Java 2 Days

IS THE FUTURE OF JAVA CLOUDY?

Oct 2017
1992: A momentous year

I joined IBM and took my tie off

Hired because I had deep knowledge of AS/400

Do you remember when skillsets were platform oriented?
A few years later I’m sitting in a Sun office in Cupertino talking about garbage collection design

Java is already the platform
For most of 20 years the answer is Java
Over the last few years Java’s dominance has faltered
Partly because squeezing a JVM into small devices is hard (and we cut corners)

Partly because Java is seen as ‘old’
Mostly because Java hasn’t evolved fast enough.

It took 5 years to get Lambdas.

It took 10 years to get Modules
The new requirements of Cloud have arrived even faster
Is Java going the way of the dodo?

Is James Goslings vision of Java on every toaster... Toast?
For Java to compete and remain relevant it has to be the platform of choice (again)
In three arenas:

Cloud
Data Analytics
Machine Learning
And it has to be selectively better than the challengers

Node
Swift
Go
Python
Ruby
Let’s have a few races
N-Body benchmark

• 1 CPU
• Model the orbits of Jovian planets, using the same simple symplectic-integrator.
• Lots of number crunching
• Quickest to the answer wins

<table>
<thead>
<tr>
<th>Language</th>
<th>seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go</td>
<td>21.47</td>
</tr>
<tr>
<td>Swift</td>
<td>21.96</td>
</tr>
<tr>
<td>Java</td>
<td>22.56</td>
</tr>
<tr>
<td>Node</td>
<td>28.74</td>
</tr>
<tr>
<td>Ruby</td>
<td>12 mins</td>
</tr>
<tr>
<td>Python3</td>
<td>13 mins</td>
</tr>
</tbody>
</table>
• Plot the Mandelbrot set \([-1.5-i, 0.5+i]\) on an 16000 -by- 16000 bitmap. Write output byte-by-byte in portable bitmap format.
<table>
<thead>
<tr>
<th>Language</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swift</td>
<td>3.32</td>
</tr>
<tr>
<td>Go</td>
<td>5.46</td>
</tr>
<tr>
<td>Java</td>
<td>6.08</td>
</tr>
<tr>
<td>Node</td>
<td>19.04</td>
</tr>
<tr>
<td>Python3</td>
<td>273.43</td>
</tr>
<tr>
<td>Ruby</td>
<td>420</td>
</tr>
</tbody>
</table>
Related statistics (python and ruby times removed)

CPU time (secs)
- Java
- Swift
- Node
- Go
- Python3
- Ruby

Memory (k)
- Java
- Swift
- Node
- Go
- Python3
- Ruby
Mandlebrot: CPU usage
A simplistic adaptation of Hans Boehm's GCBench, which in turn was adapted from a benchmark by John Ellis and Pete Kovac.
Ruby
Python3
Go
Node
Swift
Java
<table>
<thead>
<tr>
<th>Language</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swift</td>
<td>4.96</td>
</tr>
<tr>
<td>Java</td>
<td>8.58</td>
</tr>
<tr>
<td>Go</td>
<td>35.18</td>
</tr>
<tr>
<td>Node</td>
<td>53.64</td>
</tr>
<tr>
<td>Ruby</td>
<td>54.24</td>
</tr>
<tr>
<td>Python3</td>
<td>86.1</td>
</tr>
</tbody>
</table>
Binary Tree: CPU usage

CPU %

Java
Swift
Node
Go
Python3
Ruby

CPU1  CPU2  CPU3  CPU4
same simple regex patterns and actions to manipulate FASTA format data

agggtaaa|tttaccc [cgt]gggtaaa|tttacc[acg]
agggtaa[cgt]| [acg]ttaccct

64-bit Ubuntu quad core java 9 Java(TM) SE Runtime Environment (build 9+181) Java HotSpot(TM) 64-Bit Server VM (build 9+181, mixed mode)
<table>
<thead>
<tr>
<th>Language</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>4.4</td>
</tr>
<tr>
<td>Java</td>
<td>10.38</td>
</tr>
<tr>
<td>Python3</td>
<td>14.86</td>
</tr>
<tr>
<td>Ruby</td>
<td>28.8</td>
</tr>
<tr>
<td>Go</td>
<td>29.29</td>
</tr>
<tr>
<td>Swift</td>
<td>75.47</td>
</tr>
</tbody>
</table>
Regex-redux: CPU usage

![CPU usage graph for Java, Swift, Node, Go, Python3, Ruby showing CPU1, CPU2, CPU3, CPU4 usage]
benchmarksgame.alioth.debian.org

‘100 line’ benchmarks

“Non-motivation: We are profoundly uninterested in claims that these measurements, of a few tiny programs, somehow define the relative performance of programming languages.”
App Metrics
Thoughts

Micro benchmarks are fun
But not to be taken completely seriously

But there are trends …
Node.js is winning the cloud IO space.

Non blocking workloads run best with Node.

But don’t do compute intensive activities with it.
Swift is a strong contender for memory constrained devices and arenas (like iOS or Cloud).

But its limited in platform reach.
And does scale as well as Java.
Go looks impressive but it's much more aligned to C arenas

Maybe one day JVMs will have Go in them..
Ruby isn’t a challenger to Java – it’s more comparable with Node.
And its losing out.
(but I still love it)
Python: why is this a language of choice for data scientists and machine learning?

Because it has a wealth of native libraries
IBM is investing in these runtimes because between them they cover all the bases – developer communities and technical capabilities.
But what about

Data Analytics

Machine Learning
Runtime Language  Scripting Language  Modern Native Language
Type Safe means JIT’s or compilers can optimise code significantly better than a dynamically type language.

Since an `int` is always an `int` in Javascript a `int` is a `thing` until the very last minute.
JIT compilers can **optimize** as the workload changes. Pre compiled code can’t do that.
Runtime Language  Scripting Language  Modern Native Language
Type Safe  Dynamically Typed  Type Safe, with Inference
Bytecode: JIT Compiled  JIT Compiled  Pre-Compiled
Garbage Collected  Garbage Collected  Reference Counted

GC can run in the background at the cost of a buffer of memory. Can be parallelized more effectively and more accurately (no circular references)
Single threaded means no locking or synchronization needed. But CPU workload is a major Achilles heel.
Runtime Language  Scripting Language  Modern Native Language
Type Safe  Dynamically Typed  Type Safe, with Inference
Bytecode: JIT Compiled  JIT Compiled  Pre-Compiled
Garbage Collected  Garbage Collected  Reference Counted
Concurrent Threaded  Single Thread  Concurrent Work Pool
All Platforms  All Platforms  Apple Platforms and Linux
Runtime Language
Type Safe
Bytecode: JIT Compiled
Garbage Collected
Concurrent Threaded
All Platforms

These characteristics let us take Java anywhere.

No other runtime environment comes close
Our cunning plan ™
Recent changes are a sign of the ecosystem positioning for a faster pace

- predictable
- consistent
- cadence
- easier migration
- increased
- innovation
Everyone can engage in the future of Java development.

Why, and How?

Every development team has both common and unique problems to solve.

Open source is key to fast innovation and adoption.

OpenJDK
Eclipse OpenJ9
Open Liberty
Eclipse MicroProfile
Java EE
IBM Cloud
Docker
Kubernetes
Giving Java innovation a faster cadence

lambda streams
modules reactive streams
panama valhalla penrose amber
And a variety of implementations to choose from

<table>
<thead>
<tr>
<th>Tomcat</th>
<th>OpenJDK + Hotspot</th>
<th>J2EE Micro-profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glassfish</td>
<td>OpenJDK + OpenJ9</td>
<td></td>
</tr>
<tr>
<td>Open Liberty</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
We’re going to take Java to places it’s never been before.

And you’re coming too.
Where code goes, where data flows, cognition will follow.
CONSIDER:

Cognitive systems can understand the world through sensing and interaction, reason using hypotheses and arguments and learn from experts and through data. Watson is the most advanced such system.

Today, businesses in 36 countries across.

17 industries are applying cognitive technologies.

There are 350+ Watson ecosystem partner companies, with 100 of those have taken their product to market.

78% of business and IT executives believe that successful business will manage employees alongside intelligent machines.

There are 350+ Watson ecosystem partner companies, with 100 of those have taken their product to market.

On average there are 1.3B Watson API calls a month and growing.

Among C-Suite executives familiar with cognitive computing:

- 96% in insurance intend to invest in cognitive capabilities.
- 84% in healthcare believe it will play a disruptive role in the industry, and 60% believe they lack the skilled professionals and technical experience to achieve it.
- 94% in retail intend to invest in cognitive capabilities.
- 89% in telecommunications believe it will have a critical impact on the future of their business.

© 2015 INTERNATIONAL BUSINESS MACHINES CORPORATION
We can now confer on every digitized object, product, process and service a kind of thinking ability.

How, and why now?

Data is transforming industries and professions.

The world is being reinvented in code.

Computing is entering a new Cognitive Era.
Worried yet?

Cloud: Makes you think differently but mostly about footprint, resilience and scaling

Modularity, Lambdas
J2EE going to Eclipse

That’s not really a big change.

How do you do things like…
Imagine create systems that can

- Tailoring responses to the personalities of your customers without meeting a single one of them.
- Knowing the latest, most significant developments in your profession or industry the moment they take place.
- Products and services that improve themselves over time, learning from and adapting to the world around them.
- Processes that identify their own inefficiencies and address them automatically in real time.
- Uncovering patterns, resources, trends and other competitive advantages invisible to competitors and their information systems.
The biggest challenge for all of us is learning to solve problems differently
PROBLEM:

...so let's solve a word search

java
ibm
cognitive
CLASSIC SOLUTION:

```java
for (int x=0; x < grid_width; x++)
{
    for (int y=0; y< grid_height; y++)
    {
        for (String word : words)
        {
            if(does_match_first_letter(x, y, word))
            {
                if(match_whole_word(x,y,word))
                {
                    System.out.println("Found word ! : "+word+" at "+x", "+y);
                }
            }
        }
    }
}
```

java
ibm
cognitive
PARTITIONING:

cognitive

Executor node

Executor node

Executor node

Executor node

java

ibm
PARTITIONING:

Executor node

Executor node

Executor node

Executor node

java
ibm
cognitive

CPU
Improving Java application performance with GPU exploitation is available in IBM SDK for Java 8 and OpenJDK 9 with Eclipse Open9

Standard SE API optimisation as well as CUDA4J API for explicit low level control

new Java APIs

- CudaDevice — a CUDA capable GPU device
- CudaBuffer — a region of memory on the GPU
- CudaModule — user library of kernels to load into GPU
- CudaKernel — for launching a device function
- CudaFunction — a kernel's entry point
- CudaEvent — for timing and synchronization
- CudaException — for when something goes wrong
GPU’s don’t work like CPU’s
They want their data in different forms
They behave differently
You’ll have to think differently too
NEURAL:

java
ibm
cognitive

DL4J
DEEPLEARNING4J

© 2015 INTERNATIONAL BUSINESS MACHINES CORPORATION
SYNAPSE:

A program to develop a neuromorphic processor that is a new kind of cognitive computer

Designed to simulate the neurones and dendrites of the brain for low power efficient operation
Event driven, Non Von Neumann Neural Network.

Neural Nets want their data in different forms
They behave differently

You’ll have to think differently too
Now it gets even stranger..
In 1996 a search algorithm was defined by Lov Grover. This algorithm can transform the problem into an $O(\sqrt{N})$ search.

On a traditional computer this search problem is solved in no fewer than $O(N)$ evaluations.
In 1996 a search algorithm was defined by Lov Grover. This algorithm can transform the problem into an $O(\sqrt{N})$ search.

On a traditional computer this search problem is solved in no fewer than $O(N)$ evaluations.
On a traditional computer this search problem is solved in no fewer than $O(N)$ evaluations.

In 1996 a search algorithm was defined by Lov Grover. This algorithm can transform the problem into an $O(\sqrt{N})$ search.
QUANTUM:

www.research.ibm.com/ibm-q/
Quantum Computers can and will solve linear equations break cryptographic systems or model new medicines In times that are a fraction of what existing computers can achieve now.

Word searches are easy. But you present the data differently and get a statistical response.

No more 0&1’s
You can play battleships with QC’s

But damage is not HIT or MISS

It’s +/- % hit

Maybe you sunk my battleship.

Maybe you didn’t
The JVM design means that we can easily imagine running Java on new forms of processors. It may not end up being today's Java but it will be close.

The challenge is that you have to change how you think about problems.
OPPORTUNITY:

Data flows from every device, replacing guessing and approximations with precise information. Yet 80% of this data is unstructured; therefore, invisible to computers and of limited use to business.

By 2020, **1.7 MB** of new information will be created **every second** for **every human being** on the planet.

<table>
<thead>
<tr>
<th>HEALTHCARE DATA</th>
<th>GOVERNMENT &amp; EDUCATION DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>99%</strong> growth by 2017</td>
<td><strong>94%</strong> growth by 2017</td>
</tr>
<tr>
<td><strong>88%</strong> unstructured</td>
<td><strong>84%</strong> unstructured</td>
</tr>
</tbody>
</table>

Healthcare data comes from sources such as:

- Patient Sensors
- Electronic Medical Records
- Test Results

Government & education data comes from sources such as:

- Vehicle Fleet Sensors
- Traffic Sensors
- Student Evaluations

<table>
<thead>
<tr>
<th>UTILITIES DATA</th>
<th>MEDIA DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>93%</strong> growth by 2017</td>
<td><strong>97%</strong> growth by 2017</td>
</tr>
<tr>
<td><strong>84%</strong> unstructured</td>
<td><strong>82%</strong> unstructured</td>
</tr>
</tbody>
</table>

Utilities data comes from sources such as:

- Utility Sensors
- Employee Sensors
- Location Data

Media data comes from sources such as:

- Video and Film
- Images
- Audio
The world is being reinvented in code. Java code.
Computing is entering a new cognitive era.

What do you Think when you solve a problem?
Quantum enabled, Neural Networked, Clustered, Containerized, Analysed, Cloudified, Toaster 4J
Trademarks, notes and disclaimers

© IBM Corporation 2017

• IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (© or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

• Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

• Other company, product, and service names may be trademarks or service marks of others.

• References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.