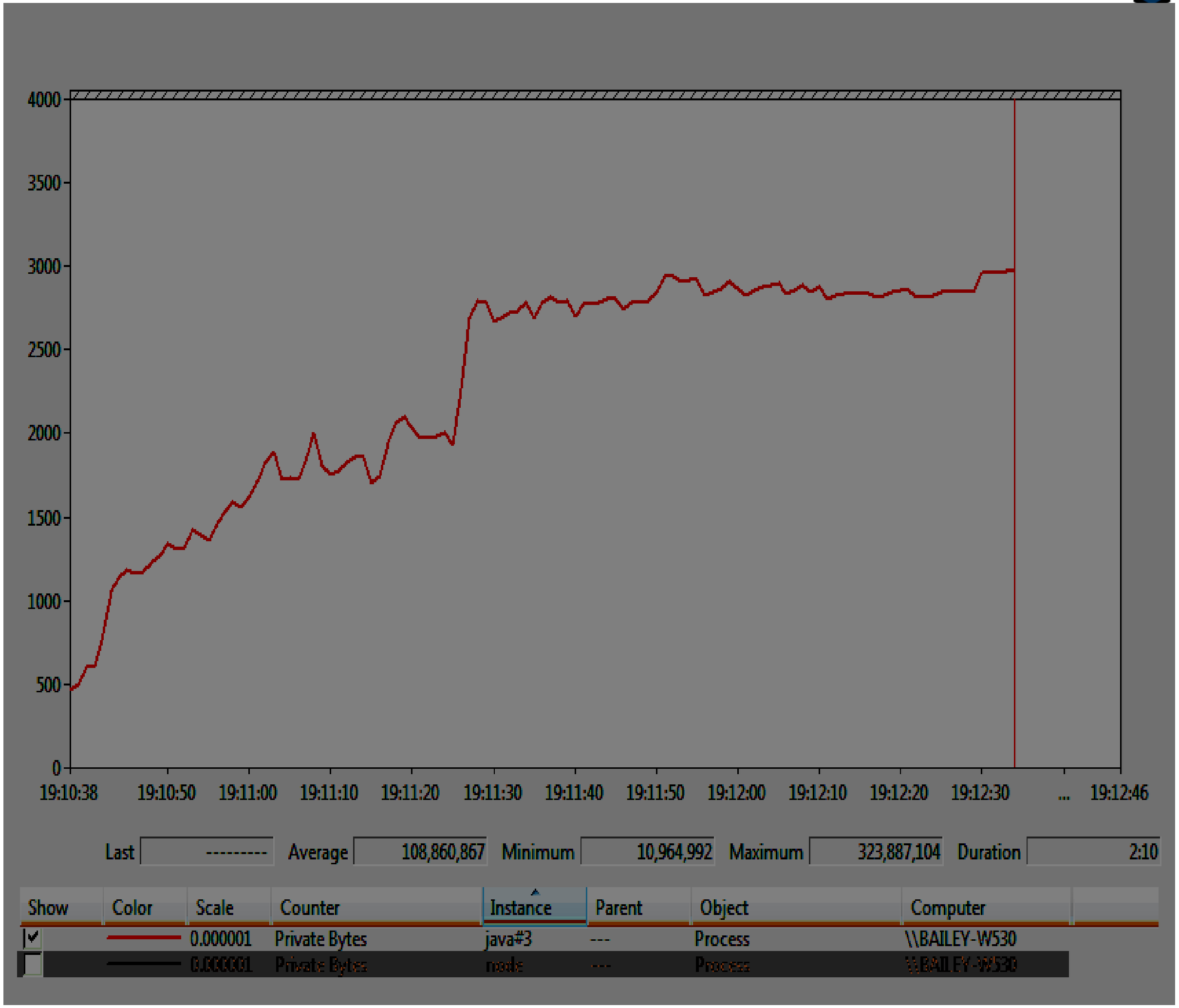
- Run on a 8 CPU Windows using:
  - Node.js v0.10.31
  - HotSpot 8u20

- Settings are “out of the box” with no attempt to tune

Case Study: Octane Benchmark

Avatar.js

- Duration: 1m 56s
- Peak memory: 2830MB



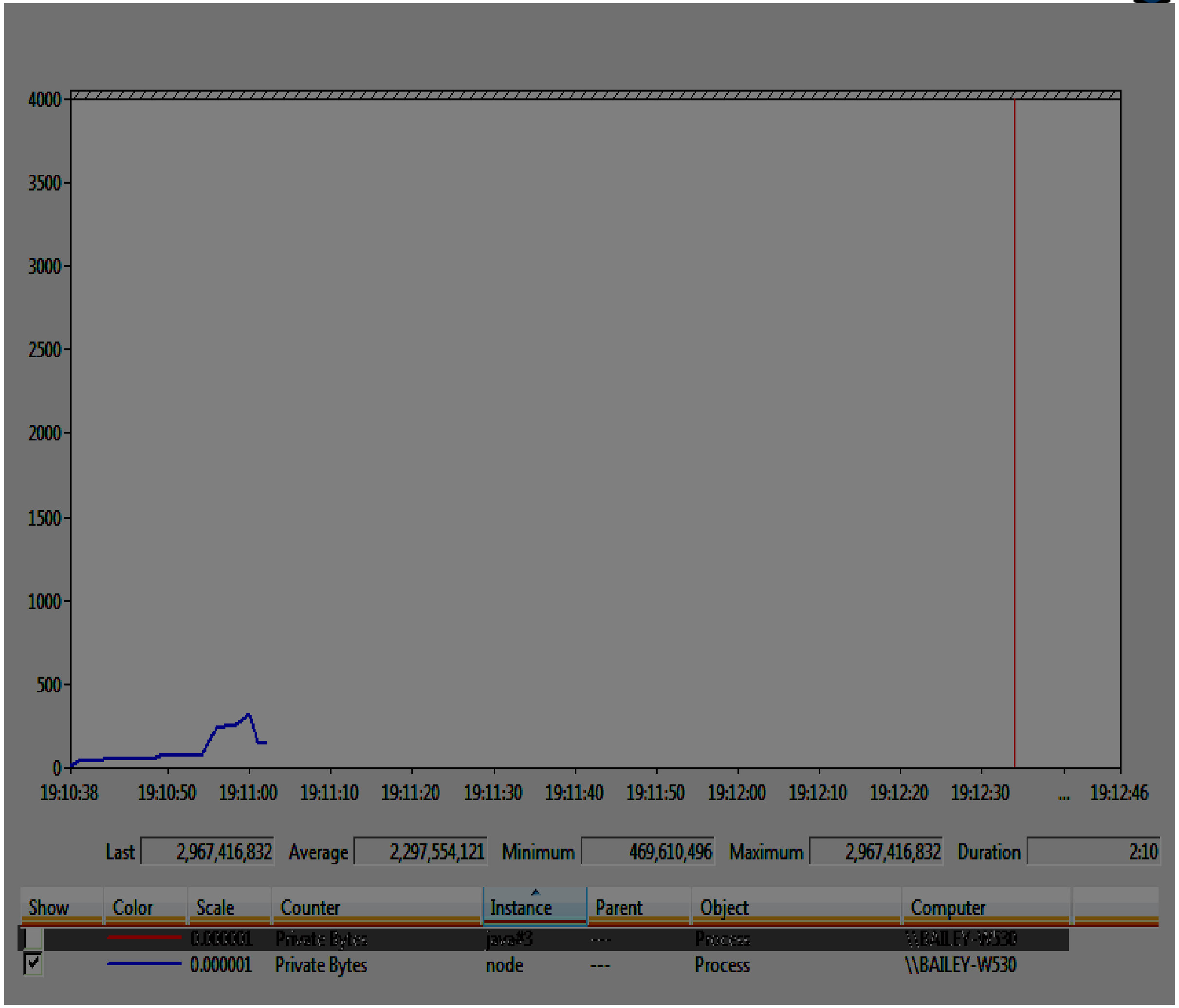
Last	Average	Minimum	Maximum	Duration
---------	108,860,867	10,964,992	323,887,104	2:10

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

Case Study: Octane Benchmark

Node.js

- Duration: 24s
- Peak memory: 268MB



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

Case Study: Octane Benchmark

Avatar.js

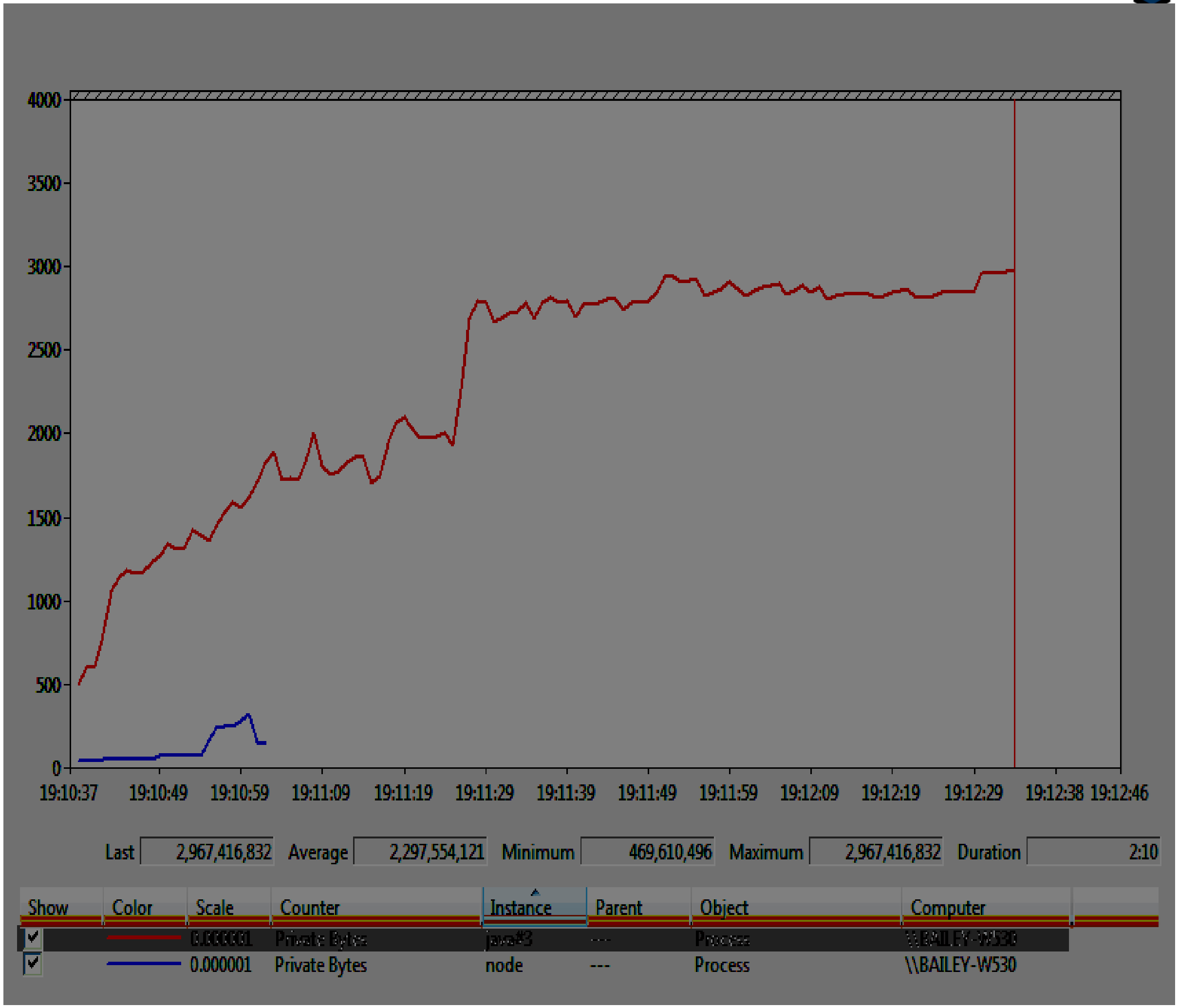
- Duration: 1m 56s
- Peak memory: 2830MB

Node.js

- Duration: 24s
- Peak memory: 268MB

Node.js is 4.8x faster

Avatar.js is >10x bigger



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

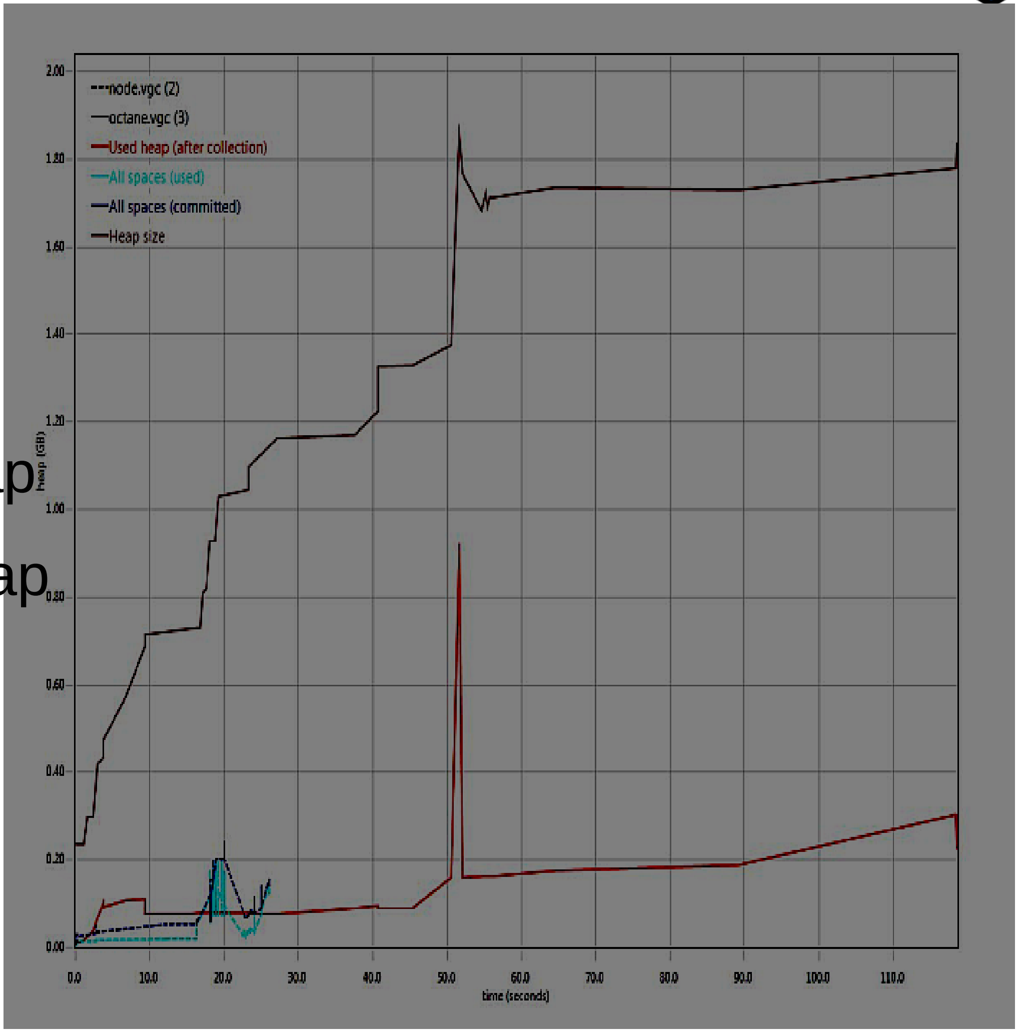
Show	Color	Scale	Counter	Instance	Parent	Object	Computer
☑	🟥	0.000001	Private Bytes	java#3	---	Process	\\BAILEY-W530
☑	🟦	0.000001	Private Bytes	node	---	Process	\\BAILEY-W530

Case Study: Garbage Collection

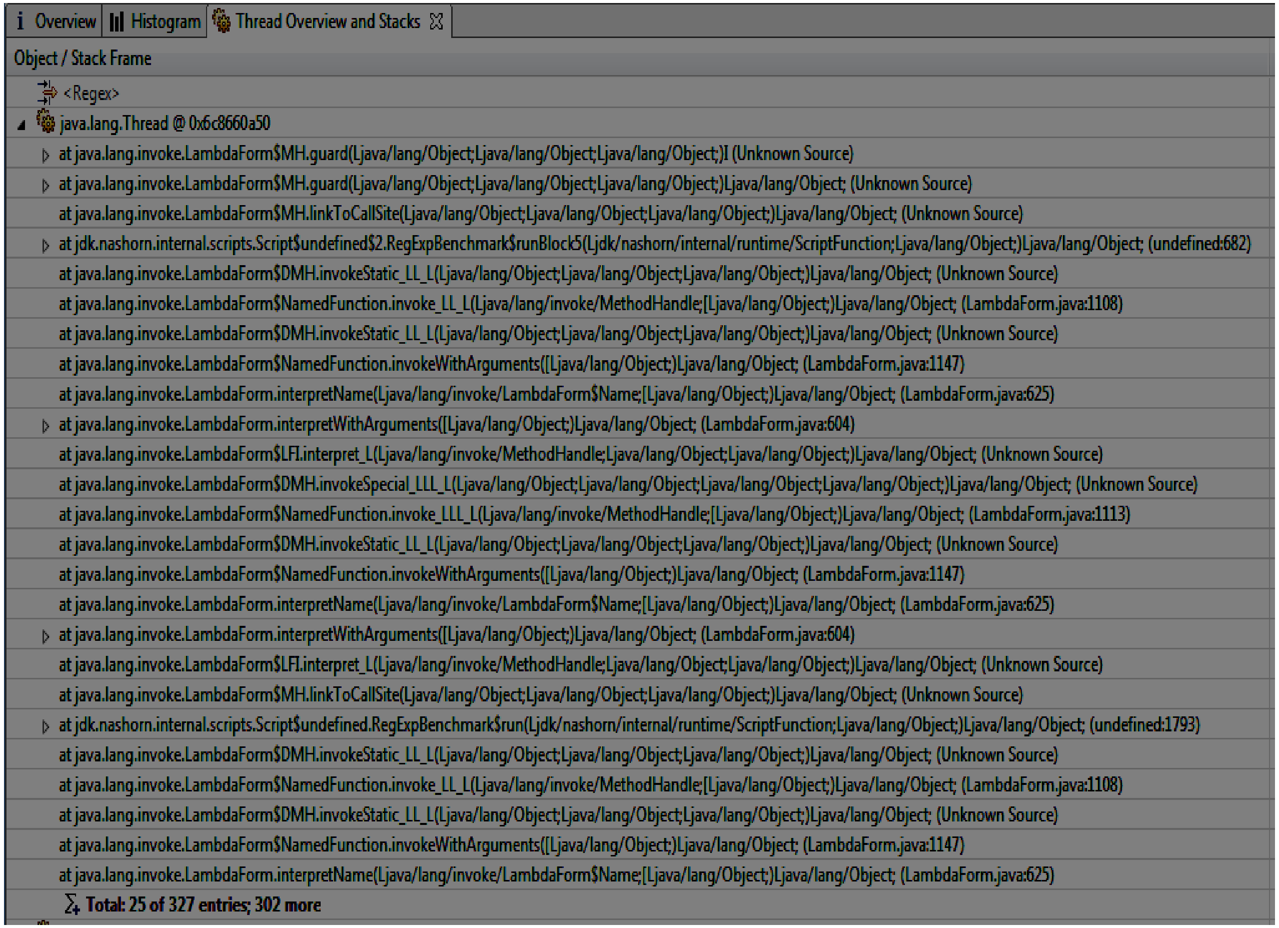
- Avatar.js peak is:
  - 920MB / 1.85GB

- Node.js peak is:
  - 200MB / 220MB

- Extra 700+MB of Object Heap
- Extra 1GB of non-Object Heap



Case Study: Avatar.js Stack Traces

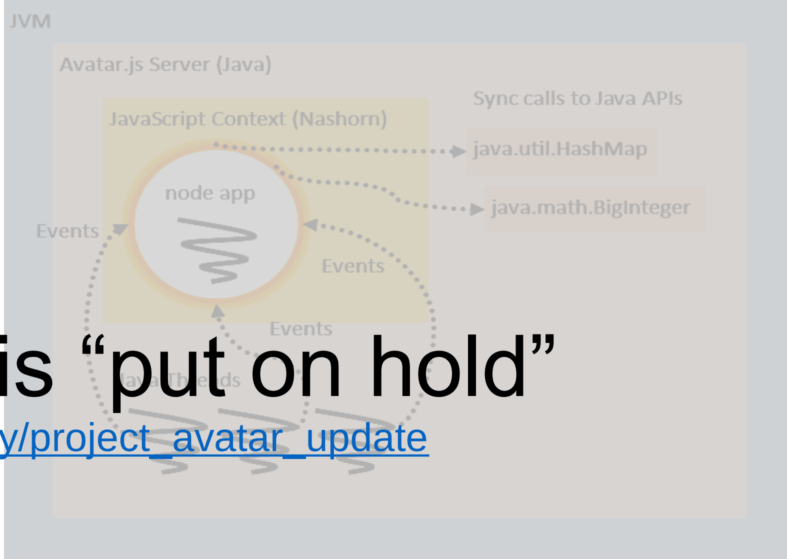


```
java.lang.Thread @ 0x6c8660a50
  at java.lang.invoke.LambdaForm$MH.guard(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)I (Unknown Source)
  at java.lang.invoke.LambdaForm$MH.guard(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (Unknown Source)
  at java.lang.invoke.LambdaForm$MH.linkToCallSite(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (Unknown Source)
  at jdk.nashorn.internal.scripts.Script$undefined$2.RegExpBenchmark$runBlock5(Ljdk/nashorn/internal/runtime/ScriptFunction;Ljava/lang/Object;)Ljava/lang/Object; (undefined:682)
  at java.lang.invoke.LambdaForm$DMH.invokeStatic_LL_L(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (Unknown Source)
  at java.lang.invoke.LambdaForm$NamedFunction.invoke_LL_L(Ljava/lang/invoke/MethodHandle;[Ljava/lang/Object;)Ljava/lang/Object; (LambdaForm.java:1108)
  at java.lang.invoke.LambdaForm$DMH.invokeStatic_LL_L(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (Unknown Source)
  at java.lang.invoke.LambdaForm$NamedFunction.invokeWithArguments([Ljava/lang/Object;)Ljava/lang/Object; (LambdaForm.java:1147)
  at java.lang.invoke.LambdaForm.interpretName(Ljava/lang/invoke/LambdaForm$Name;[Ljava/lang/Object;)Ljava/lang/Object; (LambdaForm.java:625)
  at java.lang.invoke.LambdaForm.interpretWithArguments([Ljava/lang/Object;)Ljava/lang/Object; (LambdaForm.java:604)
  at java.lang.invoke.LambdaForm$LFI.interpret_L(Ljava/lang/invoke/MethodHandle;Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (Unknown Source)
  at java.lang.invoke.LambdaForm$DMH.invokeSpecial_LLL_L(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (Unknown Source)
  at java.lang.invoke.LambdaForm$NamedFunction.invoke_LLL_L(Ljava/lang/invoke/MethodHandle;[Ljava/lang/Object;)Ljava/lang/Object; (LambdaForm.java:1113)
  at java.lang.invoke.LambdaForm$DMH.invokeStatic_LL_L(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (Unknown Source)
  at java.lang.invoke.LambdaForm$NamedFunction.invokeWithArguments([Ljava/lang/Object;)Ljava/lang/Object; (LambdaForm.java:1147)
  at java.lang.invoke.LambdaForm.interpretName(Ljava/lang/invoke/LambdaForm$Name;[Ljava/lang/Object;)Ljava/lang/Object; (LambdaForm.java:625)
  at java.lang.invoke.LambdaForm.interpretWithArguments([Ljava/lang/Object;)Ljava/lang/Object; (LambdaForm.java:604)
  at java.lang.invoke.LambdaForm$LFI.interpret_L(Ljava/lang/invoke/MethodHandle;Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (Unknown Source)
  at java.lang.invoke.LambdaForm$MH.linkToCallSite(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (Unknown Source)
  at jdk.nashorn.internal.scripts.Script$undefined.RegExpBenchmark$run(Ljdk/nashorn/internal/runtime/ScriptFunction;Ljava/lang/Object;)Ljava/lang/Object; (undefined:1793)
  at java.lang.invoke.LambdaForm$DMH.invokeStatic_LL_L(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (Unknown Source)
  at java.lang.invoke.LambdaForm$NamedFunction.invoke_LL_L(Ljava/lang/invoke/MethodHandle;[Ljava/lang/Object;)Ljava/lang/Object; (LambdaForm.java:1108)
  at java.lang.invoke.LambdaForm$DMH.invokeStatic_LL_L(Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object; (Unknown Source)
  at java.lang.invoke.LambdaForm$NamedFunction.invokeWithArguments([Ljava/lang/Object;)Ljava/lang/Object; (LambdaForm.java:1147)
  at java.lang.invoke.LambdaForm.interpretName(Ljava/lang/invoke/LambdaForm$Name;[Ljava/lang/Object;)Ljava/lang/Object; (LambdaForm.java:625)
  Total: 25 of 327 entries; 302 more
```

- Interpretation of JavaScript at the Java layer means stacks are large and anonymous

Nashorn and Avatar.js

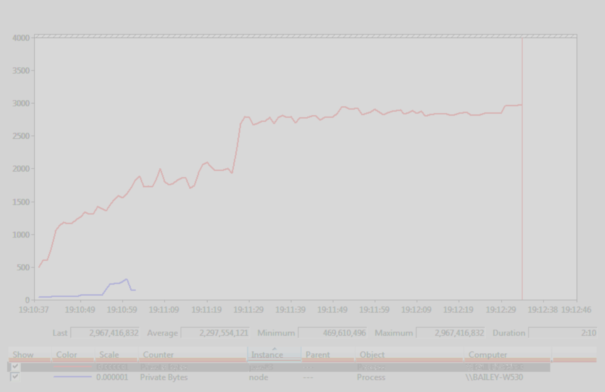
- Nashorn JavaScript engine delivered in JDK8
  - Utilizes new JVM level features for performance
- Avatar.js provides Node.js support on Nashorn



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

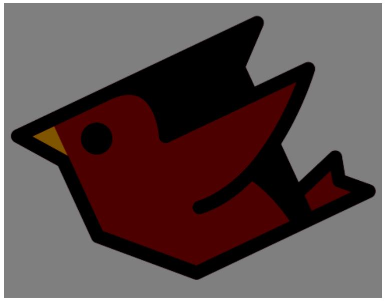


Last 2,967,416,832 Average 2,297,554,121 Minimum 469,610,496 Maximum 2,967,416,832 Duration 2:10

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

* Using Java 8 pre-u40

Case Study: JRuby

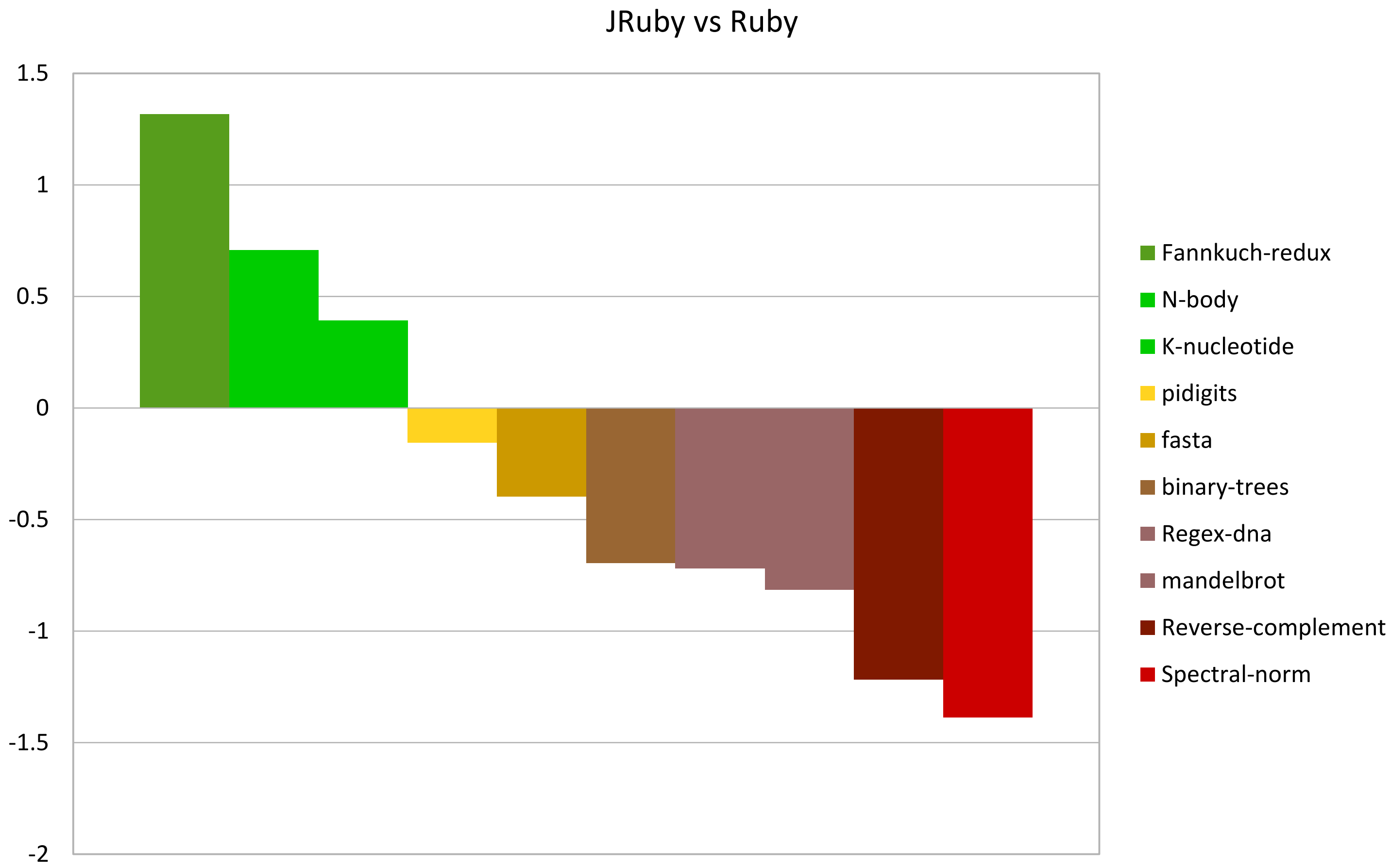


- Ruby programming language on the JVM
  - High performance
  - Real threading
  - Vast array of libraries
  - Platform independent

- Ruby 2.2 API compatible

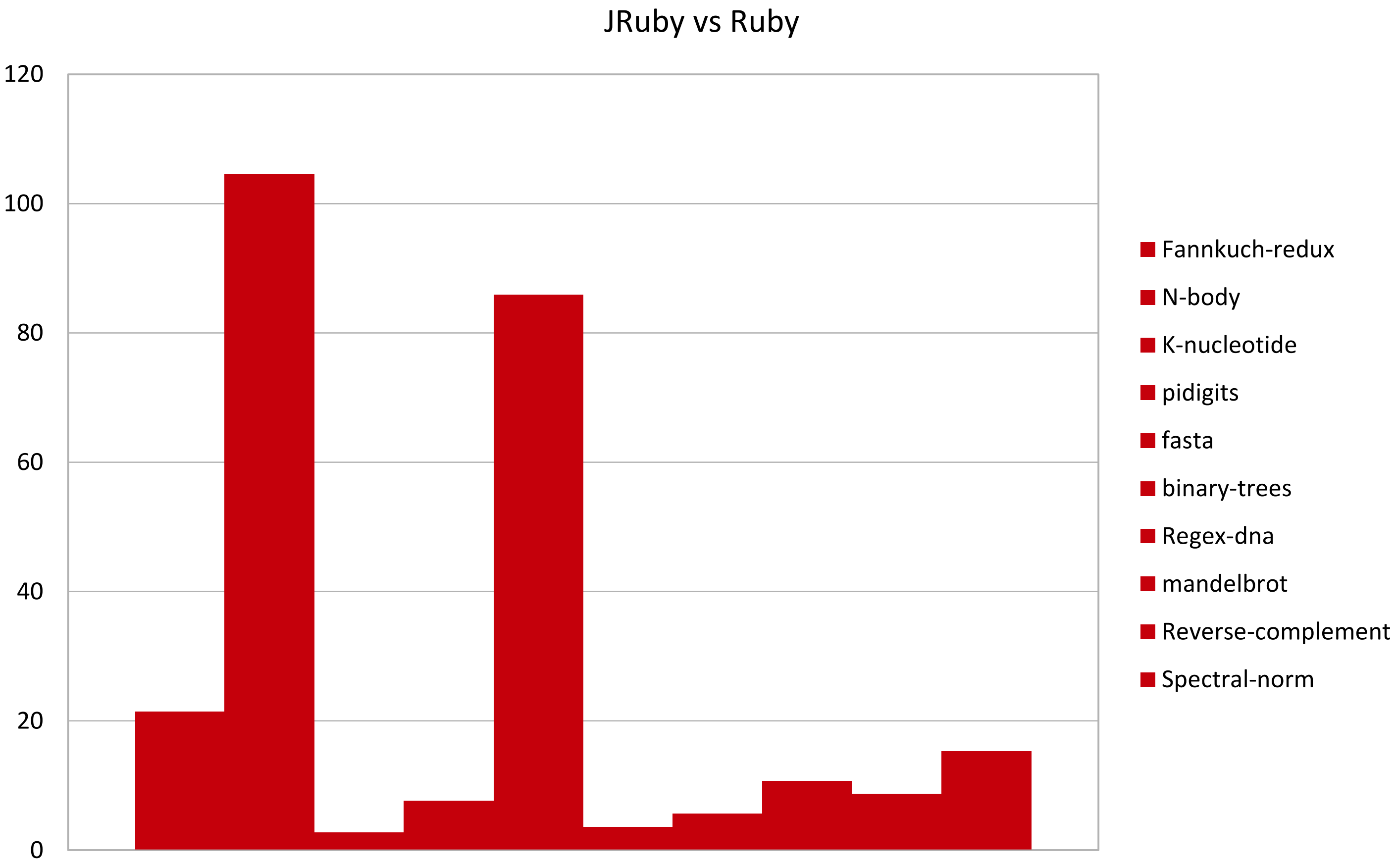
- Homepage: http://www.jruby.org/
- Download: http://www.jruby.org/download

Case Study: JRuby



Source: http://benchmarksgame.alioth.debian.org/

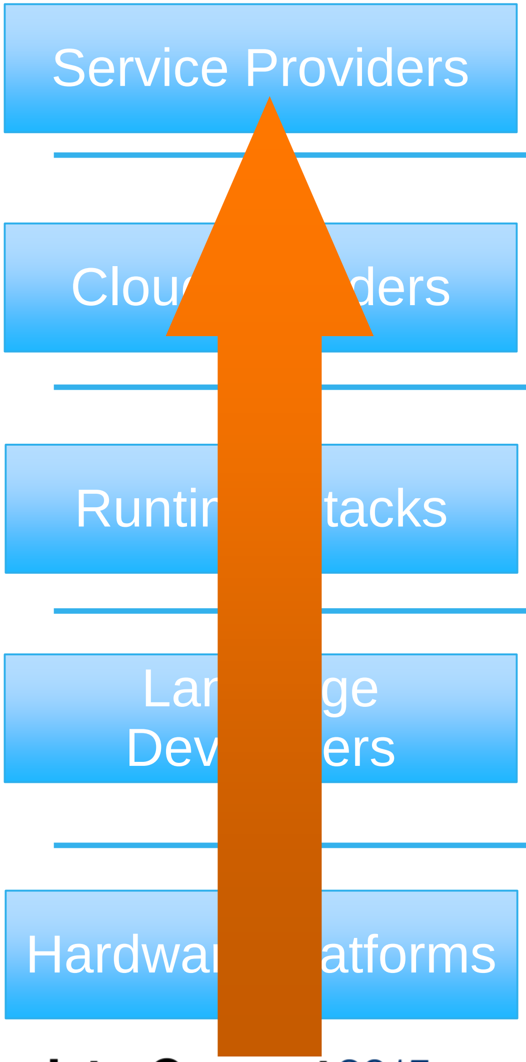
Case Study: JRuby



Source: http://benchmarksgame.alioth.debian.org/

IBM and Other Languages

The Ecosystem is Polyglot



Leverage the platform capabilities to accelerate computation and communication regardless of language runtime

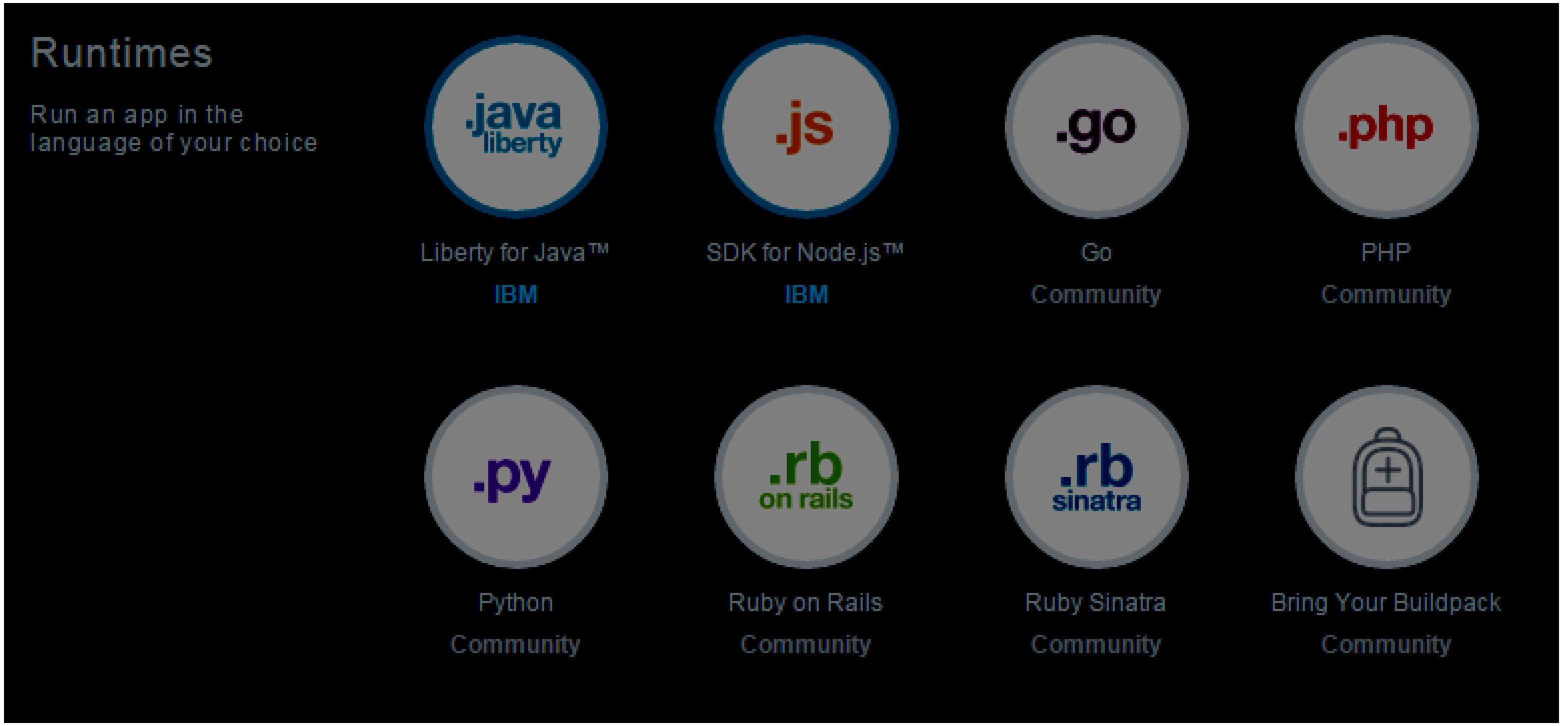
Need a premium deployment platform where the language runtime showcases their density, performance, scaling and reliability

Want hardware platform benefits to showcase their stack regardless of the language runtime engine

Want high performance infrastructure but lack expertise. Lack a common point of investment for runtime acceleration on all platforms

Want to ensure quality implementations of language runtimes on their HW platforms and drive innovation to exploit hardware

Bluemix is Polyglot



Runtimes

Run an app in the language of your choice

Liberty for Java™
IBM

SDK for Node.js™
IBM

Go
Community

PHP
Community

Python
Community

Ruby on Rails
Community

Ruby Sinatra
Community

Bring Your Buildpack
Community

IBM is working in Open Technologies

Cloud Foundry Foundation

Founding member & Platinum Sponsor

Docker Community

Working with Docker on defining an open engagement model

OpenStack Foundation

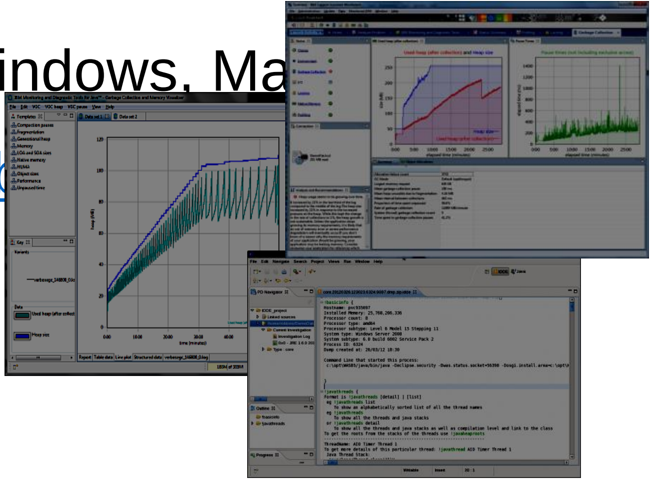
Founding member & Platinum Sponsor

Node.js Foundation

Founding member & Platinum Sponsor

IBM SDK for Node.js

- Node.js Foundation
  - Founding Member and Platinum Sponsor
  - 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, Ma
  - http://www.ibm.com/developerworks/web
- IBM Monitoring and Diagnostics Tools
  - Live monitoring: Health Center
  - GC log analysis: GCMV



```
!basicinfo {
Hostname: pvc935097
Installed Memory: 25,768,206,336
Processor count: 8
Processor type: amd64
Processor subtype: Level 6 Model 15 Stepping 11
System type: Windows Server 2008
System subtype: 6.0 build 6002 Service Pack 2
Process ID: 6324
Dump created at: 26/03/12 18:30

Command Line that started this process:
 c:\opt\WAS85/java/bin/java -Declipse.security -Dwas.status.socket=56398 -Dosgi.install.area=c:\opt\

}

!javathreads {
Format is !javathreads [detail] | [list]
 eg !javathreads list
    To show an alphabetically sorted list of all the thread names
 eg !javathreads
    To show all the threads and java stacks
 or !javathreads detail
    To show all the threads and java stacks as well as compilation level and link to the class
To get the roots from the stacks of the threads use !javaheaproots
==========================================================
ThreadName: AIO Timer Thread 1
To get more details of this particular thread: !javathread AIO Timer Thread 1
 Java Thread Stack:
```

Questions?

Today's Sessions

Monday March 23rd		
11:00 – 12:00	Introduction to the IBM Monitoring and Diagnostic Tools for Java and JavaScript	
	Wilberforce	*Ian Partridge*
13:30 – 14:30	Virtualization Aware JVM	
	Wilberforce	*Tim Ellison*
14:40 – 15:40	Java vs JavaScript for Enterprise Web Applications	
	Wilberforce	*Chris Bailey*
16:00 – 17:00	What's new in IBM Java 8 SE?	
	Wilberforce	*Tim Ellison*

ibm.biz/wug15feedback

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.

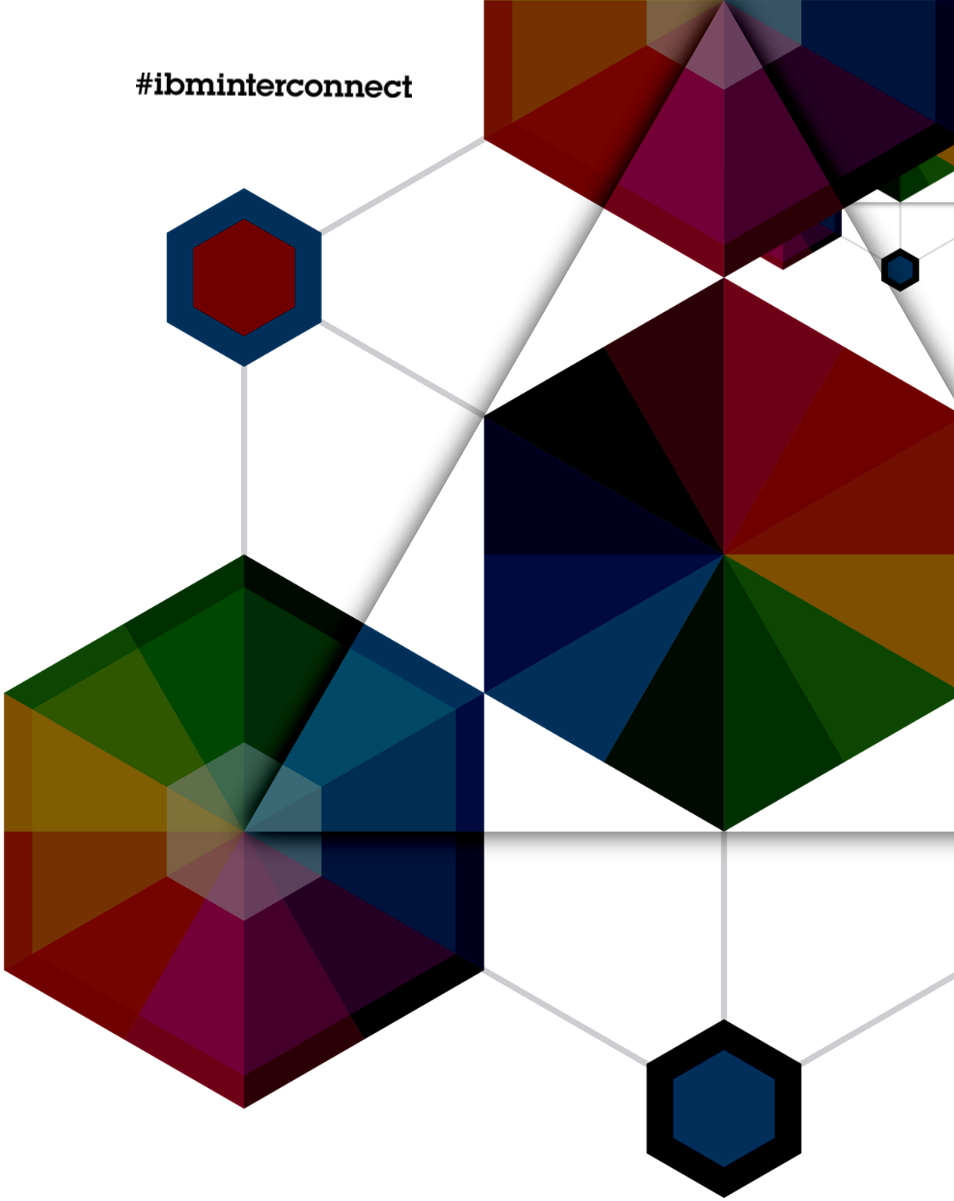
Notices and Disclaimers (con’t)

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.

#ibminterconnect



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.

IBM®

InterConnect2015

The Premier Cloud & Mobile Conference