Java 2 Days

IS THE FUTURE OF JAVA CLOUDY?

Steve Poole

Developer Advocate

IBM Runtime Technologies

@spoole167

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

‘palm’	OS	‘OS390’
’K’	Memory	‘T’
‘toaster’	Device	‘Mainframe’
‘midlets’	Display	‘Headless’
‘arm’	Architecture	‘s390’

Over the last few years Java's dominance has faltered

Partly because squeezing a JVM into small devices is hard (and we cut corners)

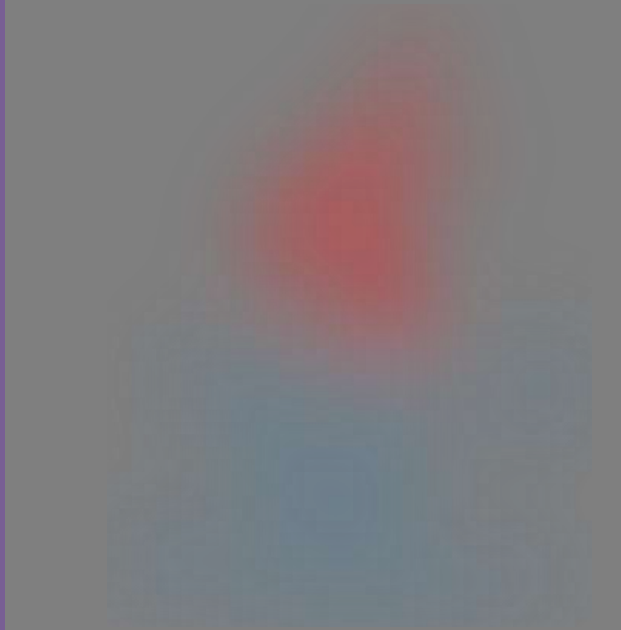
Partly because Java is seen as ‘old’

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

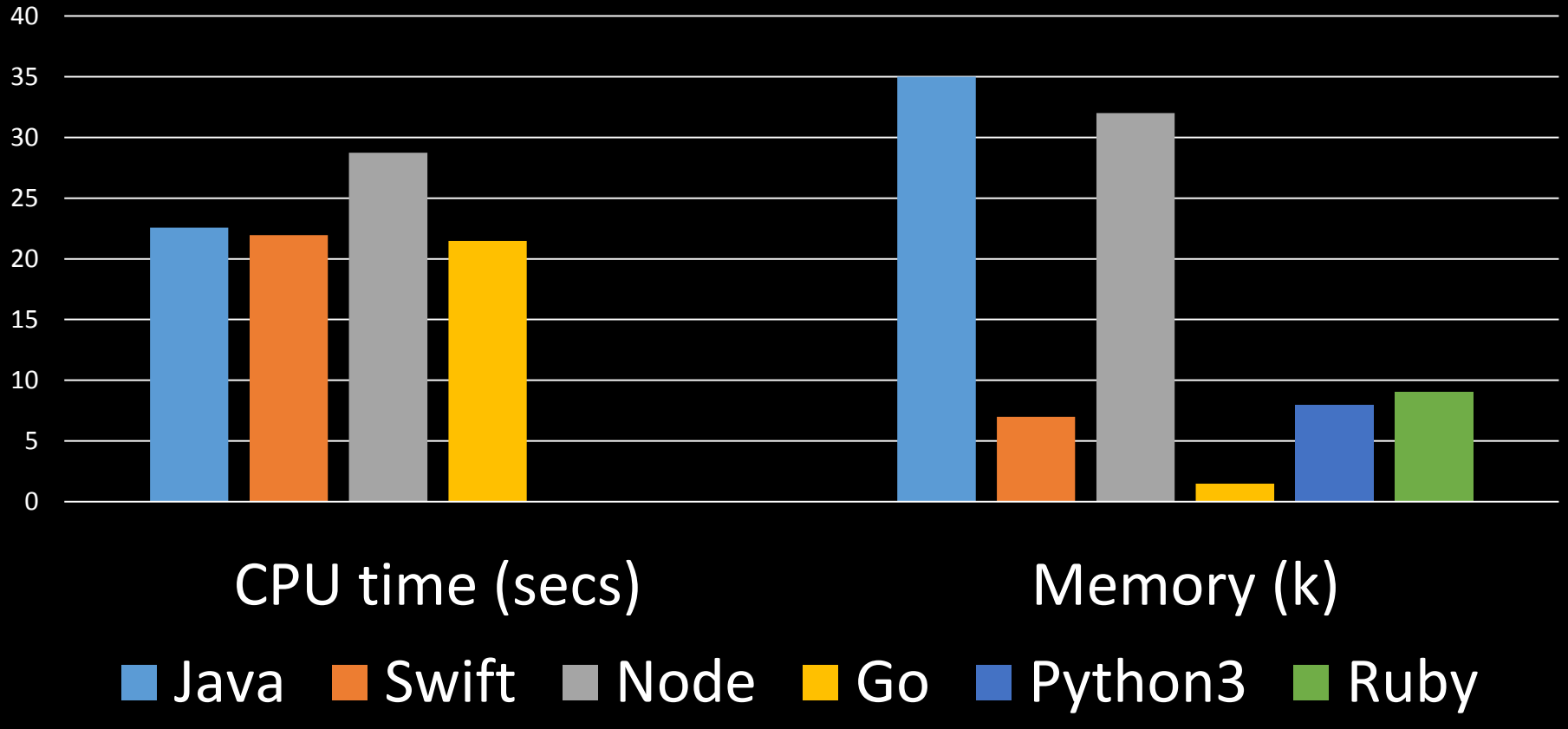
- 1 CPU
- Model the orbits of Jovian planets, using the same simple symplectic-integrator.
- Lots of number crunching
- Quickest to the answer wins

https://en.wikipedia.org/wiki/Outer_planets#/media/File:Gas_giants_in_the_solar_system.jpg

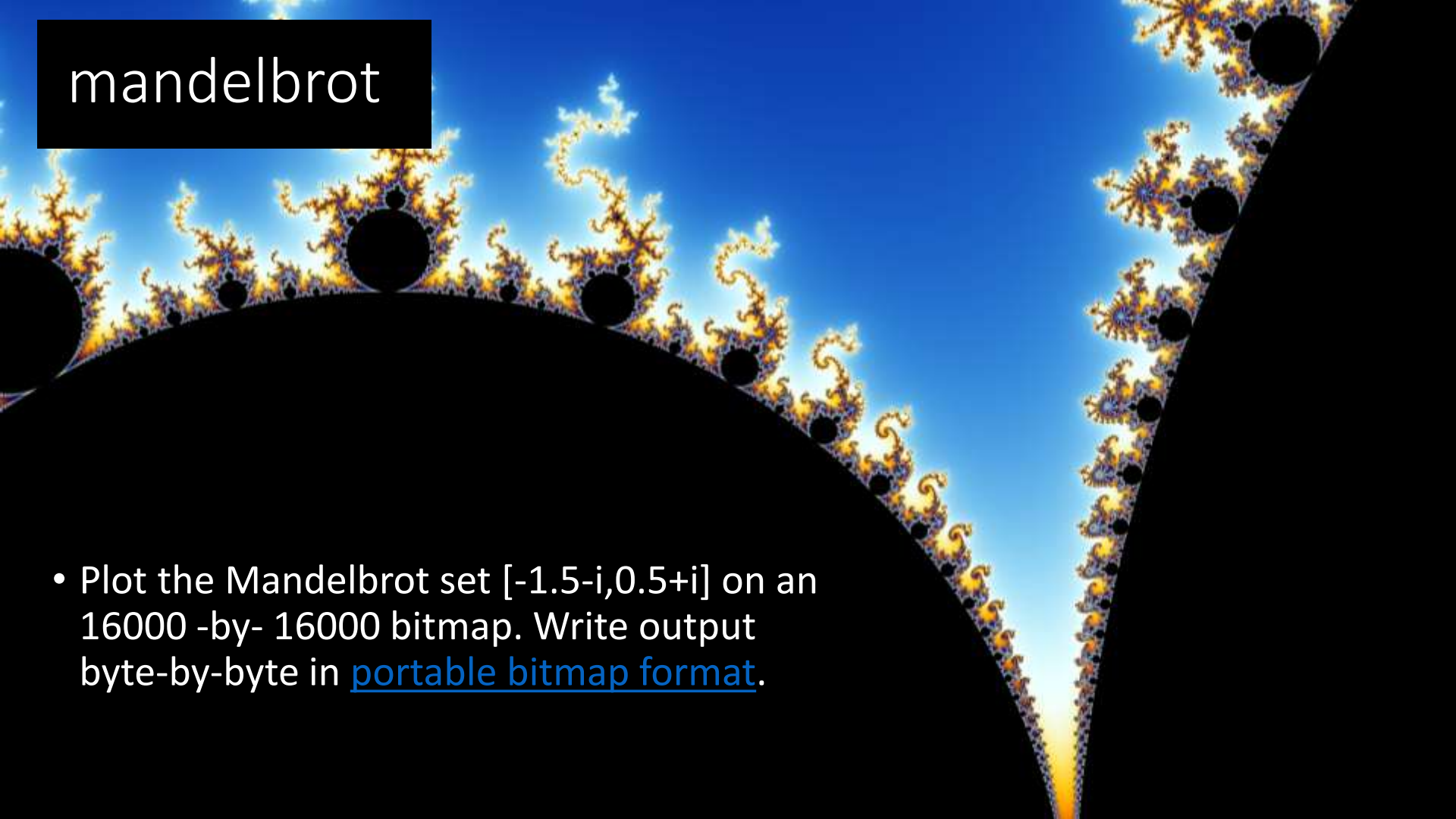

Ruby
Python3
Go
Node
Swift
Java
0
5
10
15
20
25
30

Language	seconds
Go	21.47
Swift	21.96
Java	22.56
Node	28.74
Ruby	12 mins
Python3	13 mins

Related statistics (python and ruby times removed)



mandelbrot

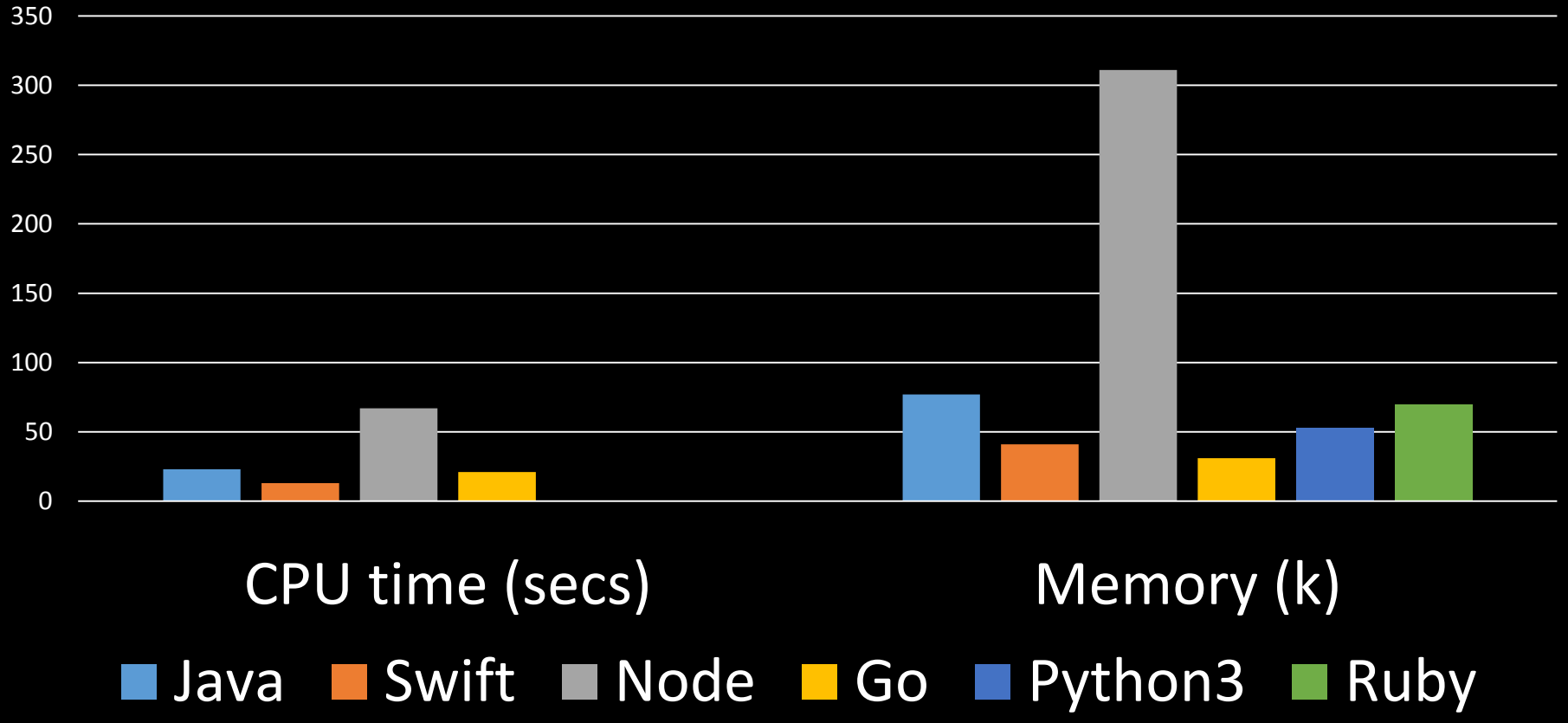


- Plot the Mandelbrot set [-1.5-i,0.5+i] on an 16000 -by- 16000 bitmap. Write output byte-by-byte in portable bitmap format.

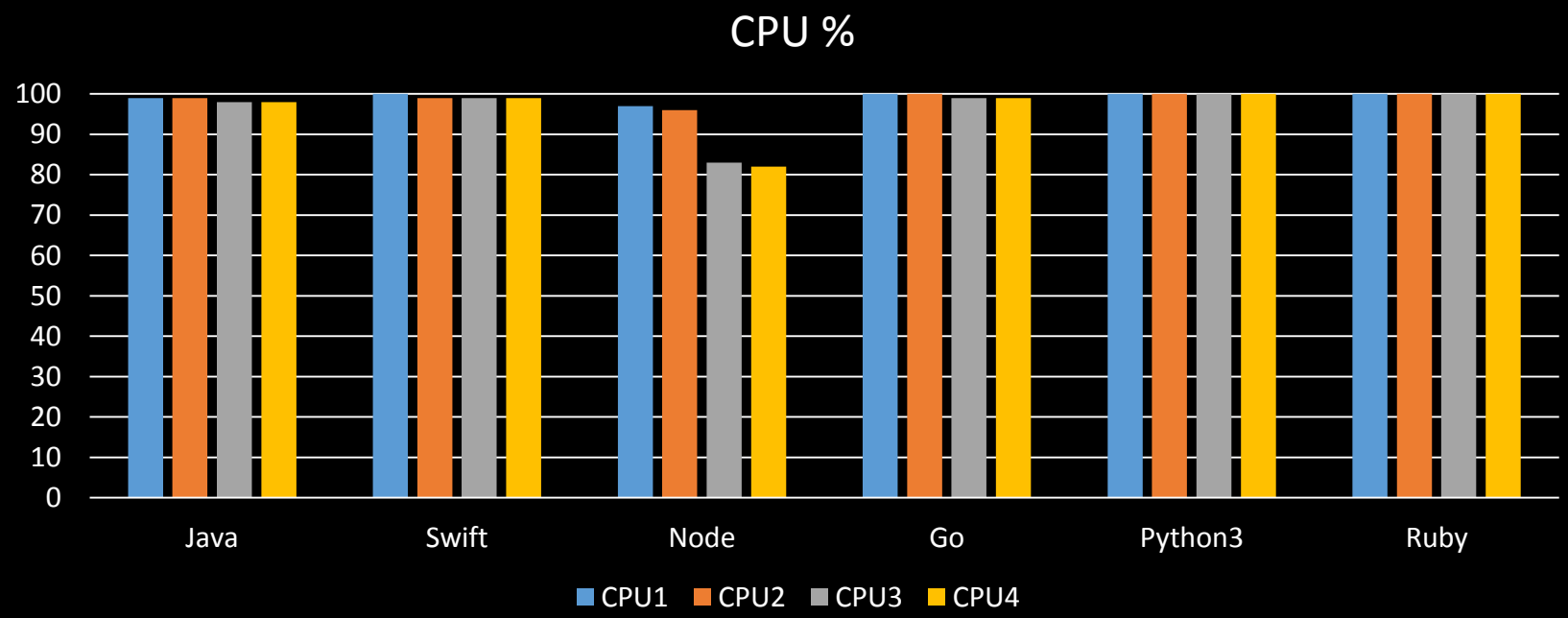

Ruby
Python3
Go
Node
Swift
Java
0
5
10
15
20
25
30

Language	Score
Swift	3.32
Go	5.46
Java	6.08
Node	19.04
Python3	273.43
Ruby	420

Related statistics (python and ruby times removed)

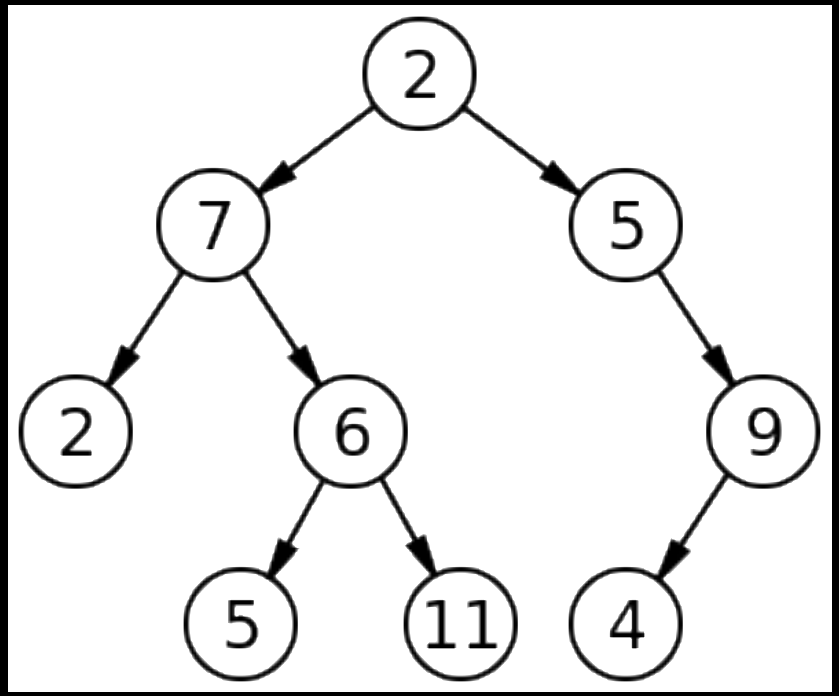


Mandlebrot: CPU usage



binary tree

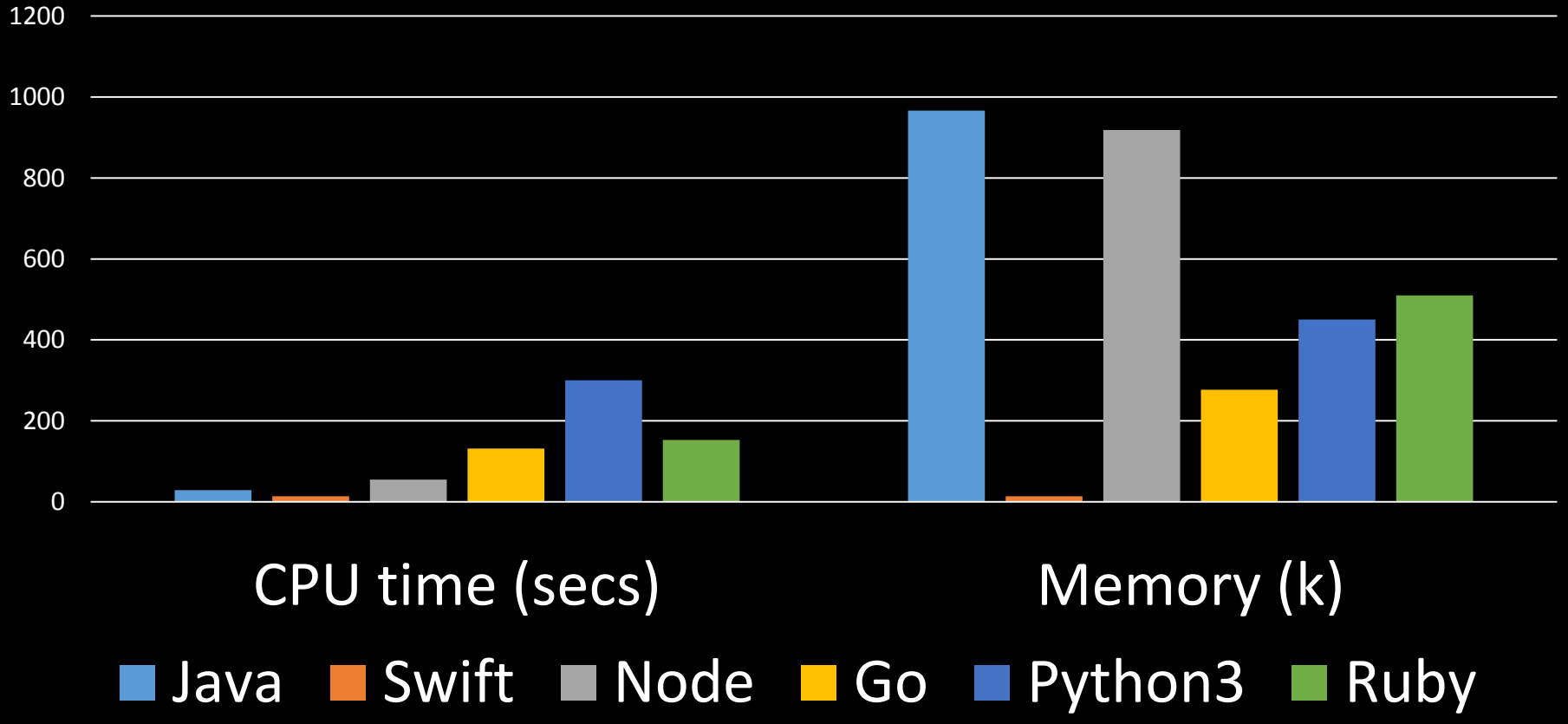
A simplistic adaptation of Hans Boehm's GCBench, which in turn was adapted from a benchmark by John Ellis and Pete Kovac.



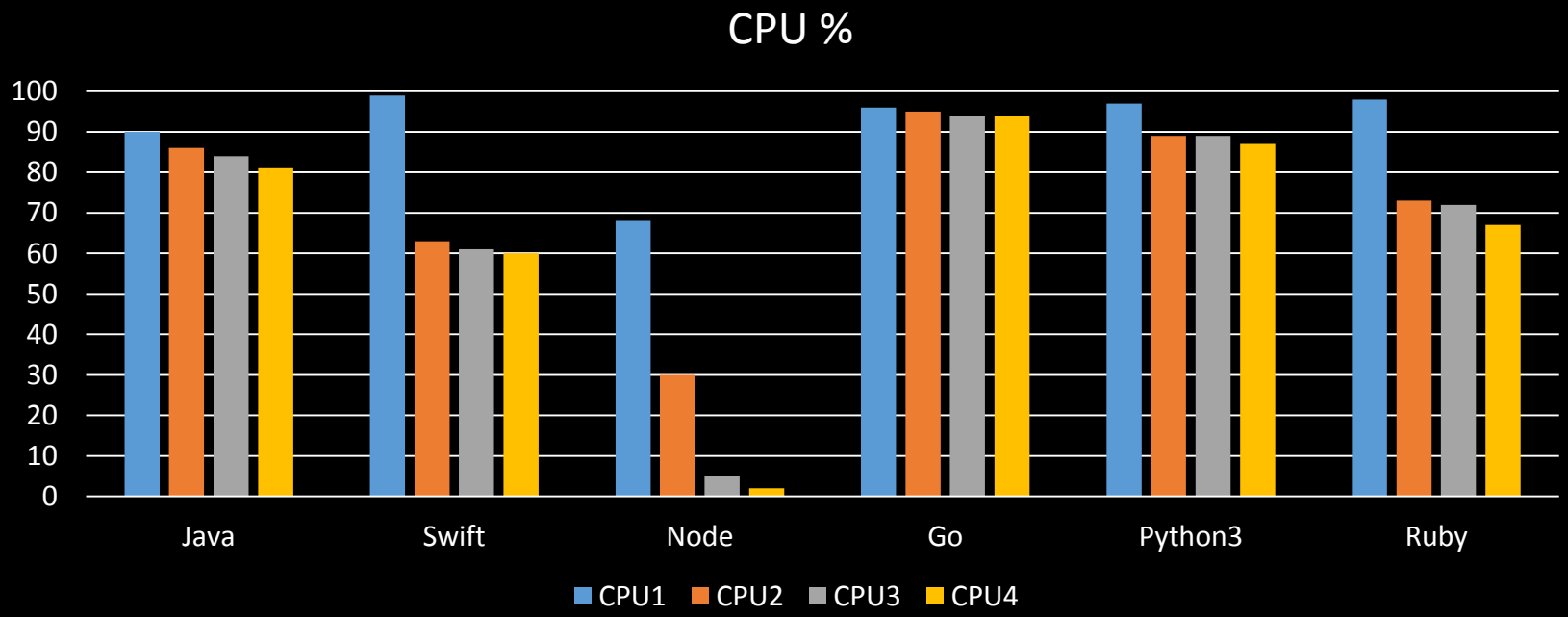

Ruby
Python3
Go
Node
Swift
Java
0
5
10
15
20
25
30

Language	Score
Swift	4.96
Java	8.58
Go	35.18
Node	53.64
Ruby	54.24
Python3	86.1

Related statistics



Binary Tree: CPU usage



Regex-redux

same simple regex patterns and actions to manipulate FASTA format data

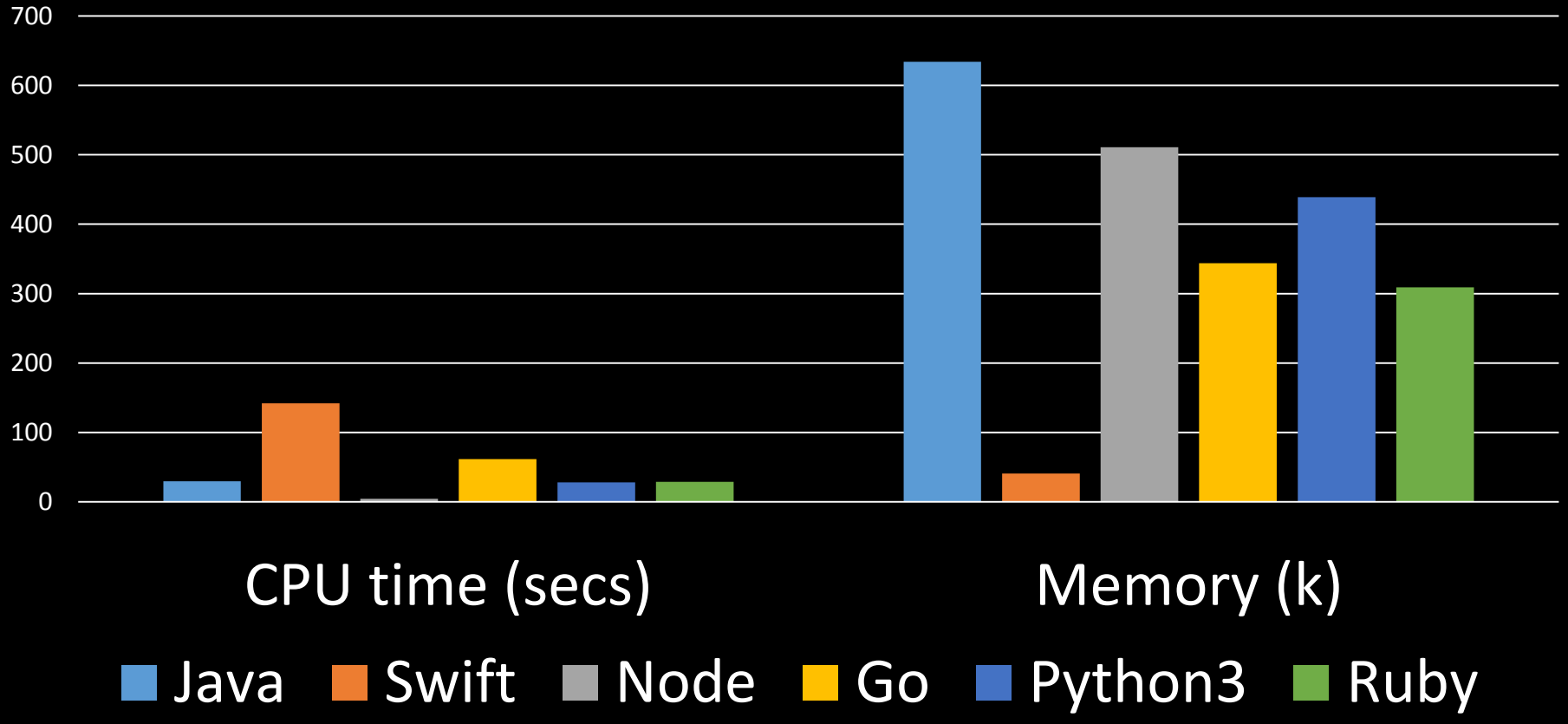
64-bit Ubuntu quad core java 9 Java(TM) SE Runtime Environment (build 9+181) Java HotSpot(TM) 64-Bit Server VM (build 9+181, mixed mode)

agggtaaa|tttaccct [cgt]gggtaaa|tttaccc[acg]
a[act]ggtaaa|tttacc[agt]t ag[act]gtaaa|tttac[agt]ct
agg[act]taaa|ttta[agt]cct aggg[acg]aaa|ttt[cgt]ccct
agggt[cgt]aa|tt[acg]accct agggta[cgt]a|t[acg]taccct
agggtaa[cgt]|[acg]ttaccct

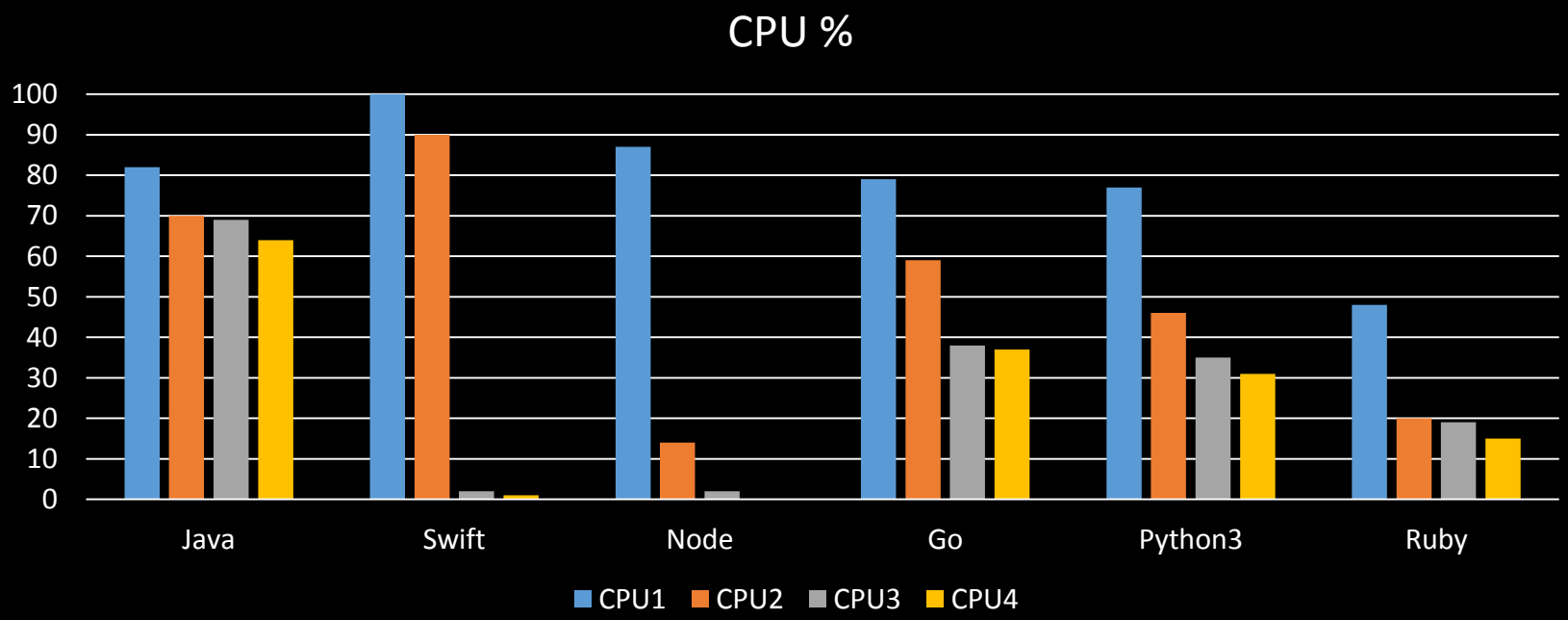

Ruby
Python3
Go
Node
Swift
Java
0
5
10
15
20
25
30

Language	Score
Node	4.4
Java	10.38
Python3	14.86
Ruby	28.8
Go	29.29
Swift	75.47

Related statistics



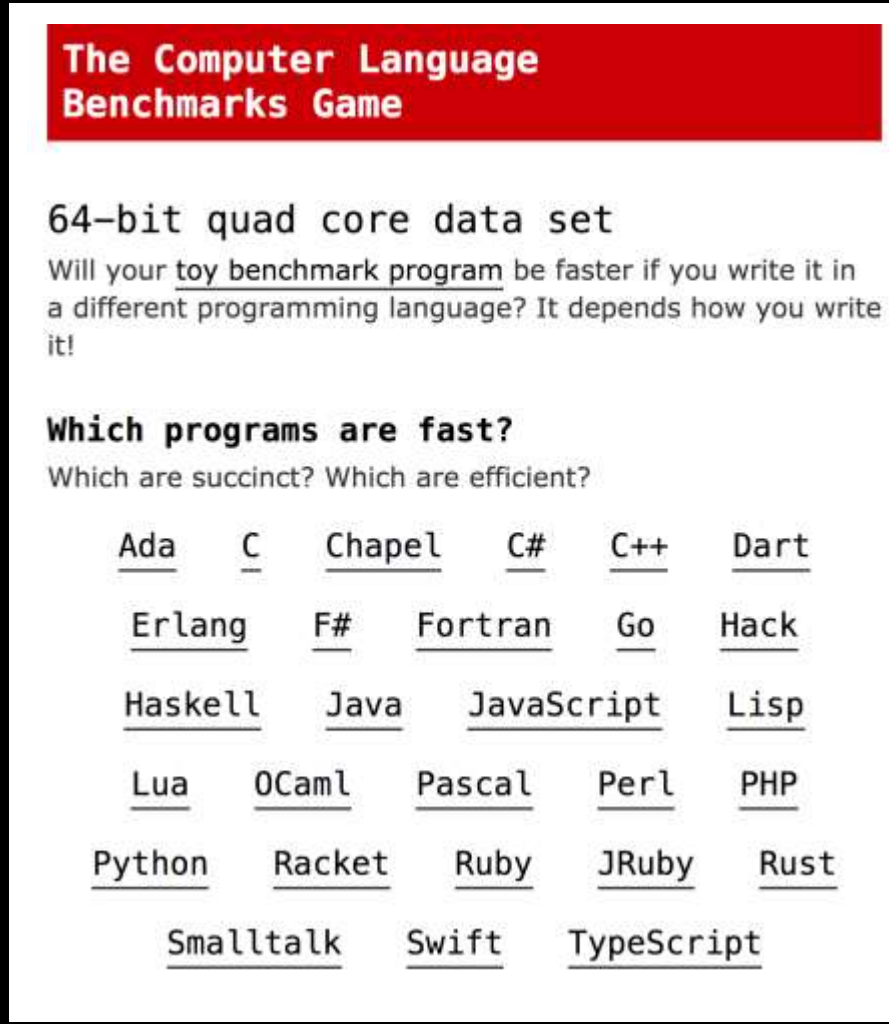
Regex-redux: CPU usage



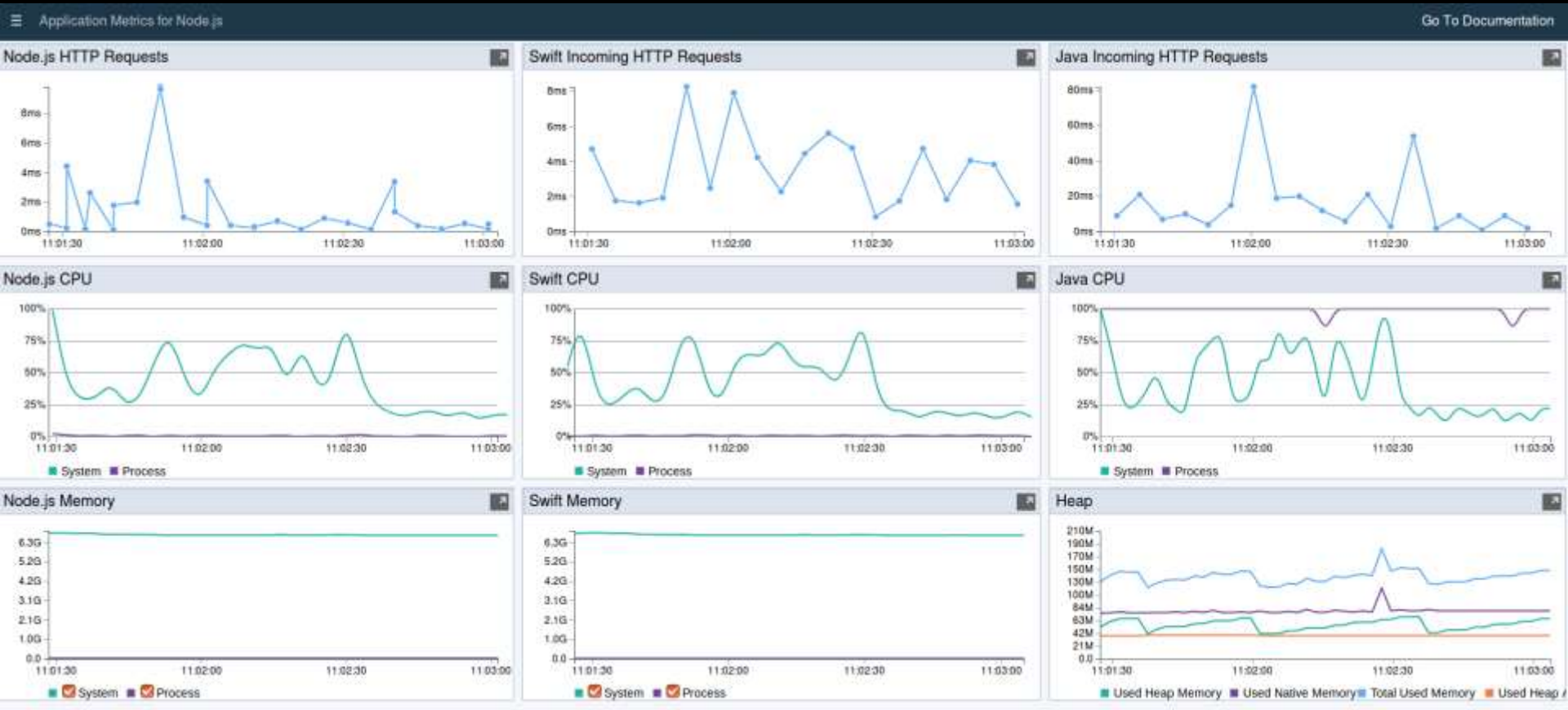
benchmarksgame.alioth.debian.org

‘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


Application Metrics for Node.js
Go To Documentation
Node.js HTTP Requests
8ms
6ms
4ms
2ms
0ms
11:01:30
11:02:00
11:02:30
11:03:00
Swift Incoming HTTP Requests
8ms
6ms
4ms
2ms
0ms
11:01:30
11:02:00
11:02:30
11:03:00
Java Incoming HTTP Requests
80ms
60ms
40ms
20ms
0ms
11:01:30
11:02:00
11:02:30
11:03:00
Node.js CPU
100%
75%
50%
25%
0%
11:01:30
11:02:00
11:02:30
11:03:00
System
Process
Swift CPU
100%
75%
50%
25%
0%
11:01:30
11:02:00
11:02:30
11:03:00
System
Process
Java CPU
100%
75%
50%
25%
0%
11:01:30
11:02:00
11:02:30
11:03:00
System
Process
Node.js Memory
6.3G
5.2G
4.2G
3.1G
2.1G
1.0G
0.0
11:01:30
11:02:00
11:02:30
11:03:00
System
Process
Swift Memory
6.3G
5.2G
4.2G
3.1G
2.1G
1.0G
0.0
11:01:30
11:02:00
11:02:30
11:03:00
System
Process
Heap
210M
190M
170M
150M
130M
100M
84M
63M
42M
21M
0.0
11:01:30
11:02:00
11:02:30
11:03:00
Used Heap Memory
Used Native Memory
Total Used Memory

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.

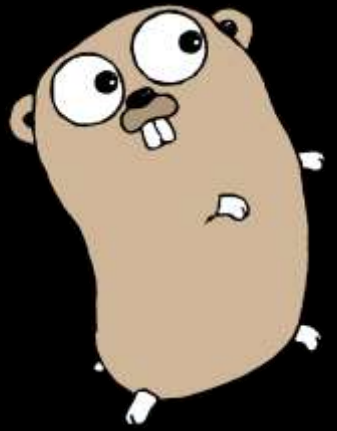
But don’t do compute intensive activities with it

Swift is strong contender for memory constrained devices and arenas (like iOS or Cloud)

But its limited in platform reach
And does scale as well as

Go looks impressive but its much more aligned to C arenas

Maybe one day JVMs will have Go in them..



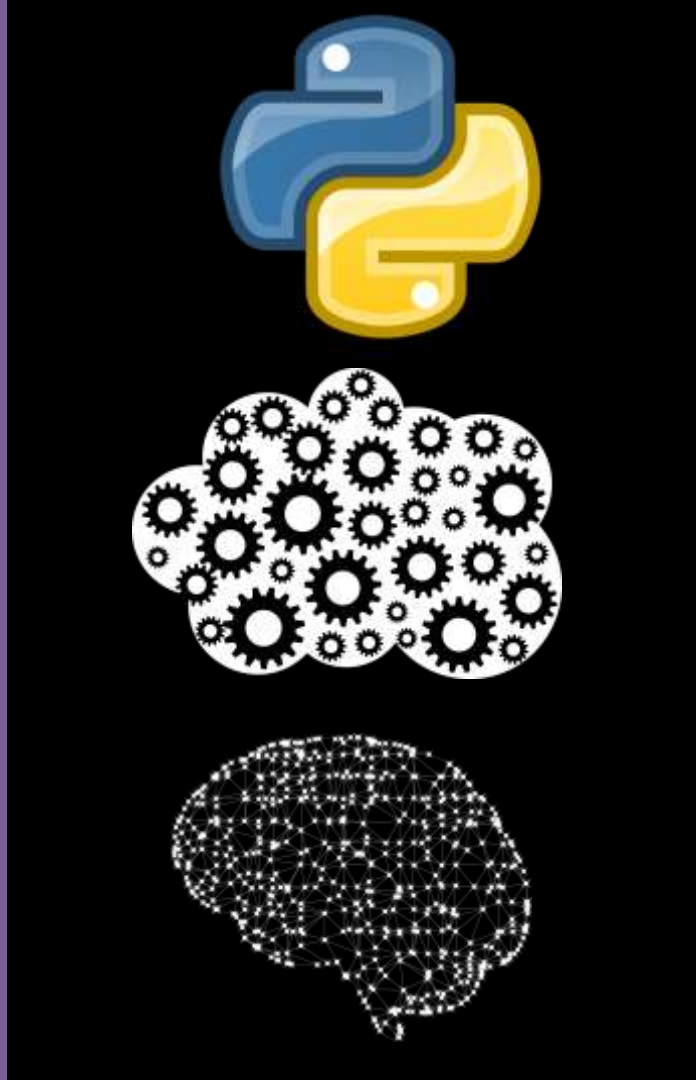
Ruby isn’t a challenger to Java – it’s more comparable with Node.

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

THE C PROGRAMMING LANGUAGE

THE C PROGRAMMING LANGUAGE

THE C PROGRAMMING LANGUAGE

Runtime Language

Scripting Language

Modern Native Language

Runtime Language

Type Safe

Scripting Language

Dynamically Typed

Modern Native Language

Type Safe, with Inference

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

Runtime Language

Type Safe

Bytecode: JIT Compiled

Scripting Language

Dynamically Typed

JIT Compiled

Modern Native Language

Type Safe, with Inference

Pre-Compiled

JIT compilers can optimize as the workload changes.
Pre compiled code can’t do that

Runtime Language

Type Safe

Bytecode: JIT Compiled

Garbage Collected

Scripting Language

Dynamically Typed

JIT Compiled

Garbage Collected

Modern Native Language

Type Safe, with Inference

Pre-Compiled

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)

Runtime Language

Type Safe

Bytecode: JIT Compiled

Garbage Collected

Concurrent Threaded

Scripting Language

Dynamically Typed

JIT Compiled

Garbage Collected

Single Thread

Modern Native Language

Type Safe, with Inference

Pre-Compiled

Reference Counted

Concurrent Work Pool

Single threaded means no locking or syncronisation needed. But CPU workload is a major Achilles heel

Runtime Language

Type Safe

Bytecode: JIT Compiled

Garbage Collected

Concurrent Threaded

All Platforms

Scripting Language

Dynamically Typed

JIT Compiled

Garbage Collected

Single Thread

All Platforms

Modern Native Language

Type Safe, with Inference

Pre-Compiled

Reference Counted

Concurrent Work Pool

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

Tomcat
Glassfish
Open Liberty

OpenJDK +
Hotspot

OpenJDK +
OpenJ9

J2EE
Micro-profile

We're going to take Java to places its 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.

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:

in **insurance** intend to invest in cognitive capabilities.

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.

in **retail** intend to invest in cognitive capabilities.

in **telecommunications** believe it will have a critical impact on the future of their business.

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.

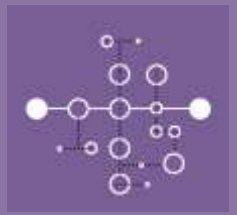
How do you do things like…

Imagine create systems that can

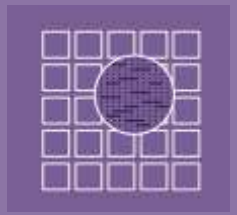
Tailoring responses to the personalities of your customers without meeting a single one of them.

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 lets solve a word search

a	e	k	j	c	b	a	o	a	j
e	j	a	x	l	o	n	d	od	n
v	n	a	v	a	a	j	v	i	o
i	c	me	i	e	t	u	p	a	a
t	d	a	t	a	j	a	j	k	l
i	t	w	i	q	a	y	a	m	h
n	u	e	n	x	v	d	v	g	a
g	o	d	g	t	f	f	a	u	i
o	a	s	o	g	a	g	t	v	c
c	i	a	c	h	j	a	i	b	m

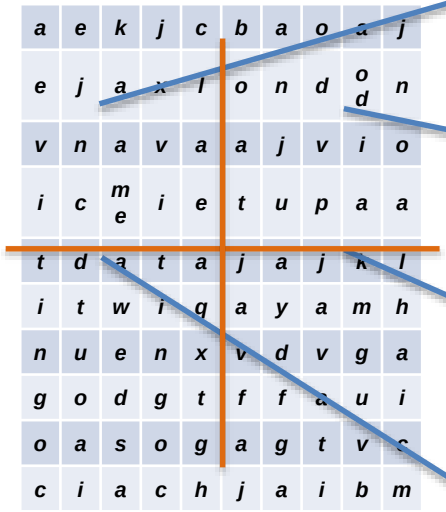
java
ibm
cognitive

CLASSIC SOLUTION:

java
ibm
cognitive

```
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);
                }
            }
        }
    }
}
```

PARTITIONING:



java
ibm
cognitive

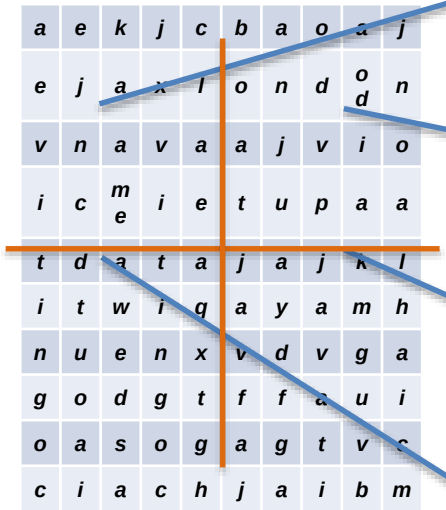
Executor node

Executor node

Executor node

Executor node

PARTITIONING:



java
ibm
cognitive

Executor node

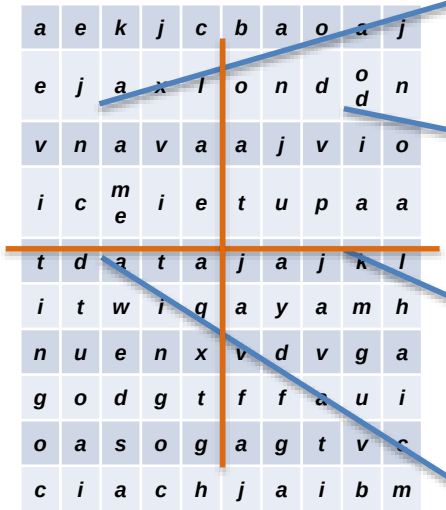
Executor node

Executor node

Executor node

CPU

PARTITIONING:



java
ibm
cognitive

Executor node

Executor node

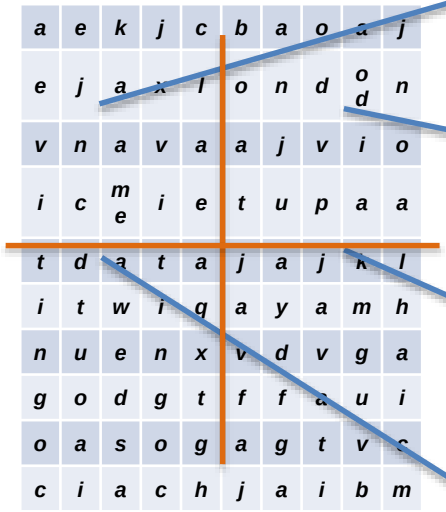
Executor node

Executor node

CPU

GPU

PARTITIONING:



java
ibm
cognitive

Executor node

Executor node

Executor node

Executor node

CPU

GPU

FPGA

ASIC

hadoop

APACHE Spark

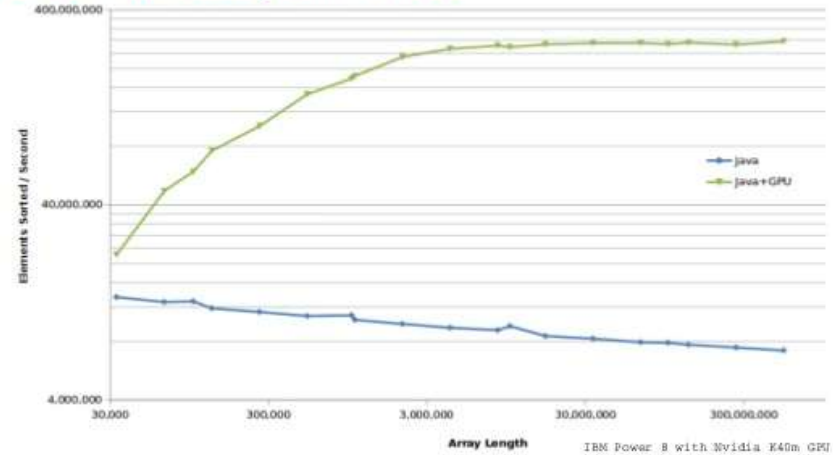
IBM + NVIDIA

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

+

GPU-enabled array sort method



21


```
IntStream.range(0, N).parallel().forEach(i -> c[i] = a[i] + b[i]);
```

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

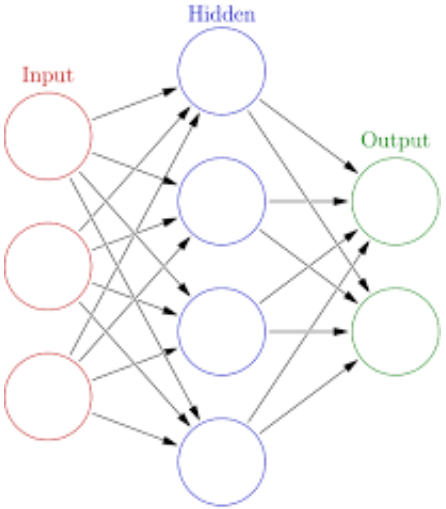
They want their data in different forms
They behave differently

You’ll have to think differently too

NEURAL:

a	e	k	j	c	b	a	o	a	j
e	j	a	x	l	o	n	d	o d	n
v	n	a	v	a	a	j	v	i	o
i	c	m e	i	e	t	u	p	a	a
t	d	a	t	a	j	a	j	k	l
i	t	w	i	q	a	y	a	m	h
n	u	e	n	x	v	d	v	g	a
g	o	d	g	t	f	f	a	u	i
o	a	s	o	g	a	g	t	v	c
c	i	a	c	h	j	a	i	b	m

java
ibm
cognitive



DL4J
DEEPLEARNING4J

Neuroph
Java Neural Network Framework

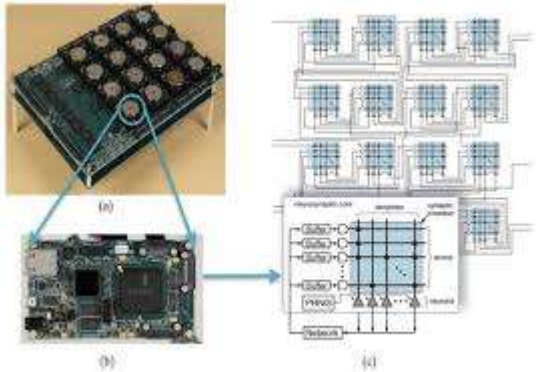
IBM Watson

TensorFlow

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



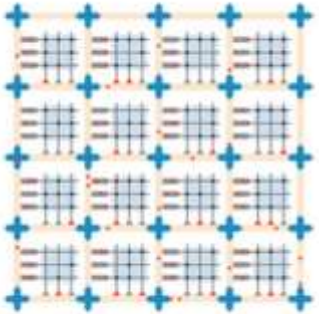
Different from a standard chip

Traditional chips run all of the time This new neurosynaptic chip is event-driven and **operates only when it needs to**, resulting in a cooler operating environment and lower energy use.

The neurosynaptic chip veers from the traditional von Neumann architecture, which inherently creates a bottleneck limiting performance of the system.

New architecture

IBM's brain-inspired architecture consists of a network of neurosynaptic cores. Cores are distributed and operate in parallel. Cores operate—without a clock—in an event-driven fashion. Cores integrate memory, computation, and communication. Individual cores can fail and yet, like the brain, the architecture can still function. Cores on the same chip communicate with one another via an on-chip event-driven network. Chips communicate via an inter-chip interface leading to seamless scalability like the cortex, enabling creation of scalable neuromorphic systems.



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

IBM

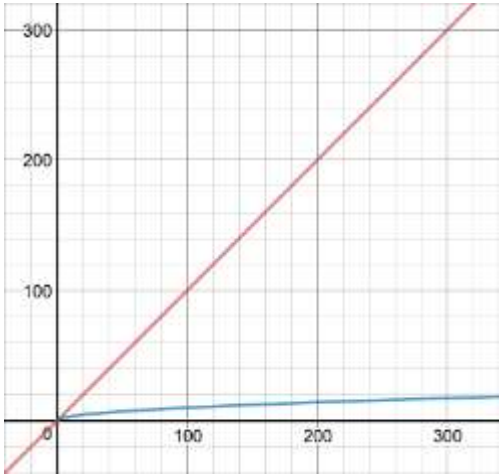
Now it gets even stranger..

SEARCH

SEARCH

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

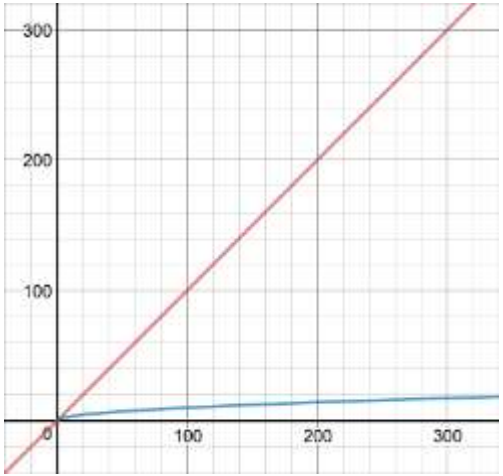
In 1996 a search algorithm was defined by Lov Grover. This algorithm can transform the problem into an O(√N) search.

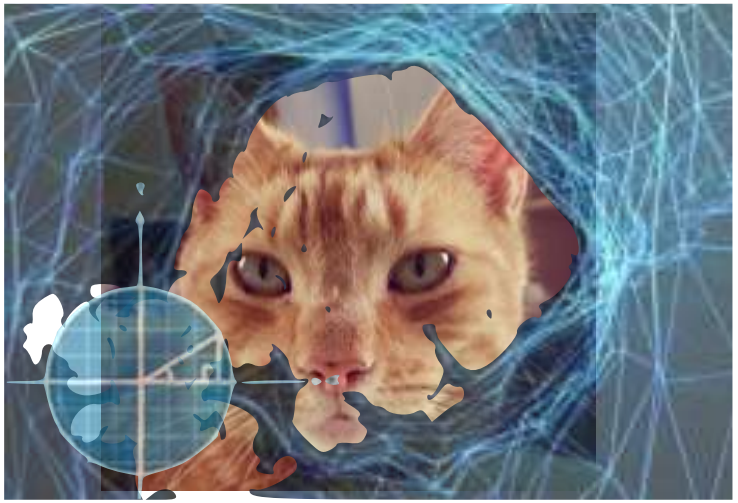


SEARCH

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(√N) search.



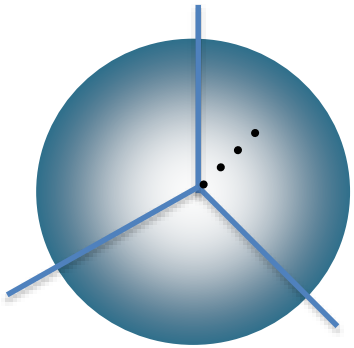


bit

1

0

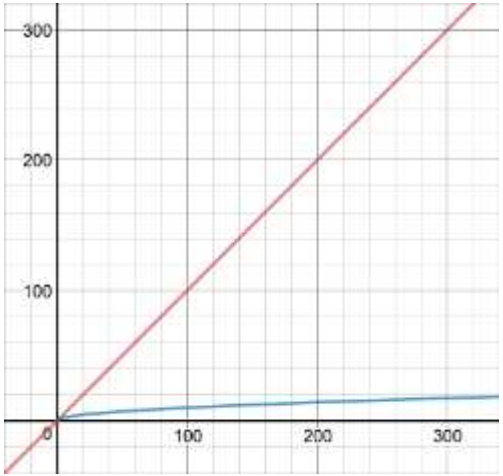
qubit

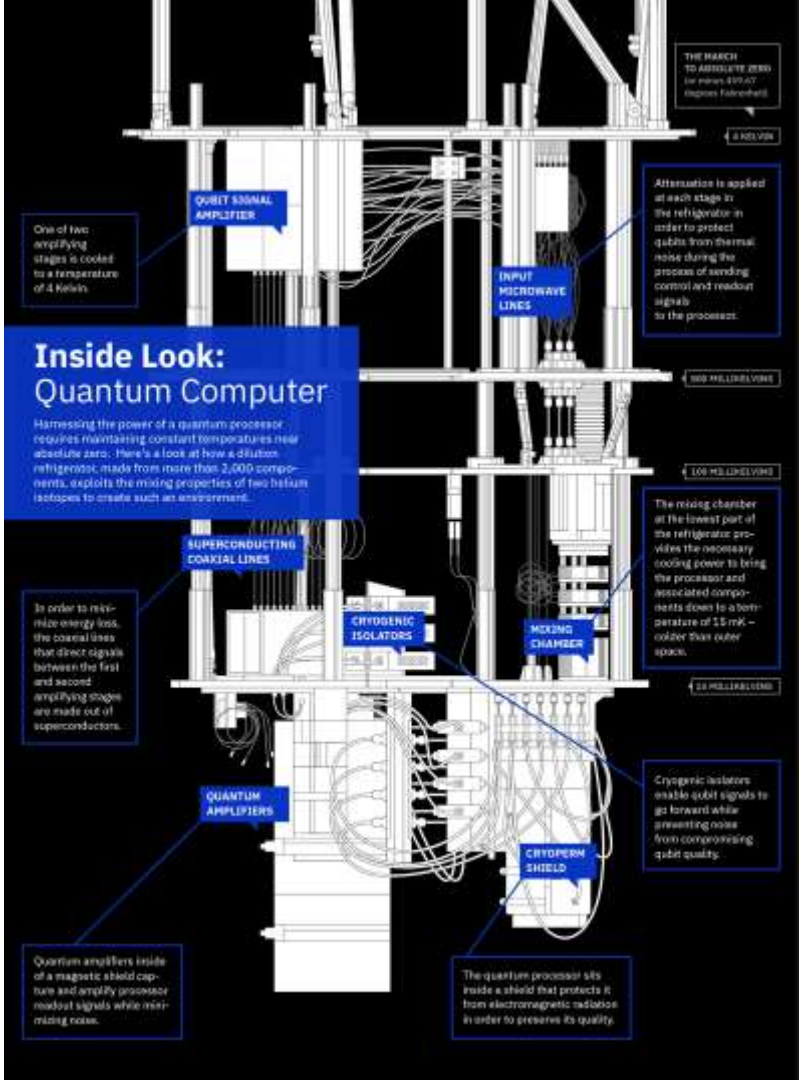


SEARCH

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(√N) search.



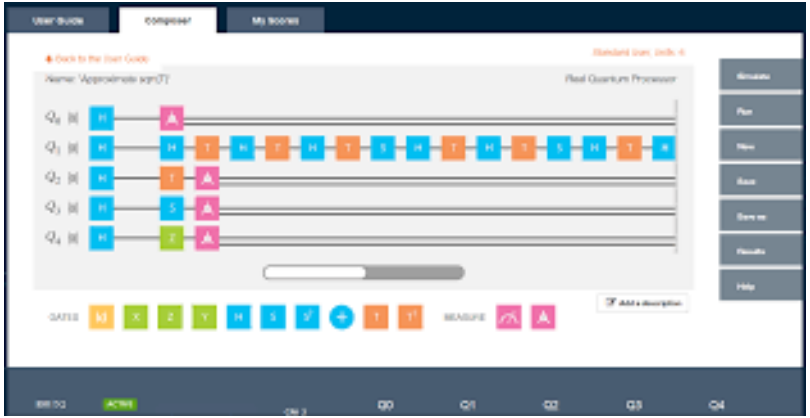


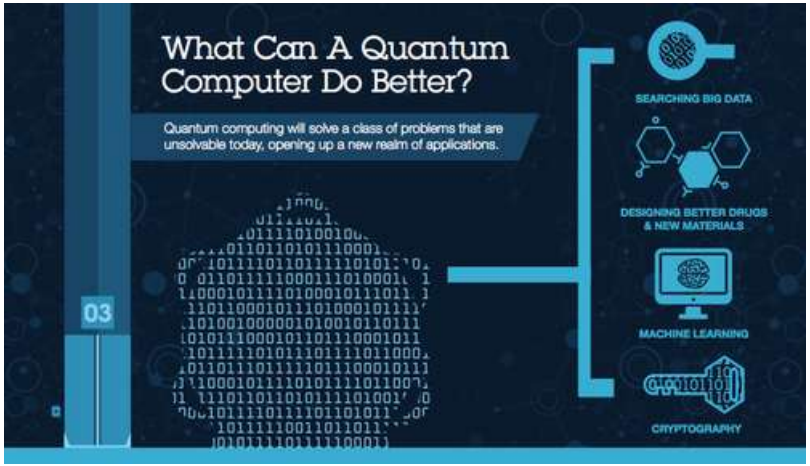
QUANTUM:

a	e	k	j	c	b	a	o	a	j
l	g	m	e	e	t	u	p	a	p
l	n	a	v	a	a	j	v	i	o
p	c	me	i	e	t	u	p	a	a
r	d	a	t	a	j	a	j	k	l
u	t	w	i	q	a	y	a	m	h
d	u	e	n	x	v	d	v	g	a
j	o	d	g	t	f	f	a	u	i
a	a	s	o	g	a	g	t	v	c
z	i	a	c	h	j	a	i	b	m

java
ibm
cognitive

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

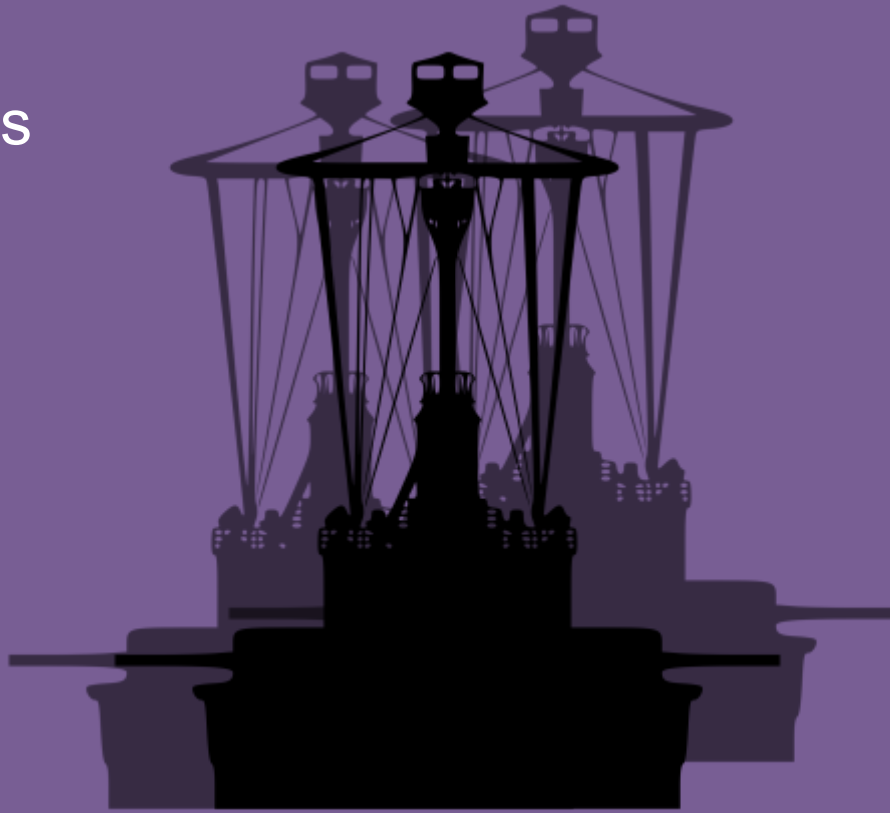
You can play battleships with QC’s

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 todays 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.

HEALTHCARE DATA

99% growth by 2017

88% unstructured

Healthcare data comes from sources such as:

Patient Sensors | Electronic Medical Records | Test Results

GOVERNMENT & EDUCATION DATA

94% growth by 2017

84% unstructured

Government & education data comes from sources such as:

Vehicle Fleet Sensors | Traffic Sensors | Student Evaluations

UTILITIES DATA

93% growth by 2017

84% unstructured

Utilities data comes from sources such as:

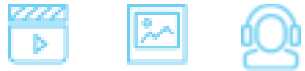
Utility Sensors | Employee Sensors | Location Data

MEDIA DATA

97% growth by 2017

82% unstructured

Media data comes from sources such as:



Video and Film | Images | Audio

HOW, AND WHY NOW?

The world is being reinvented in code. Java code.

HOW, AND WHY NOW?

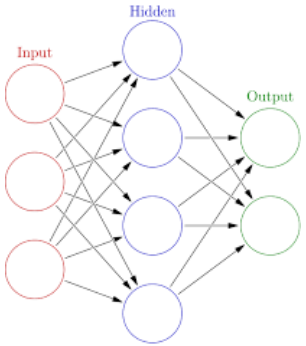
Computing is entering a new cognitive era.

What do you Think when you solve a problem?

HOW DO YOU THINK?

```
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);
                }
            }
        }
    }
}
```

CLASSIC NEURAL



BIG DATA QUANTUM

Quantum enabled, Neural Networked, Clustered, Containerized, Analysed, Cloudified, Toaster 4J

In shops soon

Trademarks, notes and disclaimers

- 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.