



## Why Node.js?

Michael Dawson  
IBM Community Lead for Node.js

Jesse Gorzinski  
jgorzins@us.ibm.com  
Powerpoint Stealer



## Atwood's Law: 2007



“Any application that can be written in JavaScript, will eventually be written in JavaScript.”

—Jeff Atwood, Cofounder of StackOverflow

## Agenda



- Why Node.js ?
- Node.js deep dive (maybe knee-deep)
- Positioning versus Java<sup>TM</sup>
- IBM involvement
- IBM i Integration
- A “Happy” Ending



© 2018 IBM Corporation

3

## Why Node.js – What is it?



- JavaScript != Java
- Node.js = **Server-side** JavaScript
  - Event-oriented
  - Non-blocking
  - Asynchronous



© 2018 IBM Corporation

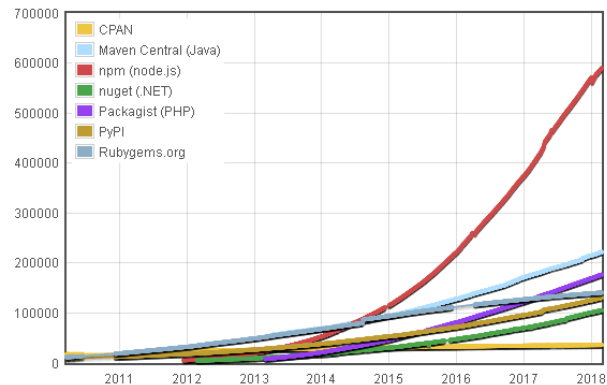
4

## Why Node.js ? – Ecosystem



- There is a module for that
  - Over 1 million modules
  - #1 on module counts
- #1 on Github (#projects)

### Module Counts



<http://www.modulecounts.com/>

© 2018 IBM Corporation

6

## Why Node.js ? – Ecosystem



- Most used runtime in IBM Cloud



### Infrastructure

#### Containers

Get started by creating a Kubernetes cluster, or manage your Docker images in the registry.



#### Containers in Kubernetes Clusters

Deploy secure, highly available apps in a

IBM

### Cloud Foundry Apps

Deploy your app without managing underlying infrastructure.



#### SDK for Node.js™

Develop, deploy, and scale server-side

IBM

© 2018 IBM Corporation

7

## Why Node.js ? – Productivity



- Reuse of “isomorphic” code components
- Availability of JavaScript talent
- Developer satisfaction

## Why Node.js ? – Productivity



- **Faster development less code**
- **PayPal** - <https://www.paypal-engineering.com/2013/11/22/node-is-at-paypal/>
  - Took 1/2 time with less people
  - 33% fewer lines of code
  - 40% fewer files
- **NextFlix** - <http://www.infoworld.com/article/2610110/javascript/paypal-and-netflix-cozy-up-to-node-js.html>
  - "We're used to working in JavaScript all day long. Having Node just makes it feel like a very natural extension of our work environment,"

## Who's Using in Production?



J.P.Morgan



© 2018 IBM Corporation



## Knee-Deep Dive



© 2018 IBM Corporation

10

## Node.js – Deep Dive – Key Characteristics



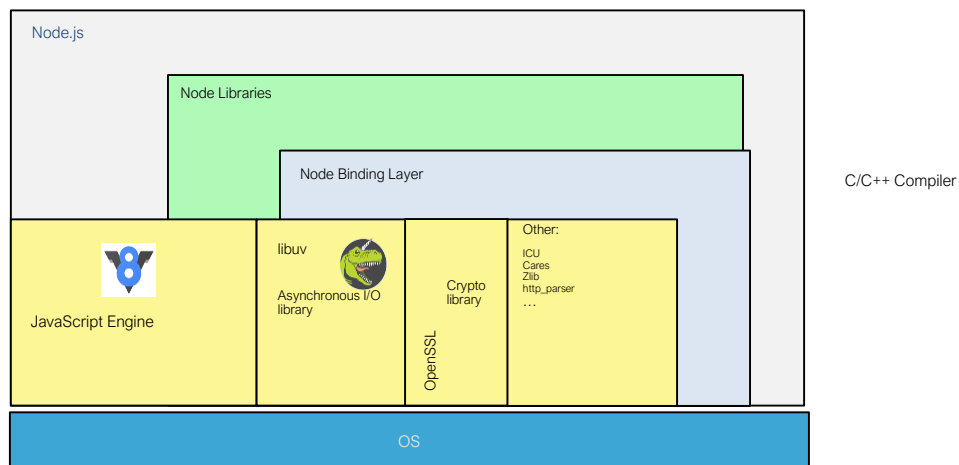
- Small (IBM i RPM)
  - Download **20 Mb**
- Fast startup
  - 60 ms
- Small footprint
  - 18 MB

<https://benchmarking.nodejs.org/>

© 2018 IBM Corporation

11

## Node.js – Deep Dive - Components



© 2018 IBM Corporation

16

# Node.js – Deep Dive - Programming Model



- Event Based

```
var http = require('http');

var server = http.createServer();
server.listen(8080);

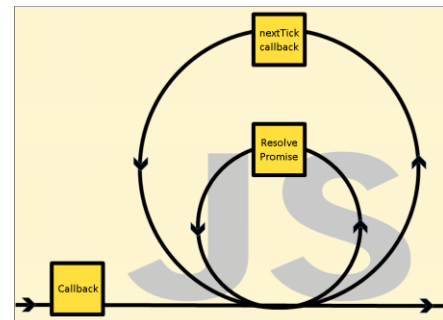
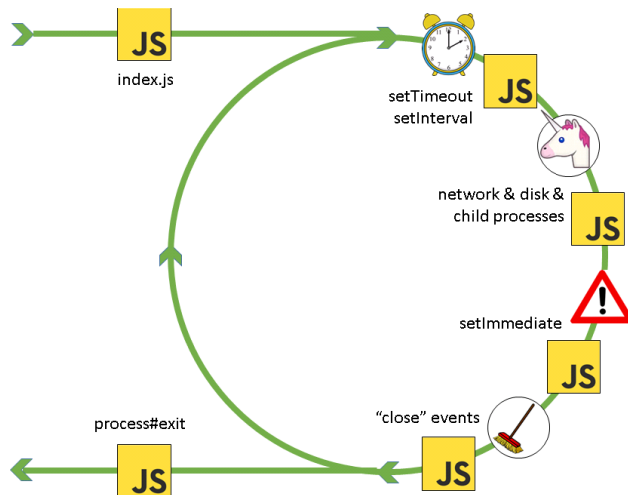
server.on('request', function(request, response) {
    response.writeHead(200, {"Content-Type": "text/plain"});
    response.write("Hello World!\n");
    response.end();
});

server.on('connection', function(socket) {});
server.on('close', function() {});
server.on('connect', function(socket) {});
server.on('upgrade', function(request, socket, head) {});
server.on('clientError', function(exception, socket) {});
```

13

© 2018 IBM Corporation

## Node.js – Deep Dive – Event Loop



19

© 2018 IBM Corporation

## Node.js – Deep Dive – NPM



- 1,000,000+ modules!!
- Two types of installs:
  - Global: use for command-line utilities
  - Local (default): use for application dependencies
- Fully encapsulates:
  - Dependency list within package.json file
  - Dependencies themselves within node\_modules/ directory
- Advantages:
  - Each application can operate independently
  - No global settings (extensions directory, classpaths, etc) to maintain
  - Portable

© 2018 IBM Corporation

15

## Node.js – Deep Dive – NPM



```

1. $ mkdir expressjs_app && cd expressjs_app
2. $ npm install express
3. express@4.12.0 node_modules/express
4.   └─ utils-merge@1.0.0
5.     └─ methods@1.1.1
6.       └─ fresh@0.2.4
7.         └─ merge-descriptors@0.0.2
8.           └─ cookie-signature@1.0.6
9.             └─ escape-html@1.0.1
10.              └─ range-parser@1.0.2
11.                └─ cookie@0.1.2
12.                  └─ finalhandler@0.3.3
13.                    └─ vary@1.0.0
14.                      └─ content-type@1.0.1
15.                        └─ parseurl@1.3.0
16.                          └─ content-disposition@0.5.0
17.                            └─ serve-static@1.9.1
18.                              └─ path-to-regexp@0.1.3
19.                                └─ depd@1.0.0
20.                                  └─ on-finished@2.2.0 (ee-first@1.1.0)
21.                                    └─ qs@2.3.3
22.                                      └─ debug@2.1.1 (ms@0.6.2)
23.                                        └─ proxy-addr@1.0.6 (forwarded@0.1.0, ipaddr.js@0.1.8)
24.                                          └─ etag@1.5.1 (crc@3.2.1)

```

© 2018 IBM Corporation



## Node.js – Deep Dive – NPM

holds meta-data about application



\$ npm init

Creates file package.json

```
{
  "name": "expressjs_app",
  "version": "0.0.0",
  "description": "",
  "main": "app.js",
  "dependencies": {
    "express": "^4.12.0"
  },
  "devDependencies": {},
  "author": "Aaron Bartell",
  "license": "ISC"
}
```

Installs these modules when  
npm install is run.

[docs.npmjs.com/cli/init](https://docs.npmjs.com/cli/init) - package.json creation  
[docs.npmjs.com/files/package.json](https://docs.npmjs.com/files/package.json) - Docs  
[browsenpm.org/package.json](https://browsenpm.org/package.json) - Easier docs

© 2018 IBM Corporation

## Debugging Node.js in a browser



More debugging options: [bit.ly/rs-debug-nodejs](https://bit.ly/rs-debug-nodejs)

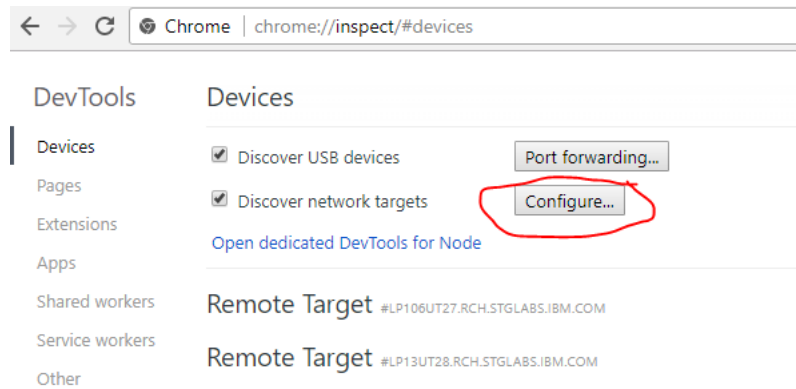
© 2018 IBM Corporation

## Debugging Node.js in a browser



Visit `chrome://inspect` in chrome

Configure your hostname and port as a "network target" (port 9229 is default port)



© 2018 IBM Corporation

## Debugging Node.js in a browser



- `$ node --inspect=0.0.0.0 hi.js`
- Debugger listening on port 9229.

Start node with `--inspect`

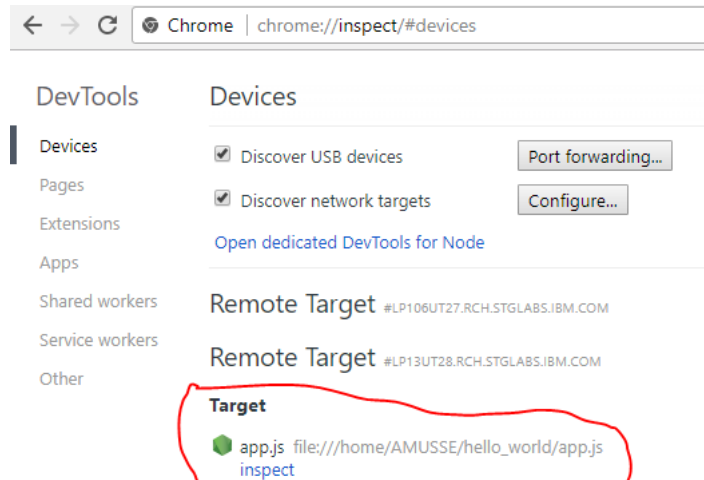
NOTE: IP Address '0.0.0.0' is important! Port will default to 9229 if not specified

© 2018 IBM Corporation

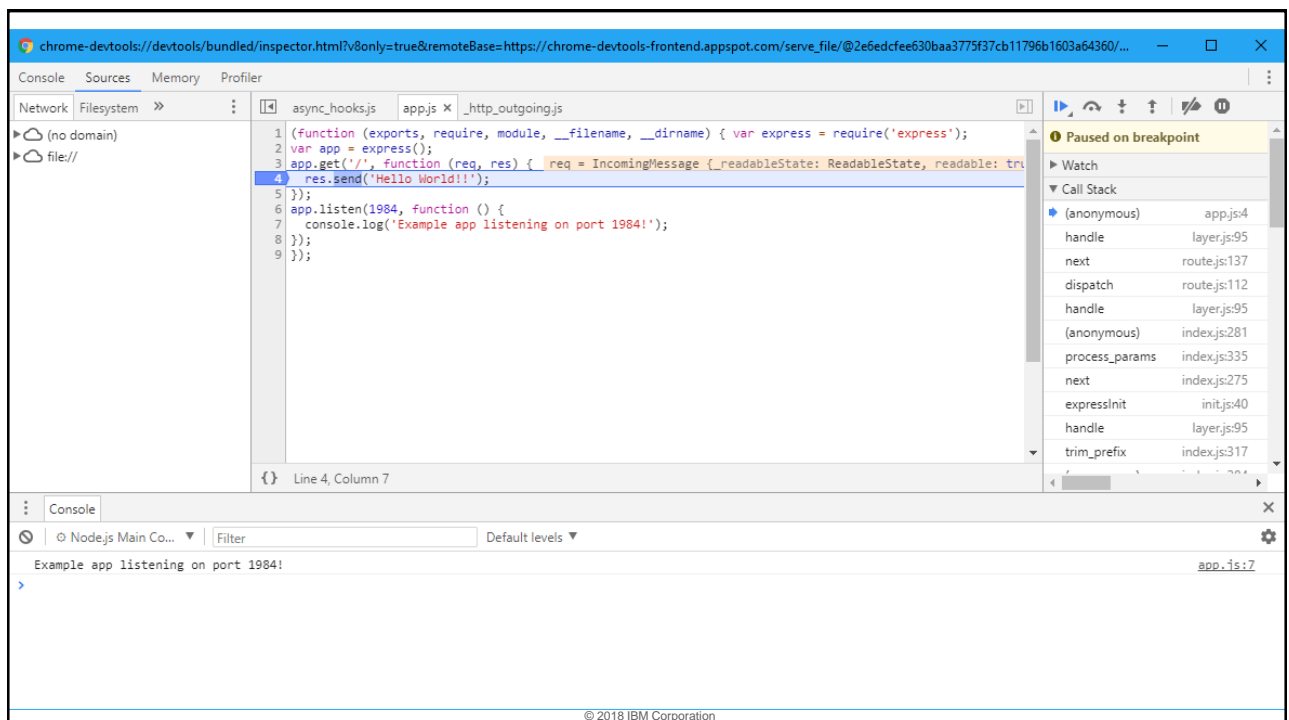
# Debugging Node.js in a browser



VOILA!! You will now see the remote target and can launch debug!

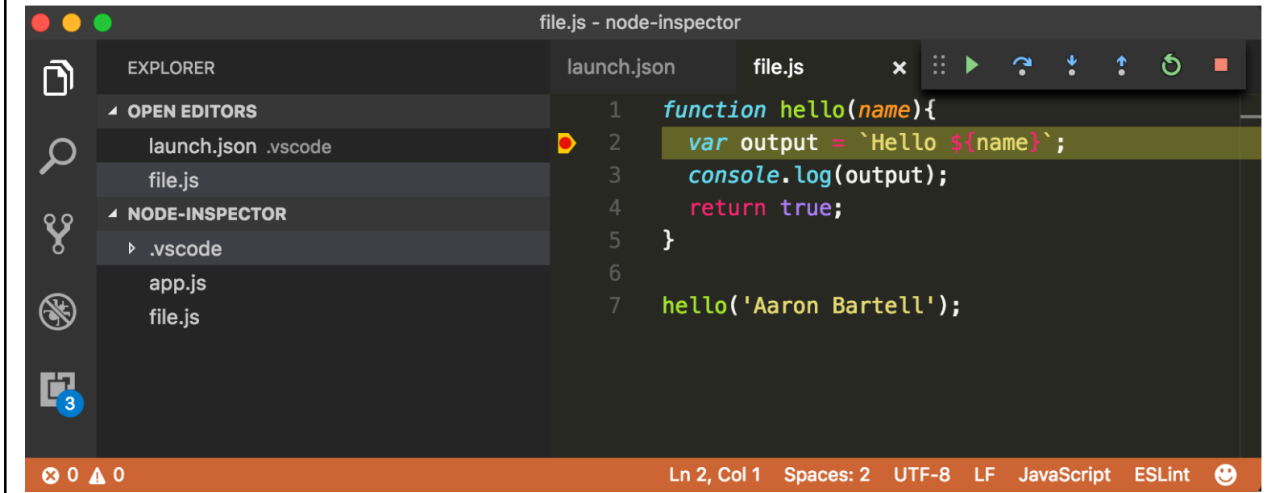


© 2018 IBM Corporation



© 2018 IBM Corporation

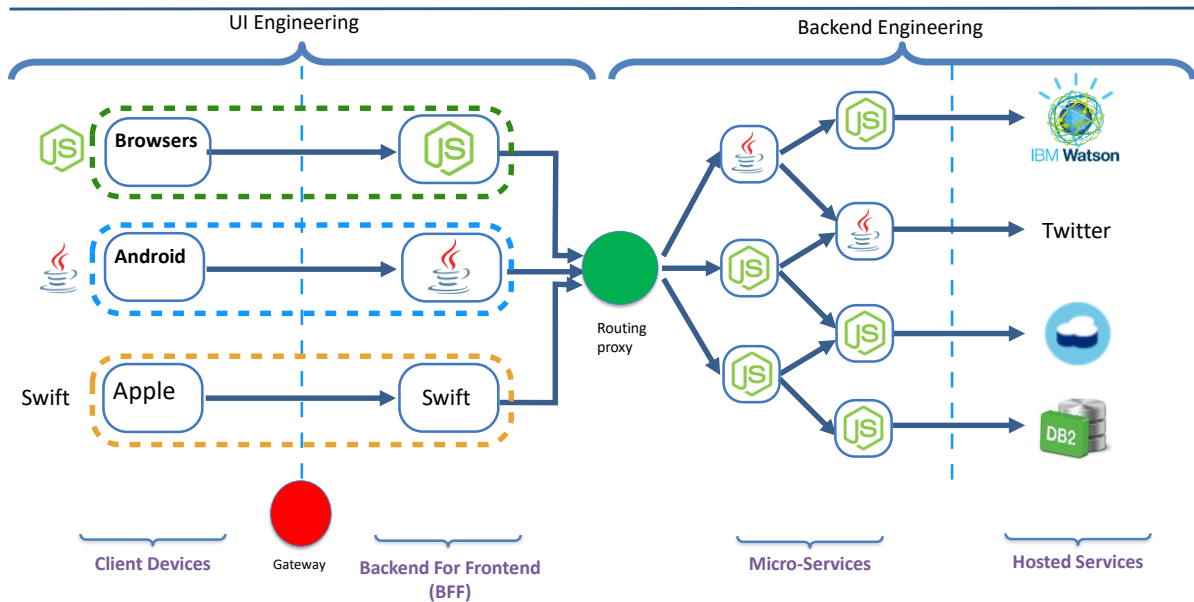
## Debugging Node.js in VSCode



Node.js and Java



## Node.js With Java



© 2018 IBM Corporation

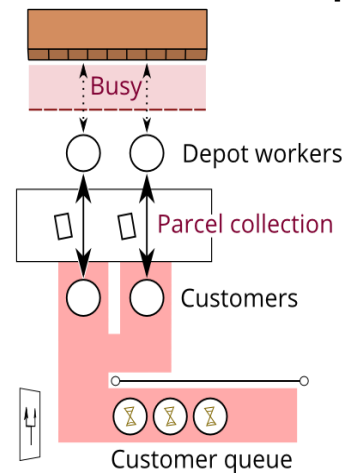
25

## Node.js With Java – Scaling with Java



- One thread (or process) per connection
  - Each thread waits on a response
  - Scalability determined by number of threads
- Each thread:
  - Consumes memory
  - Is relatively idle
- Concurrency determined by number of depot workers

### Parcel collection depot



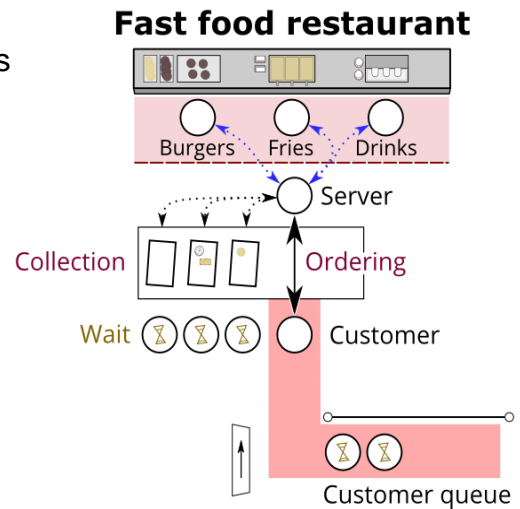
© 2018 IBM Corporation

26

## Node.js versus Java – Scaling with Node.js



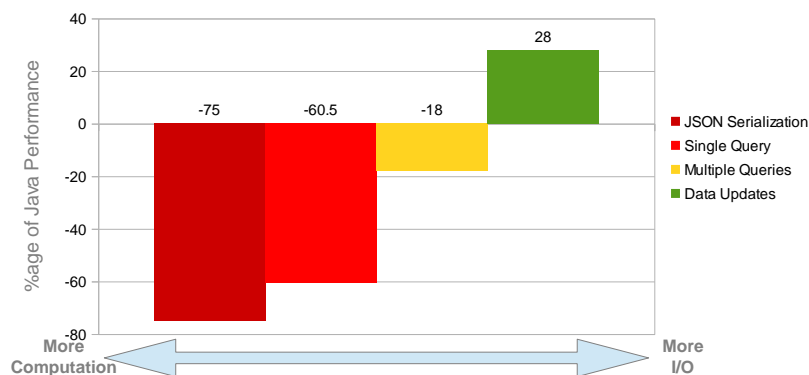
- One thread multiplexes for multiple requests
  - No waiting for a response
  - Handles return from I/O when notified
- Scalability determined by:
  - CPU Usage
  - “Back end” responsiveness
- Concurrency determined by how fast the food server can work



© 2018 IBM Corporation

27

## Node.js With Java– Tradeoffs



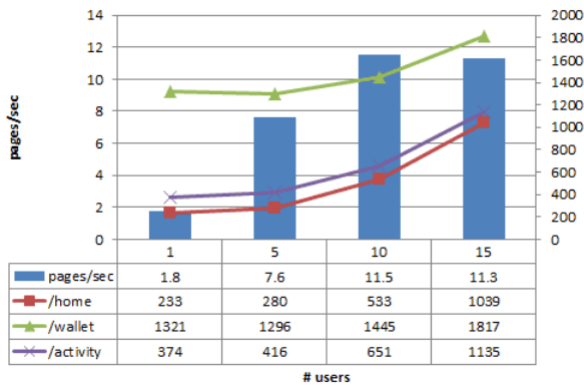
© 2018 IBM Corporation

26

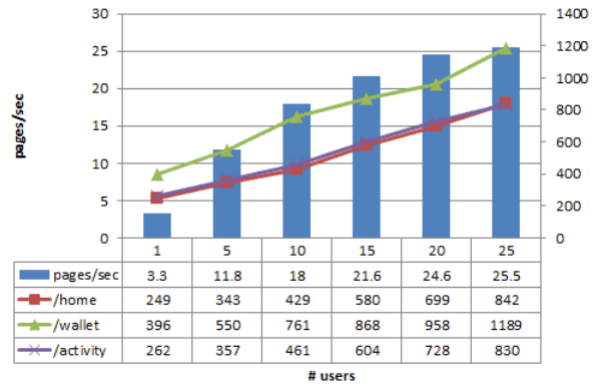
## Why Node.js ? Performance



### Java application



### Node.js application



<https://www.paypal-engineering.com/2013/11/22/node-js-at-paypal/>

© 2018 IBM Corporation

29

## Why Node.js ? - Performance



- Thousands of concurrent connections
- PayPal - <https://www.paypal-engineering.com/2013/11/22/node-js-at-paypal/>
  - **Double** number of requests/sec
  - Response times 35% **lower**
- Groupon — <http://www.nearform.com/nodecrunch/node-js-becoming-go-technology-enterprise/>
  - Reduced page load times by 50%

© 2018 IBM Corporation

13

## Node.js With Java – Choosing the Right Language



- Higher performance for I/O
- Easier async programming
- Fullstack/isomorphic development

© 2018 IBM Corporation

31

## Node.js versus Java – Choosing the Right Language



- Higher processing performance
- Type safety for calculations
- Rich processing frameworks

© 2018 IBM Corporation

32



## Node.js With Java– Choosing the Right Language



+

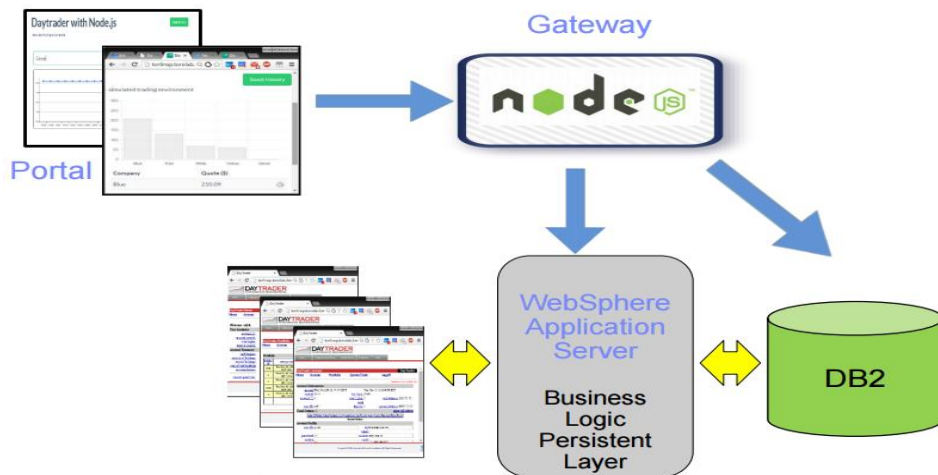


- Highly performant, scalable rich web applications
- Highly performant, reliable transaction processing
- Self-contained micro-service components

© 2018 IBM Corporation

33

## Node.js With Java– Hybrid applications



© 2018 IBM Corporation

30



## IBM involvement



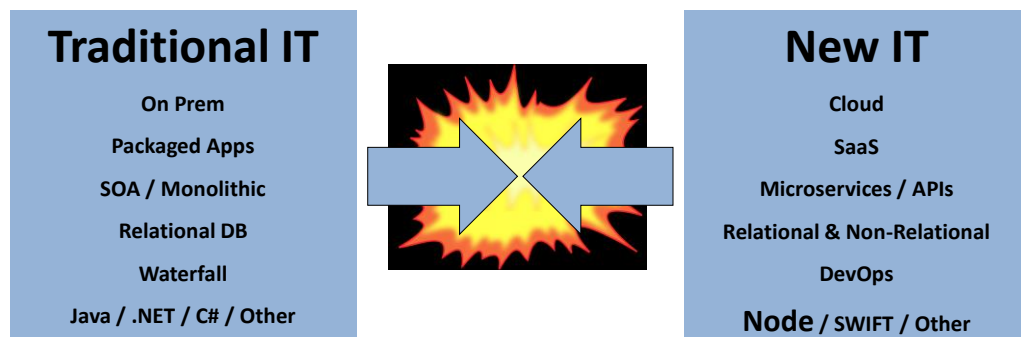
© 2018 IBM Corporation

35



## The Challenge for Every Existing Enterprise:

How to make the old work with the new?



© 2018 IBM Corporation

## IBM Node.js Strategy

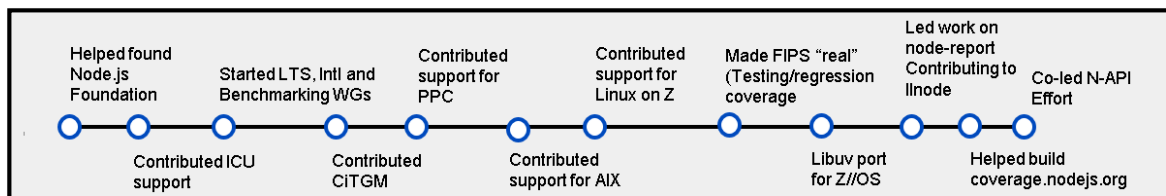


- Enterprise Ready Runtime
- Production Enablement
- Production Support

© 2018 IBM Corporation

37

## Enterprise Ready Runtime



© 2018 IBM Corporation

38

## N-API

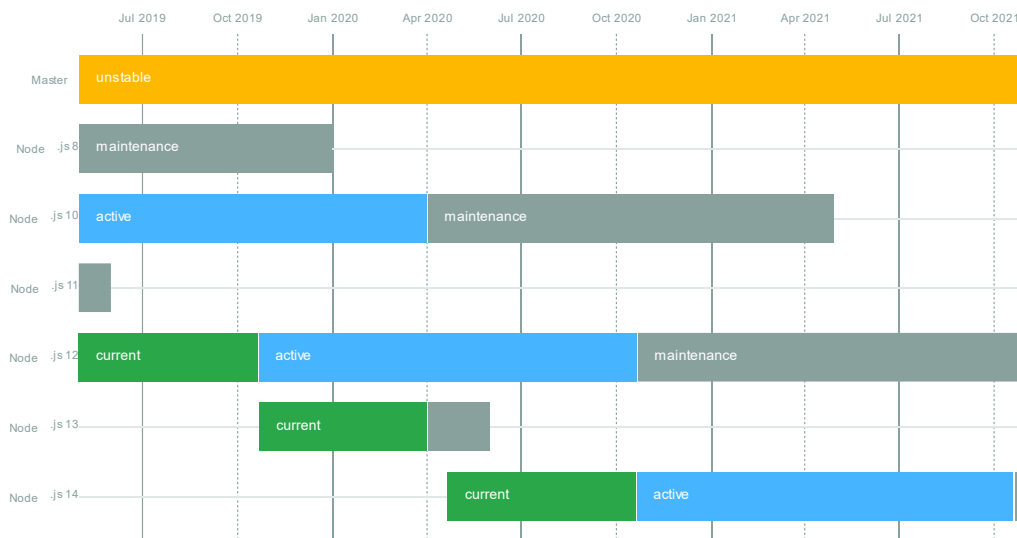


- Before N-API
  - Native modules coded to V8 API's
  - Modules needed to be recompiled for each new version of V8
  - Source changes sometimes also needed
  - Code less portable
  - Required non-trivial currency cost for module owners
- After N-API
  - Native modules coded to N-API
  - Need to be only built once for each platform
  - No code changes or recompiles needed to work with future versions of Node.js

© 2018 IBM Corporation

39

## Stable and Predictable Releases



<https://github.com/nodejs/Release>

© 2018 IBM Corporation

40

# Node.js IBM – Diagnostic Report



## NodeReport example - heap out of memory error

### NodeReport content:

- Event summary
- Node.js and OS versions
- JavaScript stack trace
- Native stack trace
- Heap and GC statistics
- Resource usage
- libuv handle summary
- Environment variables
- OS ulimit settings

Now part of Node.js core

```

NodeReport
Event: Allocation failed - JavaScript heap out of memory, location: "CALL_AND_RETRY_LAST"
Filename: NodeReport.20160815.125548.97.001.txt
Dump event time: 2016/08/15 12:55:48
Module load time: 2016/08/15 12:46:43

Node.js version: v6.3.0
(v8: 5.0.71.52, libuv: 1.9.1, ssl: 1.2.8, ares: 1.10.1-DEV)
OS version: Linux 3.15.0-33-generic #38-14.04.1-Ubuntu SMP Fri Nov 6 18:17:28 UTC 2015
Machine: 195p79815v x86_64
Process ID: 97

-----
JavaScript Stack Trace
-----
/home/vcap/app/app.js:57:15
Layer.handle ((as handle_request) (/home/vcap/app/node_modules/express/lib/router/layer.js:95:5)
next (/home/vcap/app/node_modules/express/lib/router/route.js:131:13)
Route.dispatch (/home/vcap/app/node_modules/express/lib/router/route.js:112:3)
Layer.handle ((as handle_request) (/home/vcap/app/node_modules/express/lib/router/layer.js:95:5)
/home/vcap/app/node_modules/express/lib/router/index.js:127:12
....

-----
Native Stack Trace
-----
0: [po=0x91a0a] v8::Utils::ReportApiFailure(char const*, char const*) [node]
1: [po=0x920a2] v8::internal::V8::FatalProcessOutOfMemory(char const*, bool) [node]
2: [po=0x019e08] v8::internal::Factory::NewUninitializedFixedArray(int) [node]
3: [po=0x00b47f] [node]
4: [po=0x0e102f3] v8::internal::Runtime_GrowArrayElements(int, v8::internal::Object**, v8::internal::Isolate*) [
5: [po=0x2751e3706338]

-----
JavaScript Heap and Garbage Collector
-----
Heap space name: new_space
Memory size: 1,048,576 bytes, committed memory: 52,416 bytes
Capacity: 1,031,936 bytes, used: 35,776 bytes, available: 996,160 bytes
Heap space name: old_space
Memory size: 51,298,304 bytes, committed memory: 50,673,800 bytes
Capacity: 50,117,880 bytes, used: 49,317,408 bytes, available: 800,472 bytes
  
```

© 2018 IBM Corporation

41

# AppMetrics - open-source Node.js monitoring



## What is it?

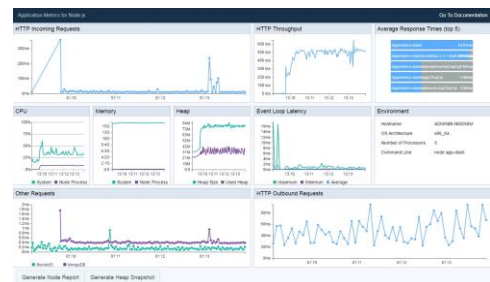
An open source module created by IBM for collecting application metrics to diagnose issues while developing your application. Metrics range from HTTP requests, event loop, memory usage, CPU usage, MongoDB connects, and more.

## Why use it?

Monitor and diagnose issues while developing your application. App Metrics then connects with IBM Cloud and API Connect for auto-scaling and more detailed availability monitoring

## How to get it?

GitHub at <https://github.com/RuntimeTools/appmetrics>. Users can view the dashboard by going to /appmetrics-dash or feeding it into their existing dashboard.



© 2018 IBM Corporation

42

## IBM Node.js Community Leadership



### Participation in Technical Steering Committee



Michael  
Dawson

© 2018 IBM Corporation

43

## Node.js Community Leadership @ IBM



Michael  
Dawson



Gireesh  
Punathil



Sam  
Roberts



Bethany  
Griggs



Ben  
Noordhuis



Christopher  
Hiller



Sam  
Ruby



Richard  
Lau



Steven  
Loomis



Yi-Hong  
Wang



Christian  
Clauss

- Four TSC members & eleven total Node.js core collaborators

© 2018 IBM Corporation

## Node.js IBM – V8 Community Involvement



- Deep expertise at V8
- Developed ports to IBM Platforms
- Contribution back to official V8 repositories:  
<https://github.com/v8/v8>
  - **PPC:** V8 4.3 and later have full functional PPC implementation
  - **s390:** V8 5.1 and later have full functional implementation
  - ~10-15 commits per week to V8 to maintain PPC/zlinux port
- Internal port for z/OS and IBM i

© 2018 IBM Corporation

45

## Freedom of Platform Choice



### • Community Binaries

- Linux on Z
- Linux on P
- AIX



Platform	Current	Previous
Windows Installer (msi)	12.0.0	10.0.0
Windows Binary (zip)	12.0.0	10.0.0
Mac OS X Installer (pkg)	12.0.0	10.0.0
Mac OS X Binary (tar.gz)	12.0.0	10.0.0
Linux Binary (tar.gz)	12.0.0	10.0.0
Source Code	12.0.0	10.0.0

Additional Platforms

Platform	Current	Previous
AIX Binary (tar.gz)	12.0.0	10.0.0
Solaris Binary (tar.gz)	12.0.0	10.0.0
Android Binary	12.0.0	10.0.0
Linux on Power Systems	12.0.0	10.0.0
Linux on System z	12.0.0	10.0.0
AIX on Power Systems	12.0.0	10.0.0

### • IBM Binaries

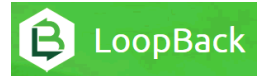
- IBM i
- z/OS



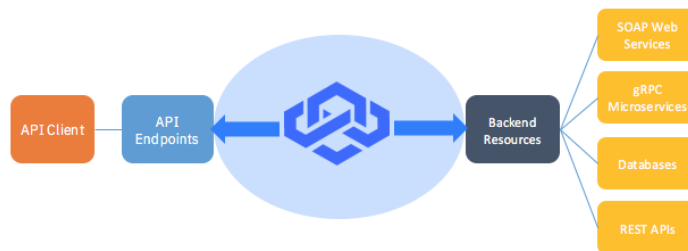
© 2018 IBM Corporation

46

## LoopBack – open-source Node.js framework



- Extends Express to accelerate API creation
- Create APIs quickly as microservices from existing services and databases
- Connects the dots between accepting API requests and interacting with backend
- Built for developers by developers (Reached 10k+ GitHub stars)



© 2018 IBM Corporation

47

## Production Support - IBM Support for Runtimes



- Years of experience
- Foundation -Community binaries
- Advanced – Key Modules from the Ecosystem (Express.js & Loopback)

<https://www.ibm.com/uk-en/marketplace/support-for-runtimes/faq>

© 2018 IBM Corporation

48



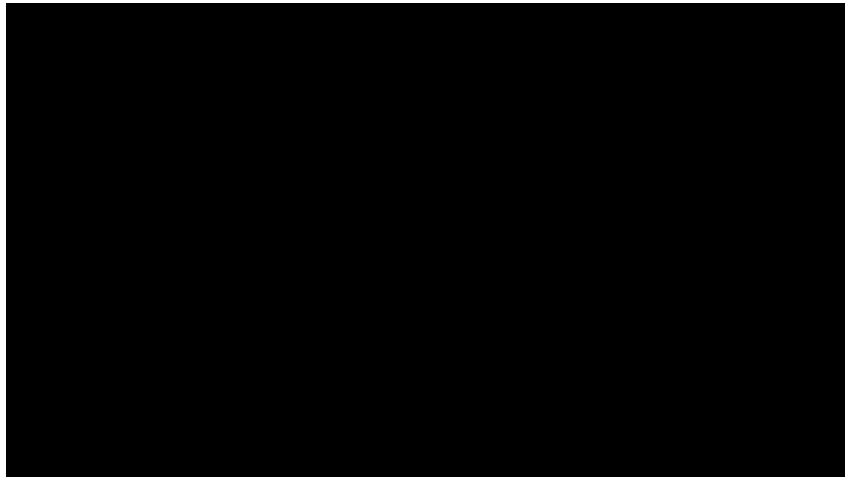
## IBM TSS Support for IBM i

---



- Git
  - Jenkins
  - rsync
  - Node.js
  - Apache Tomcat
  - WordPress
  - Python
- 
- For more resources, see <http://ibm.biz/ibmi-oss-support>

© 2018 IBM Corporation



© 2018 IBM Corporation

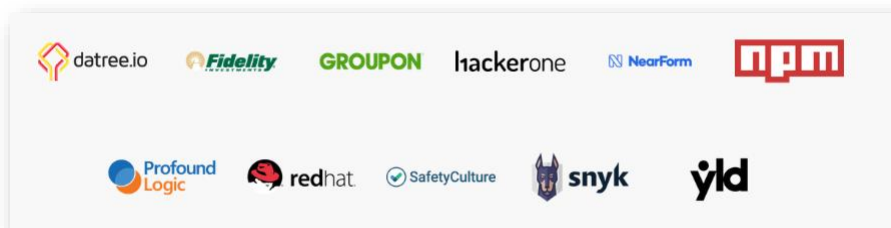
50

## Node.js foundation

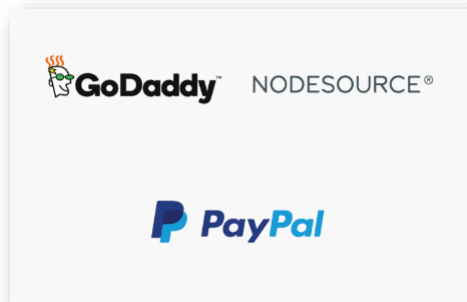


- <https://foundation.nodejs.org/>
- The Node.js Foundation's mission is to enable widespread adoption and help accelerate development of Node.js and other related modules through an open governance model that encourages participation, technical contribution, and a framework for long term stewardship by an ecosystem invested in Node.js' success.

## Node.js foundation – Silver Members



## Node.js foundation – Gold Members



© 2018 IBM Corporation

53

## Node.js foundation – Platinum Members



© 2018 IBM Corporation

54

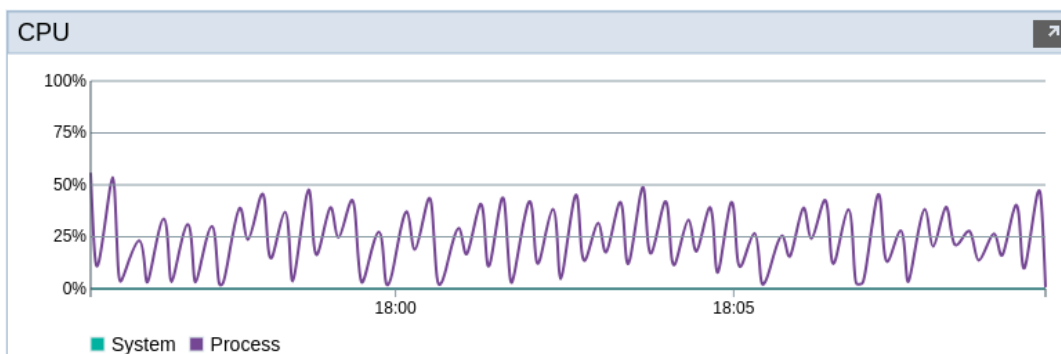


# AppMetrics-dash

© 2018 IBM Corporation

55

## AppMetrics



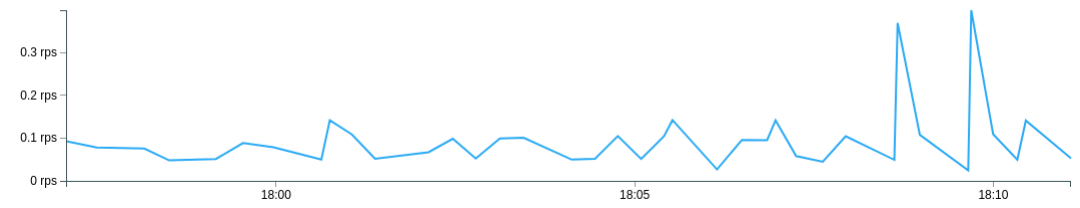
© 2018 IBM Corporation

56

## AppMetrics



HTTP Throughput



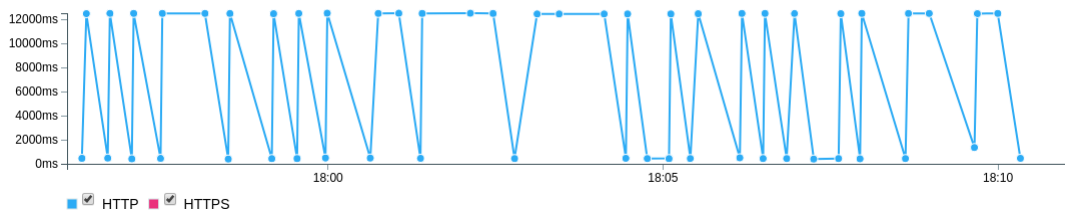
© 2018 IBM Corporation

57

## AppMetrics



HTTP Incoming Requests



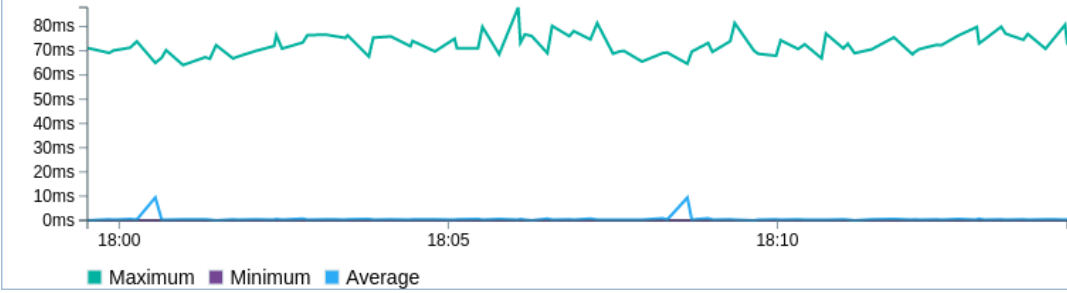
© 2018 IBM Corporation

58

## AppMetrics



## Loop Times



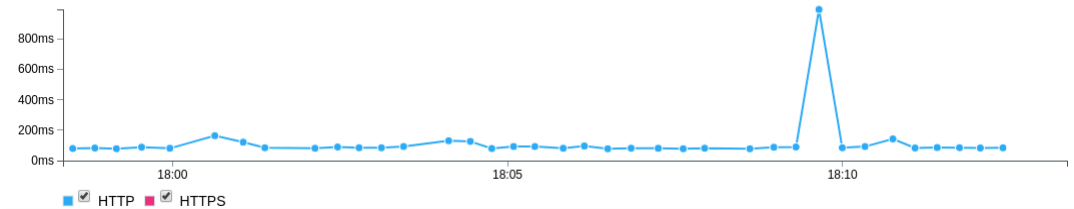
© 2018 IBM Corporation

59

## AppMetrics



## HTTP Outbound Requests



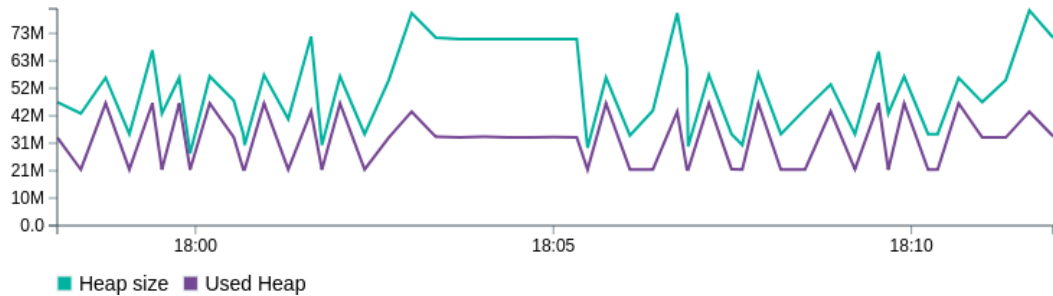
© 2018 IBM Corporation

60

## AppMetrics



## Heap



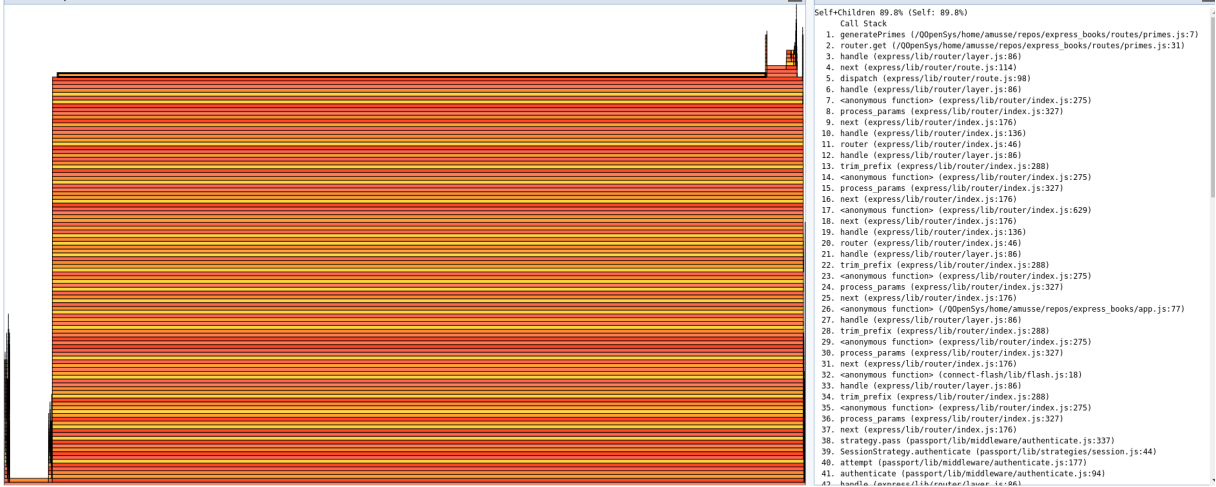
© 2018 IBM Corporation

61

## AppMetrics



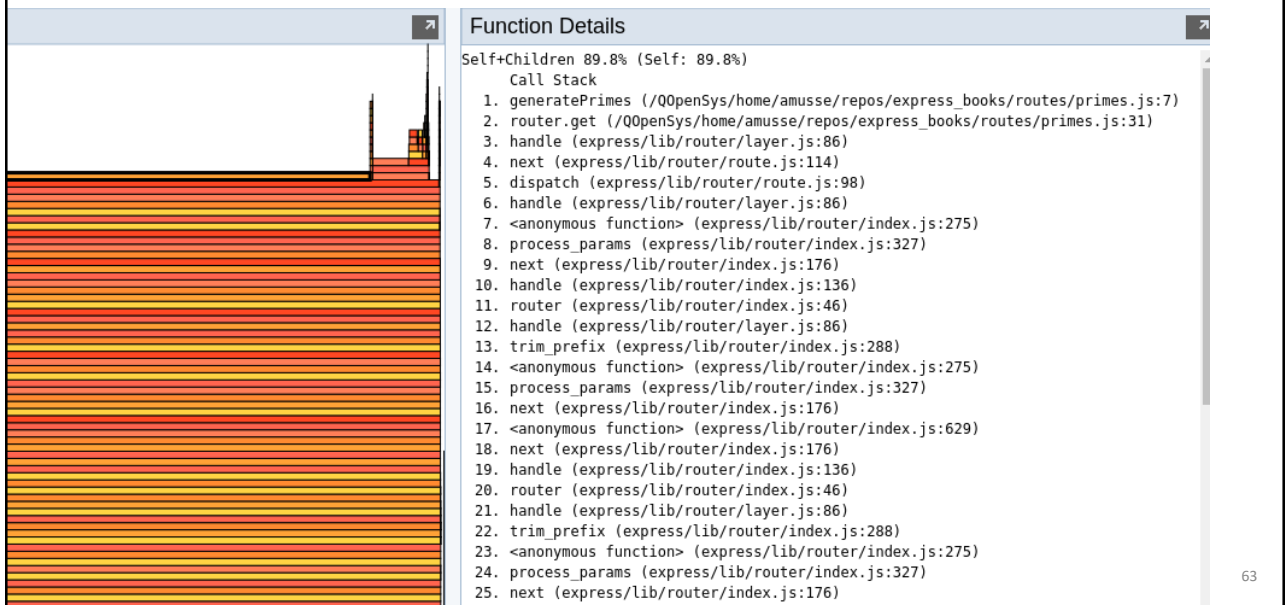
## Flame Graph



© 2018 IBM Corporation

62

## AppMetrics



IBM i Integration





## Node.js connectivity libraries - Database



- idb-connector v1.x
  - Direct API access to Db2
  - Standard JavaScript (callback-based) conventions
- idb-pconnector v1.x
  - Built around idb-connector
  - Promises-based API set (aligns with new language standards)
  - Includes connection pooling (avoids many common pitfalls)
- odbc
  - Uses standard ODBC interfaces
  - Provides both traditional and promise-based usage
  - Includes connection pooling
  - <https://github.com/IBM/ibmi-oss-examples/blob/master/odbc/odbc.md>
  - <https://www.ibmssystemsmag.com/Power-Systems/08/2019/ODBC-Driver-for-IBM-i>



© 2018 IBM Corporation

## Node.js connectivity libraries - Data



- itoolkit
  - Allows Node.js to integrate with RPG, Db2, CL, COBOL, etc.
  - **\*\*NEW in v1.x\*\*** Can use SSH or ODBC to talk to the system
- Sequelize Dialect
  - Db2 via the popular Node.js ORM
- Loopback Connector
  - Connector for connecting to Db2 on IBM i when using LoopBack
  - LoopBack is a framework for creating REST APIs for your data
  - Your data is defined in terms of models, which are objects that mirror database tables



© 2018 IBM Corporation

## odbc Example



```
$ node app.js
[ { STATE: 'TX' },
  { STATE: 'NY' },
  { STATE: 'VT' },
  { STATE: 'GA' },
  { STATE: 'NY' },
  { STATE: 'CO' },
  { STATE: 'MN' },
  { STATE: 'CA' },
  { STATE: 'WY' },
  { STATE: 'TX' },
  { STATE: 'NY' },
  { STATE: 'MN' },
  statement: 'SELECT STATE FROM QIWS.QCUSTCDT',
  parameters: [],
  return: undefined,
  count: -1,
  columns: [ { name: 'STATE', dataType: 1 } ] ]
$
```

```
$ npm init -y
$ npm install odbc
```

### app.js

```
const odbc = require('odbc');
const sql = 'SELECT STATE FROM QIWS.QCUSTCDT'

const connection =
odbc.connect('DSN=IBMISSYSTEM');

connection.query(sql, (error1, result) => {
  console.log(result);
  connection.close((error2) => {
    console.log('connection closed');
  });
});
```

© 2018 IBM Corporation



## Customer Stories




© 2018 IBM Corporation


Power Systems
Hardware
OS
Software
Resources

## IBM i Customer Stories

For over thirty years, IBM i has supported thousands of customers worldwide and continues to be the backbone of their infrastructure into the future. Read why customers are continuing to choose IBM i as their platform of choice.




### See innovation at work in the IBM i community

**PHD, Inc.**



Reducing Reporting Time and Increasing Performance with IBM i on POWER9

[Read the customer story →](#)

**Hatco Corporation**



Modernizing with Git and Open Source on IBM i

[Read the customer story →](#)

**Stonetales Properties**


Upgrading and Centralizing on the Cloud with IBM i

[Read the customer story →](#)

**Vision Banco**


Increasing Revenue and Improving Customer Service with IBM i and AI

[Read the customer story →](#)

Let's talk

## IBM i Microsite Customer Stories using Node.js



- <http://ibm.biz/ibmistories>
  - FormaServ
  - Krengeltech
  - Mutual Distributing Company
  - Geodis
  - RPC Superfos
  - HT Bendix A/S
  - Kuehne + Nagel S.à.r.l.

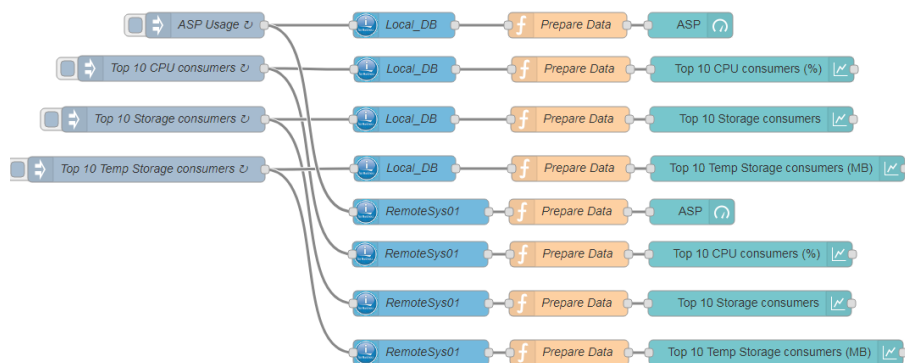


# CRAS



© 2018 IBM Corporation

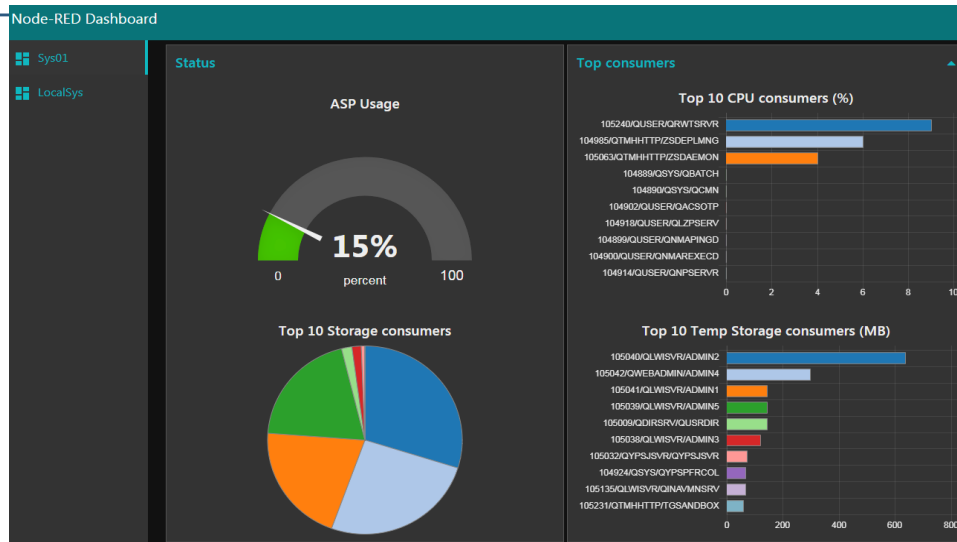
## Node-RED example



© 2018 IBM Corporation

72

## Node-RED example



© 2018 IBM Corporation

73

## CRAS Woodgroup



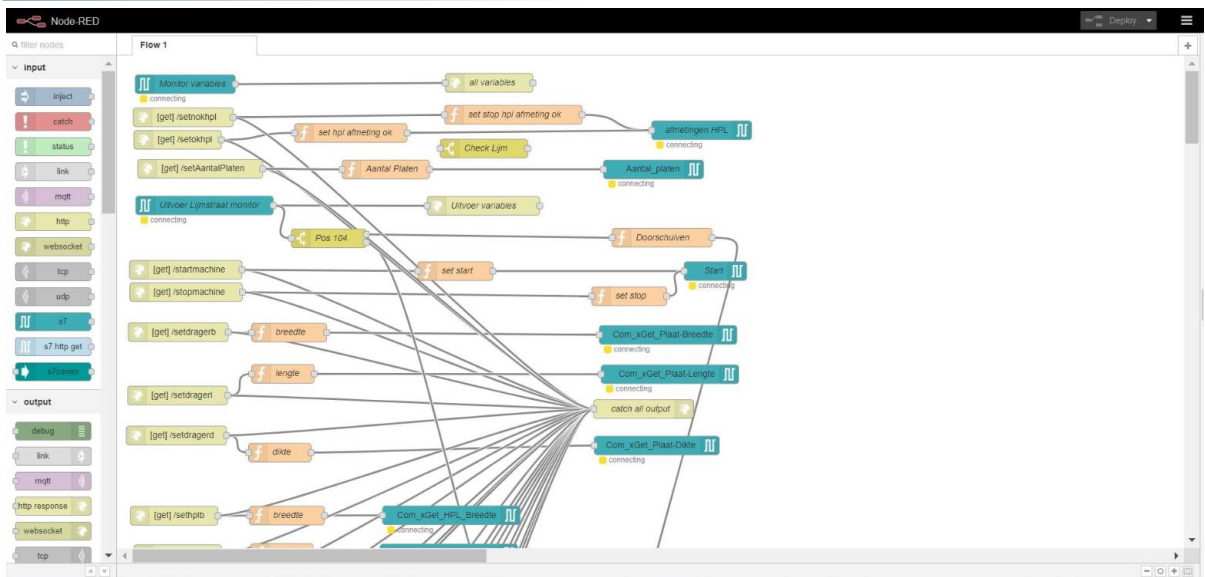
- **Who**
  - Cras creates wood products to order, adjusting humidity, temperature and other factors in its plants to meet precise customer requirements.
- **The scenario**
  - Evaluating implementations for transformation effort
  - Trouble linking all the PLCs (problem logic controllers)
- **The Solution**
  - With Node-RED, our BP was able to link all the PLCs directly and easily
- **The Result**
  - IBM i undoubtedly the right place for production

<https://www.ibm.com/case-studies/cras-systems-open-source>

**CRAS**  
WOODSHOPS

© 2018 IBM Corporation

## Node-RED in production



© 2018 IBM Corporation



Stonetales



© 2018 IBM Corporation

## Stonetales



- **Who**
  - a real estate company involved in housing marketing, cohousing and sales of properties.
- **The scenario**
  - acquired a competitor that was running on Microsoft Windows and x86.
- **The Solution**
  - Node.js on IBM i (new software development providing solutions to improve customer service and streamline processes).
- **The result**
  - Quickly decided to discard the x86 workloads
  - Consolidated to IBM i on POWER hardware.

<https://cms.ibm.com/case-studies/stonetales-properties-power-upgrade>

© 2018 IBM Corporation



# A “Happy” Ending



© 2018 IBM Corporation

78



# A “Hapi” Ending

© 2018 IBM Corporation

79



80



## Walmart creates a framework!



- Express.js appeared in 2009.
- Walmart saw Express.js insufficient for very large projects, but saw the huge potential in Node.js.
- Willing to invest millions of dollars in a new framework.
- <https://garage.socialisten.at/2016/12/enterprise-level-backend-framework-from-walmart>



© 2018 IBM Corporation

81

## Black Friday 2013



- Full deployment for all mobile shopping!!
- The hardware?
  - 10 CPU cores
  - 28 GB memory



**Eran Hammer**

@eranhammer

Follow



100% of Walmart US mobile traffic is flowing through @nodejs using @hapijs and the servers are bored out of their mind. #nodebf

11:55 PM - 28 Nov 2013

© 2018 IBM Corporation

82

## Q&amp;A



# The END!

© 2018 IBM Corporation

83

## Special notices



This document was developed for IBM offerings in the United States as of the date of publication. IBM may not make these offerings available in other countries, and the information is subject to change without notice. Consult your local IBM business contact for information on the IBM offerings available in your area.

Information in this document concerning non-IBM products was obtained from the suppliers of these products or other public sources. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquiries, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

The information contained in this document has not been submitted to any formal IBM test and is provided "AS IS" with no warranties or guarantees either expressed or implied.

All examples cited or described in this document are presented as illustrations of the manner in which some IBM products can be used and the results that may be achieved. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions.

IBM Global Financing offerings are provided through IBM Credit Corporation in the United States and other IBM subsidiaries and divisions worldwide to qualified commercial and government clients. Rates are based on a client's credit rating, financing terms, offering type, equipment type and options, and may vary by country. Other restrictions may apply. Rates and offerings are subject to change, extension or withdrawal without notice.

IBM is not responsible for printing errors in this document that result in pricing or information inaccuracies.

All prices shown are IBM's United States suggested list prices and are subject to change without notice; reseller prices may vary.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this document may have been estimated through extrapolation. Users of this document should verify the applicable data for their specific environment.

Revised September 26, 2006

© 2018 IBM Corporation

84

## Special notices (cont.)



IBM, the IBM logo, ibm.com AIX, AIX (logo), AIX 5L, AIX 6 (logo), AS/400, BladeCenter, Blue Gene, ClusterProven, Db2, ESCON, i5/OS, i5/OS (logo), IBM Business Partner (logo), IntelliStation, LoadLeveler, Lotus, Lotus Notes, Notes, Operating System/400, OS/400, PartnerLink, PartnerWorld, PowerPC, pSeries, Rational, RISC System/6000, RS/6000, THINK, Tivoli, Tivoli (logo), Tivoli Management Environment, WebSphere, xSeries, z/OS, zSeries, Active Memory, Balanced Warehouse, CacheFlow, Cool Blue, IBM Systems Director VMControl, pureScale, TurboCore, Chiphopper, Cloudscape, Db2 Universal Database, DS4000, DS6000, DS8000, EnergyScale, Enterprise Workload Manager, General Parallel File System, GPFS, HACMP, HACMP/6000, HASM, IBM Systems Director Active Energy Manager, iSeries, Micro-Partitioning, POWER, PowerExecutive, PowerVM, PowerVM (logo), PowerHA, Power Architecture, Power Everywhere, Power Family, POWERHypervisor, Power Systems, Power Systems (logo), Power Systems Software, Power Systems Software (logo), POWER2, POWER3, POWER4, POWER4+, POWER5, POWER5+, POWER6, POWER6+, POWER7, System i, System p, System p5, System Storage, System z, TME 10, Workload Partitions Manager and X-Architecture 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 a trademark 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 full list of U.S. trademarks owned by IBM may be found at: <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Altivec is a trademark of Freescale Semiconductor, Inc.

AMD Opteron is a trademark of Advanced Micro Devices, Inc.

InfiniBand, InfiniBand Trade Association and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

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

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries or both.

Microsoft, Windows and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries or both.

NetBench is a registered trademark of Ziff Davis Media in the United States, other countries or both.

SPECint, SPECfp, SPECjbb, SPECweb, SPECjAppServer, SPEC OMP, SPECviewperf, SPECcapc, SPECchpc, SPECjvm, SPECmail, SPECimap and SPECfs are trademarks of the Standard Performance Evaluation Corp (SPEC).

The Power Architecture and Power.org wordmarks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

TPC-C and TPC-H are trademarks of the Transaction Performance Processing Council (TPPC).

UNIX is a registered trademark of The Open Group in the United States, other countries or both.

•Node.js is an official trademark of Joyent. IBM SDK for Node.js is not formally related to or endorsed by the official Joyent Node.js open source or commercial project..

•TWITTER, TWEET, RETWEET and the Twitter logo are trademarks of Twitter, Inc. or its affiliates.\*

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

Revised December 2, 2010