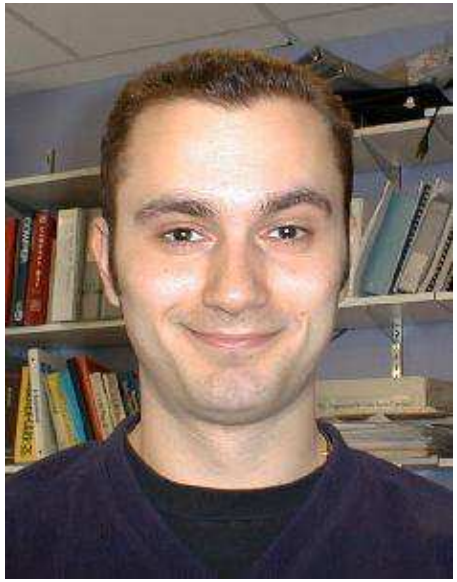
Marcel Mitran - Lead Architect, IBM Java on Z
November 19th, 2013

The Evolution of the Hardware/Software Interface

Who am I, and What Do I Do



School

- McGill B'Eng ECE class of '97
- McGill M'Eng CIM class of '01 (computer vision)

Joined IBM Sept 10th, 2001

- IBM Master Inventor and Senior Technical Staff Member (STSM)
- Lead Architect IBM Java on System z
  - Corporate-wide responsibility for developing IBM's JDK on System z
- **World-wide technical leader for compilers development for System z**
  - **Hardware/Software interface**

Work closely with:

- JVM (Ottawa Lab), C/C++, COBOL (Silicon Valley Lab), XML compiler teams
- Hardware designers (Power, System z, Intel, AMD…)
- O/S developers (AIX, z/OS, Linux…)
- Middleware developers (Websphere, DB2, Tivoli)
- Research (Tokyo Research Lab, Watson Research Lab)

Did you say Mainframe!?!?

System z run applications that **run my life**

- Used by 95% of the Fortune 500 Companies
- 80% of corporate data resides or originates on mainframes
- Runs everything from your class registration to airplane reservations
- 2/3 of business transactions for US retail banks run directly on mainframes

System z is **secure**

- Highest level of industry security rating, EAL5, awarded to the mainframe

System z is **dependable** and **available**

- Less than 5 minutes down time per year
- Mean time to failure is 40 years

System z is **virtualized**

- Create a new Linux image in as little as 90 seconds

System z is **expandable**, **adaptable** and **flexible**

- Add capacity and software updates without a reboot
- Respond automatically to spikes in workload demands
- Align processing priorities with business priorities

The Economist

The return of the mainframe: Back in fashion

Jan 14th, 2010

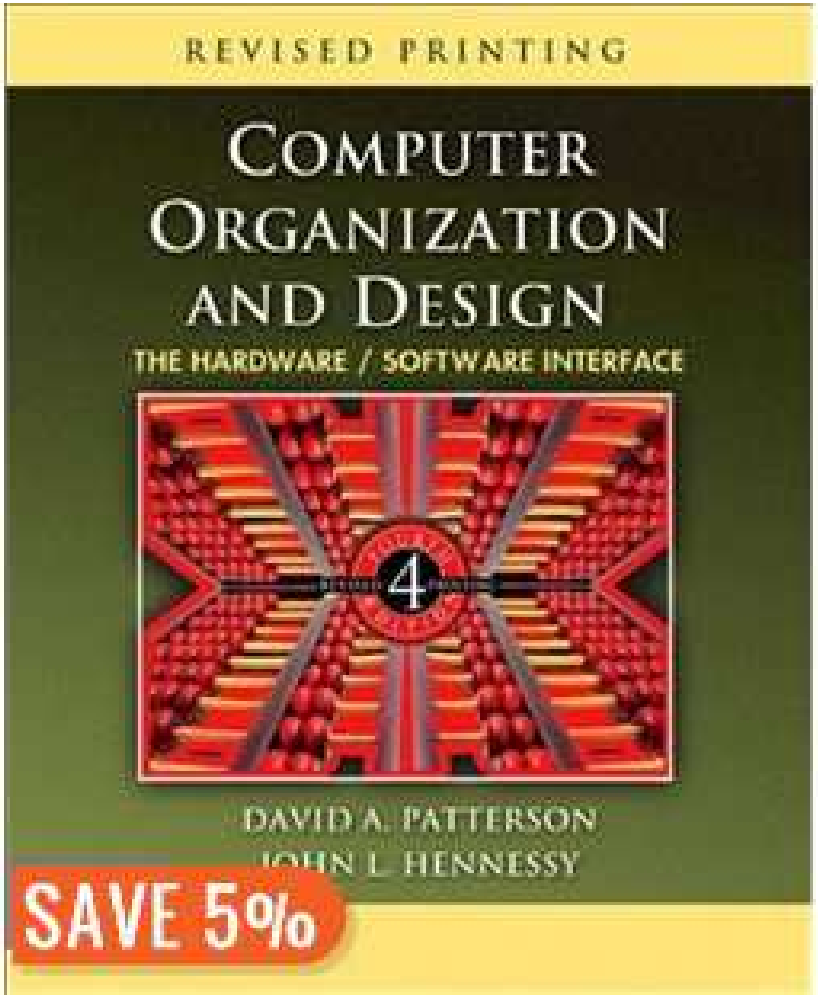
*Toni Sacconaghi of Bernstein Research estimates that **40%** of IBM's profits are mainframe-related.*

Classical View of the Hardware/Software Interface

- Classical view of the h/w + s/w interface
  (at least with my 1st Edition copy of the text)
  - Data representation
  - Instruction architecture
  - Pipelining
  - Memory hierarchy

- Little to no reference to
  - Multi-core/SMT
  - Dynamic compilation/managed runtimes
  - I/O attached accelerators
  - domain-specific devices/languages
  - distributed computing
  - parallel programming models
  - etc

- Up-until ~5 years ago, largely ok..



zEnterprise EC12 Hardware – Available since Sept 2012

Continued aggressive investment in H/W + S/W co-design

Hardware Transaction Memory (HTM)
- Better concurrency for multi-threaded/parallel applications
- Fine-grained concurrency

Run-time Instrumentation (RI)
- Real-time feedback on dynamic program characteristics
- Enables increased optimization by Java

2GB page frames
- Improved performance targeting 64-bit heaps

Page-able 1MB large pages using Flash Express
- Better versatility of managing memory

Shared-Memory-Communication
- RDMA over Converged Ethernet

zEnterprise Data Compression accelerator
- FPGA-based acceleration of gzip

Misc new instructions
- Software hints directives
- Traps
- Decimal conversion

System Specs
- **120-way** machine
- 3TB of real storage
- IBM zAware – autonomic monitoring
- Common Criteria Evaluation Assurance Level 5+ (un-matched)
- IBM DB2 Analytics Accelerator

zEnterprise EC12 Hardware – Available since Sept 2012

Continued aggressive investment in H/W + S/W co-design

Hardware Transaction Memory (HTM)
- Better concurrency for multi-threaded/parallel applications
- Fine-grained concurrency

Run-time Instrumentation (RI)
- Real-time feedback on dynamic program characteristics
- Enables increased optimization by Java

2GB page frames
- Improved performance targeting 64-bit heaps

Page-able 1MB large pages using Flash Express
- Better versatility of managing memory

Shared-Memory-Communication
- RDMA over Converged Ethernet

zEnterprise Data Compression accelerator
- FPGA-based acceleration of gzip

Misc new instructions
- Software hints directives
- Traps
- Decimal conversion

System Specs
- **120-way** machine
- 3TB of real storage
- IBM zAware – autonomic monitoring
- Common Criteria Evaluation Assurance Level 5+ (un-matched)
- IBM DB2 Analytics Accelerator

zEnterprise EC12 Hardware – Available since Sept 2012

Continued aggressive investment in H/W + S/W co-design

Hardware Transaction Memory (HTM)
- Better concurrency for multi-threaded/parallel applications
- Fine-grained concurrency

Run-time Instrumentation (RI)
- Real-time feedback on dynamic program characteristics
- Enables increased optimization by Java

2GB page frames
- Improved performance targeting 64-bit heaps

Page-able 1MB large pages using Flash Express
- Better versatility of managing memory

Shared-Memory-Communication
- RDMA over Converged Ethernet

zEnterprise Data Compression accelerator
- FPGA-based acceleration of gzip

Misc new instructions
- Software hints directives
- Traps
- Decimal conversion

System Specs
- **120-way** machine
- 3TB of real storage
- IBM zAware – autonomic monitoring
- Common Criteria Evaluation Assurance Level 5+ (un-matched)
- IBM DB2 Analytics Accelerator

Inflection Point in Computing?

Topics discussed in a typical week at work

- The end of single-thread performance
- The evolution of HPC into analytics and optimization in the enterprise
- “BigData”
- The “Cloud”
- Systems of engagement, scripting…

Scripting Ecosystem

Evolution of the Enterprise Computing Ecosystem

OLTP

Industry has spent the last decade focusing on **OnLine Transaction Processing** (OLTP)

- Enabling/optimizing data persistency and serving
- Internet of Things
- Trillions of transactions/day
- Massive amounts of data (structure/unstructured)

WebSphere
CICS/IMS
DB2

Evolution of the Enterprise Computing Ecosystem

BAO

Business Intelligence, Analytics and Optimization

- A clear need to understanding how to interpret/optimize/predict the data
  eg. fraud detection, customer relations management, low-latency trading etc

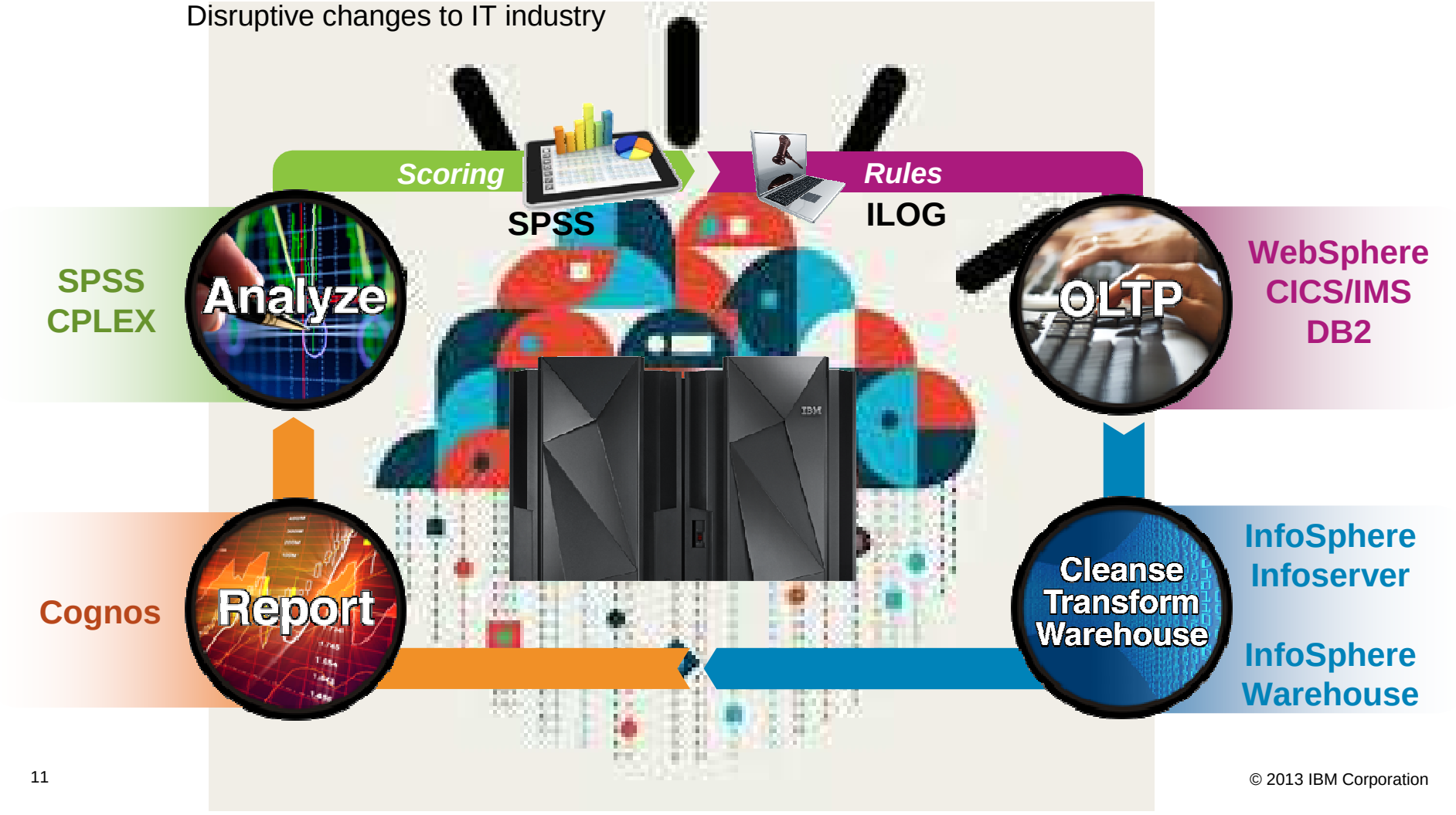
Evolution of the Enterprise Computing Ecosystem

Cloud

Economies of scale for computational infrastructure through 3rd party hosting

Driving down cost

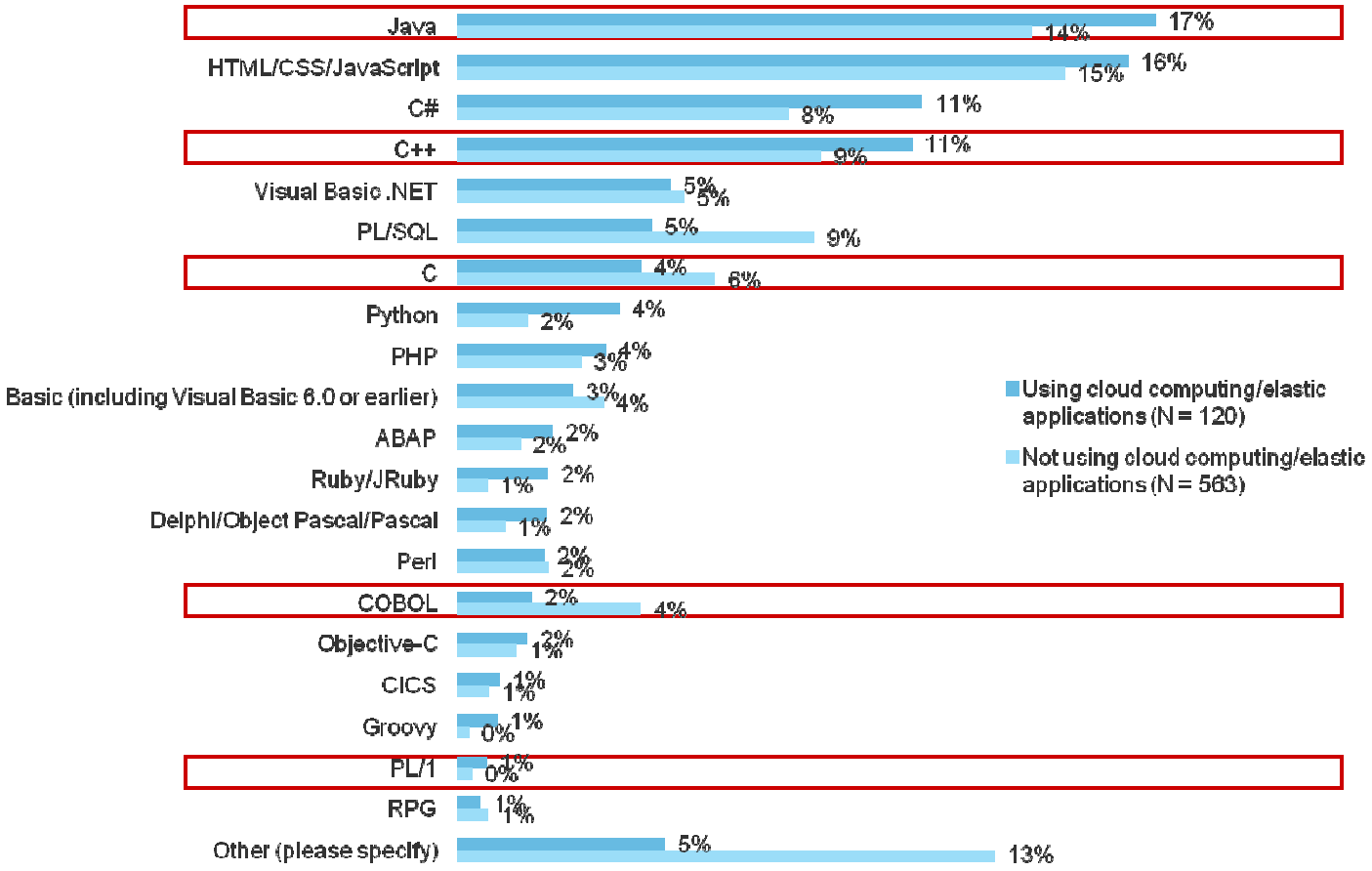
Disruptive changes to IT industry



Significant presence for traditional enterprise languages

Java leads developer language choice

"How do you allocate the time you spend writing code across the following programming languages?" (Enter a percentage for each)



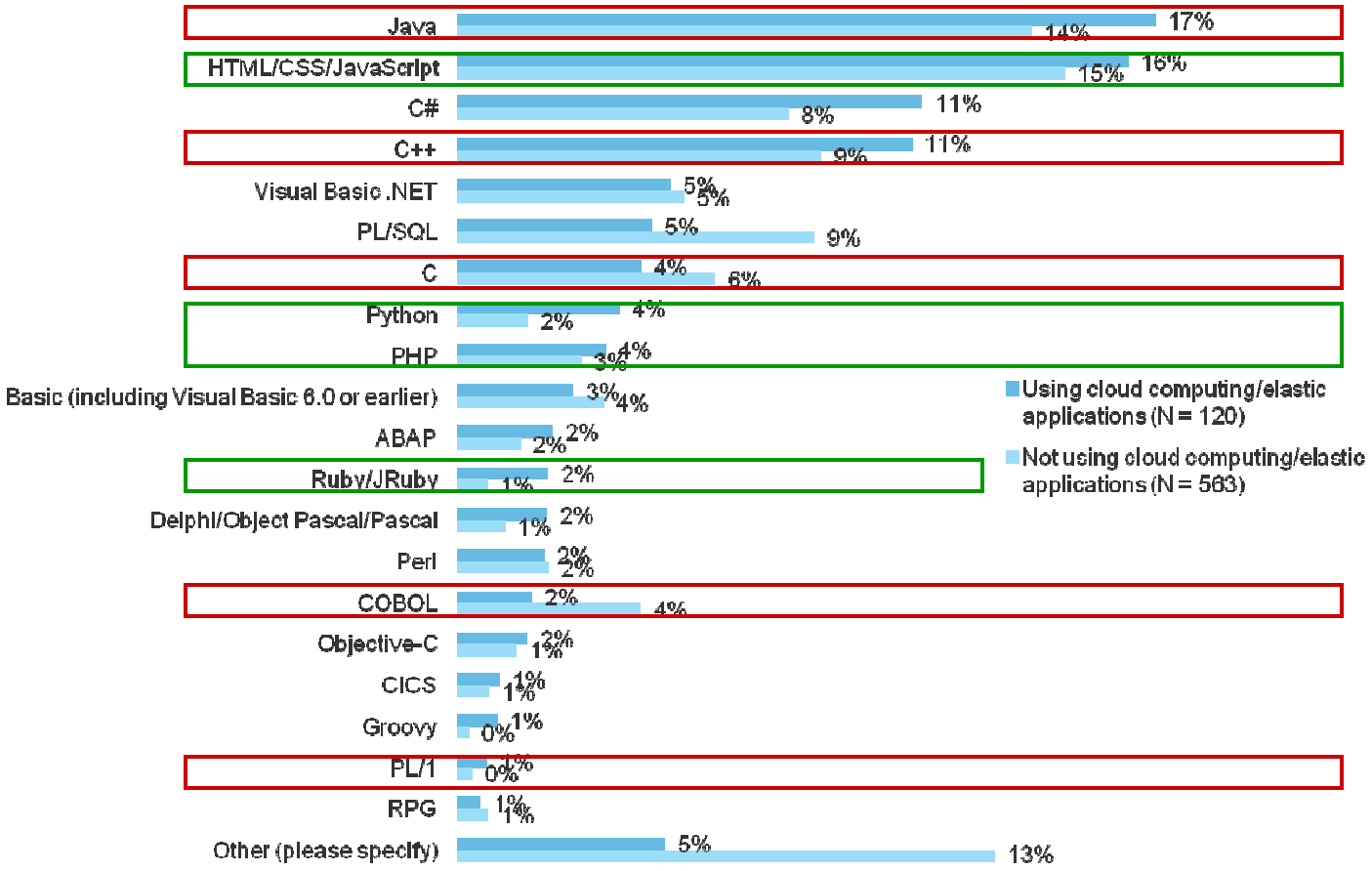
=> ~37%

Base: North American and European enterprise software developers; Source: Forrsights Developer Survey, Q1 2013

Scripting languages gaining significant momentum

Java leads developer language choice

"How do you allocate the time you spend writing code across the following programming languages?" (Enter a percentage for each)



=> ~37%

=> ~26%

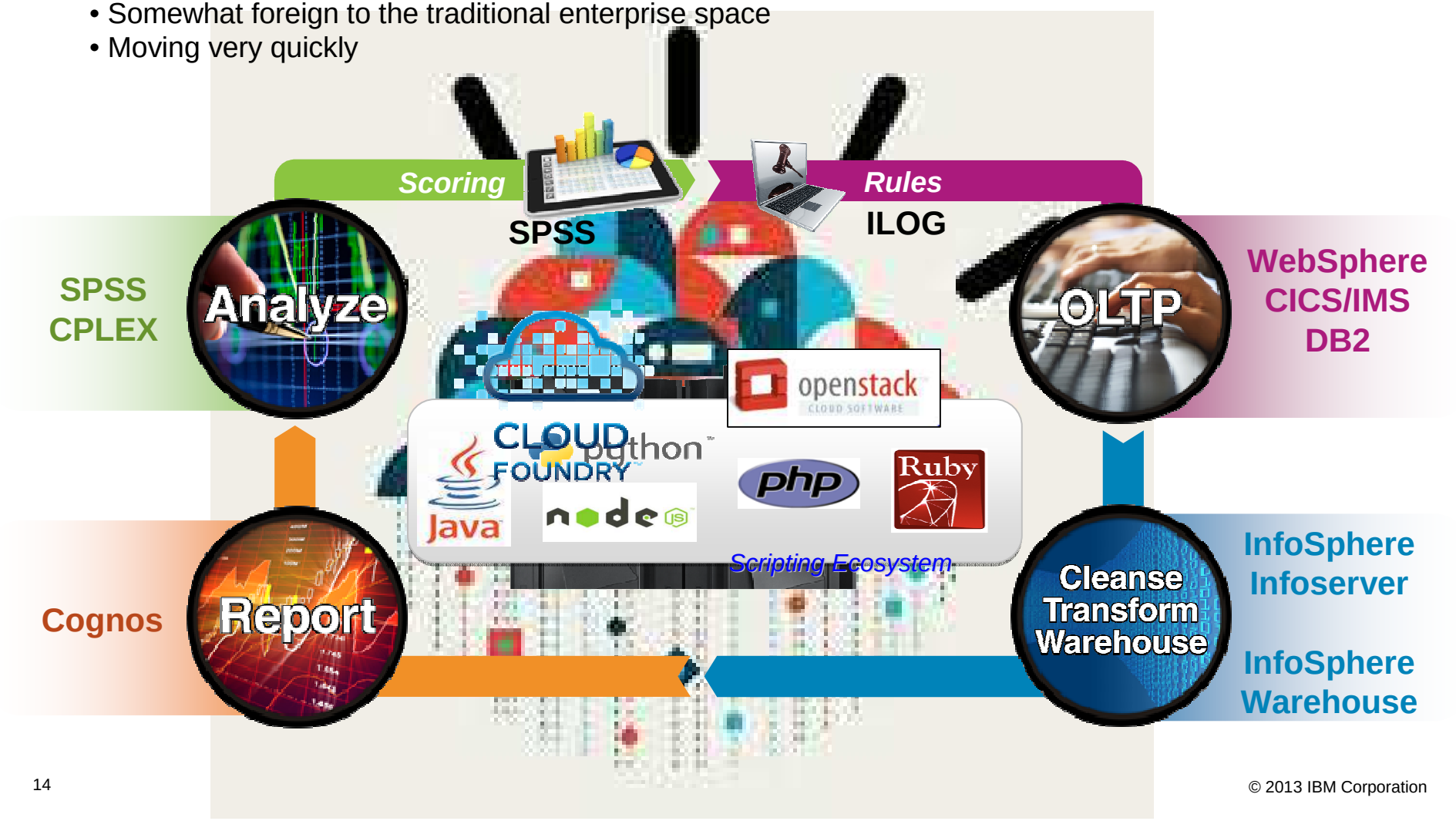
Base: North American and European enterprise software developers; Source: Forrsights Developer Survey, Q1 2013

Evolution of the Enterprise Computing Ecosystem

Scripting

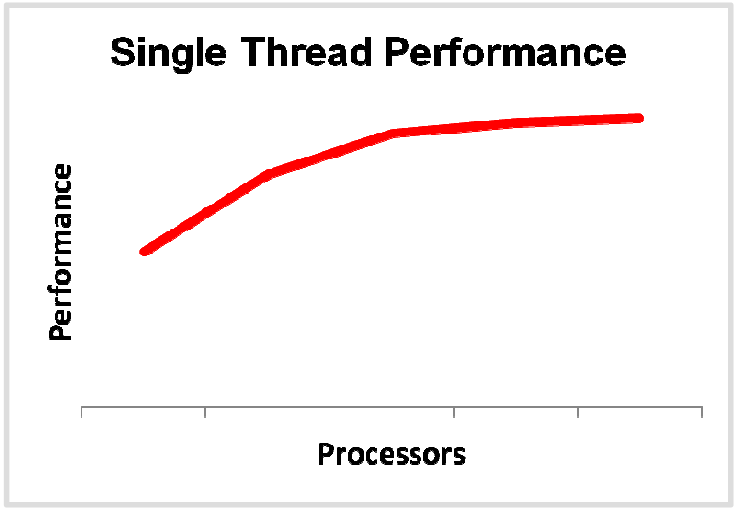
Cloud and DevOps represent a new and fast evolving tool-chain

- Client-side scritping languages migrating to the servers
- Somewhat foreign to the traditional enterprise space
- Moving very quickly



(Perceived) Single Thread Performance/Latency Matters

- SLA -> Service Level Agreements
  - Vendor guarantees response-time of transactions
  - Increasingly intelligent transactions
- Batch windows
  - Fixed elapsed window to complete
    eg. Balance books overnight before opening for business next day
- Plethora of S/W idioms that do not fall easily into divide-and-conquer
  - Finite-state-machine
  - Real-time analytics
  - Queuing/dispatching
  - Enterprise middleware
- Practical challenges of coarse-grain parallelism
  - Even very coarse parallelism can be non-trivial to implement
- Fine-grained parallelism… hold your breath?



Performance Innovation is no-Longer just a Processor Game

A range of aggressive hardware and systems designs
- Fit-for-purpose/Hybrid systems
- Appliances (eg. Netezza)
- GPUs/FGPAs
- Co-processors (eg. crypto)
- Transactional memory

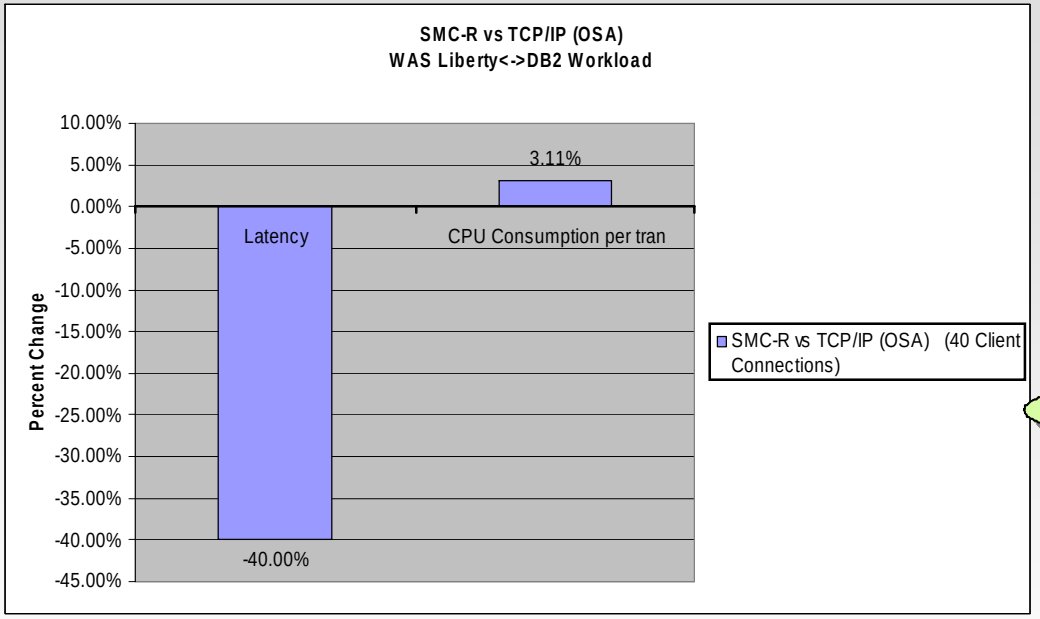
Winners/losers will be defined by a few key criteria
- Time-to-market and prevalence of core infrastructure
  - Stack opportunity is typically unclear + long lead-time/high cost => risk
- Ease of adoption/integration
  1. Runtime and middleware **(easiest)**
  2. Compiler optimization
  3. New programming models/libraries
  4. Hand written assembler **(hardest)**

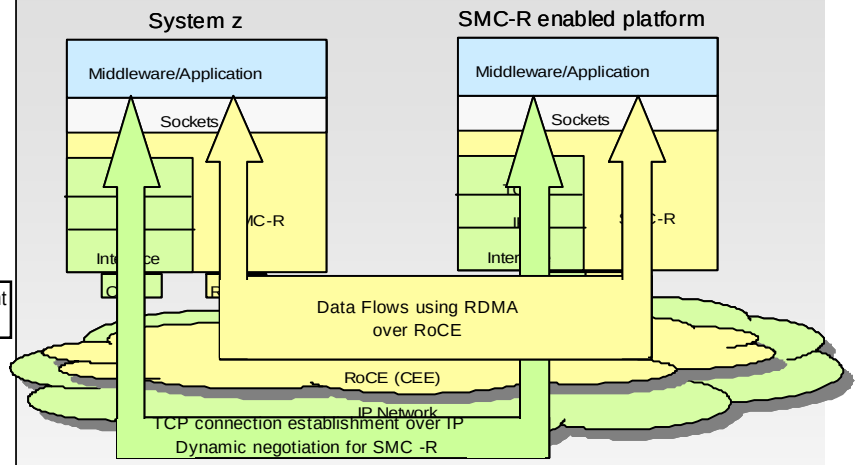
IBM® DB2® Analytics Accelerator

IBM® DB2® Analytics Accelerator for z/OS is a high-performance appliance that integrates IBM Netezza and zEnterprise technologies. The solution delivers extremely fast results for complex and data-intensive DB2 queries on data warehousing, business intelligence and analytic workloads.

Shared Memory Communications (SMC-R):

Exploit RDMA over Converged Ethernet (RoCE) with qualities of service support for dynamic failover to redundant hardware

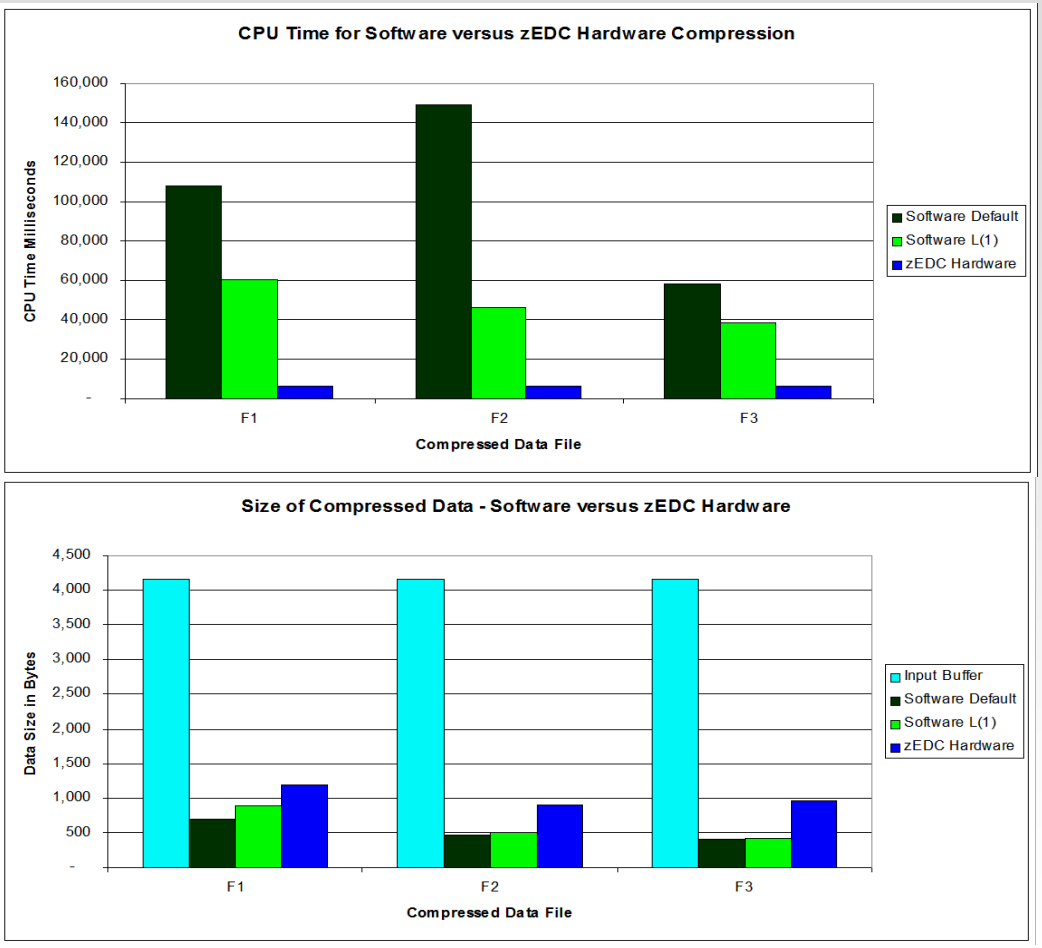




- Transparent exploitation for TCP sockets based applications
- Compatible with existing TCP/IP based load balancing solutions
- Up-to 40% reduction in end-to-end transaction latency
- Slight increase in CPU is due to very small message size in this workload (~100bytes). Workloads with larger payloads are expected to show a CPU savings

(Controlled measurement environment, results may vary)

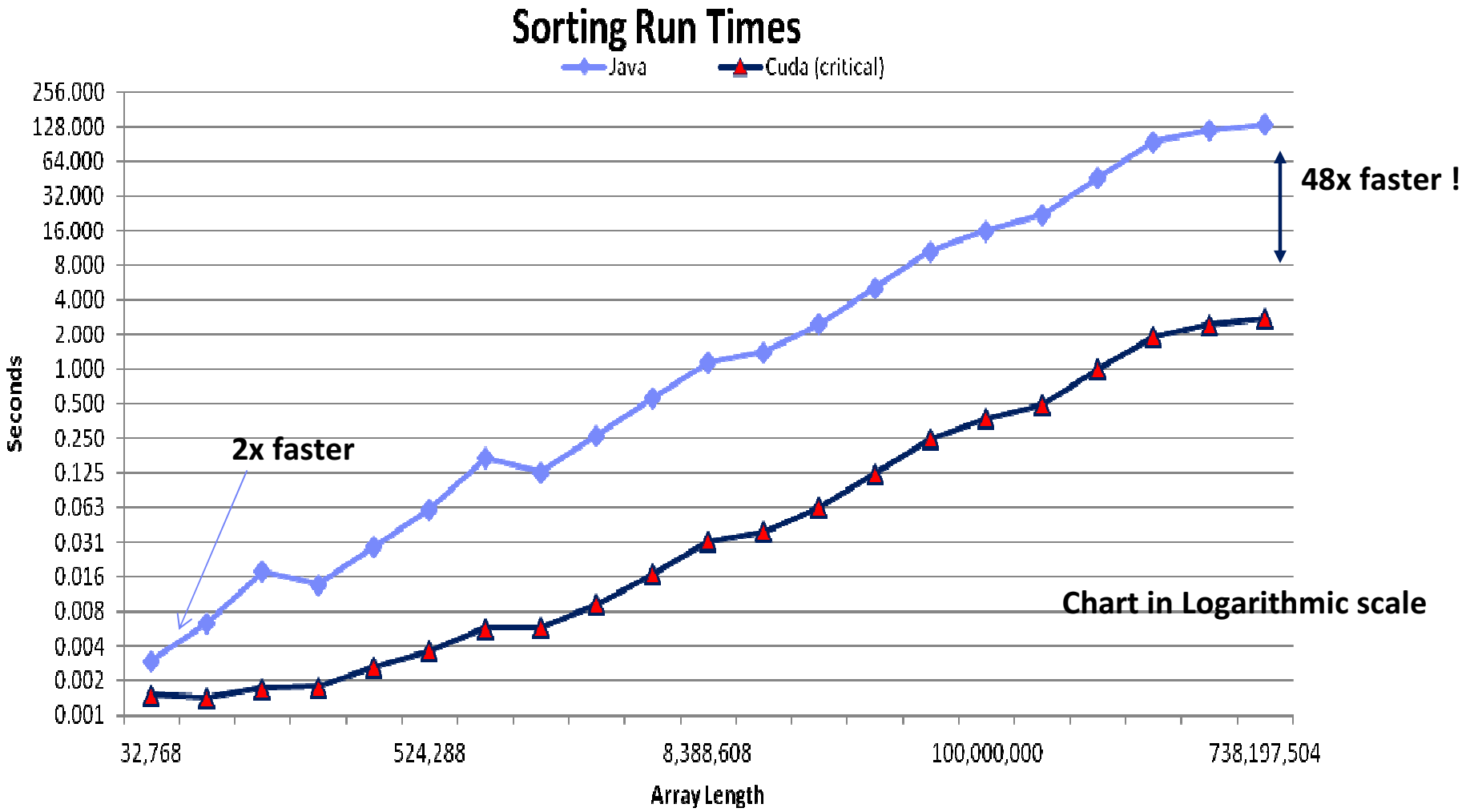
zEnterprise Data Compression (zEDC)



- **Exploited transparently through strandard Java APIs (eg. java/util/zip)**
- **Up-to 10x improvement in CPU time compressing data compared to L1 zlib**
- **Compression ratio of ~4x**

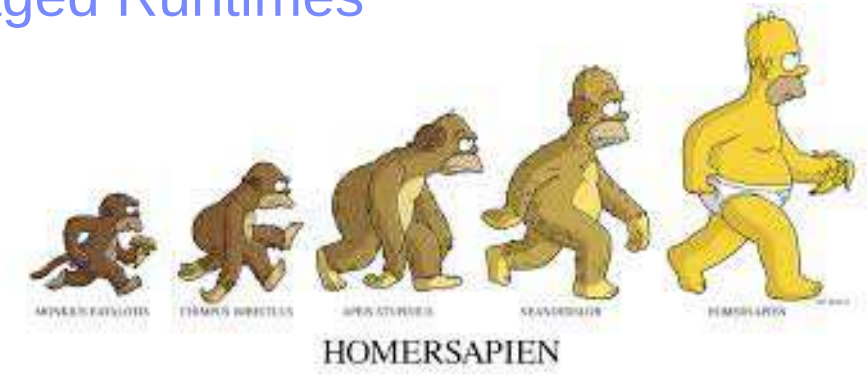
(Controlled measurement environment, results may vary)

GPU Acceleration on Standard Java Arrays



A New Age for Compilers and Managed Runtimes

Compilers and managed runtimes will need to evolve

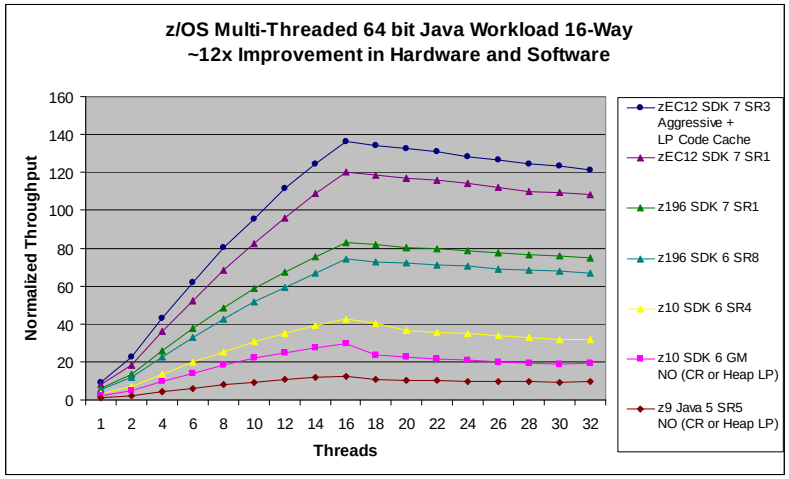


Today:

- Optimize once for single static architecture
- Synergy with micro-architecture is open-loop
- Parallelism is in its infancy
- Some dynamic in Java (Just-in-time)

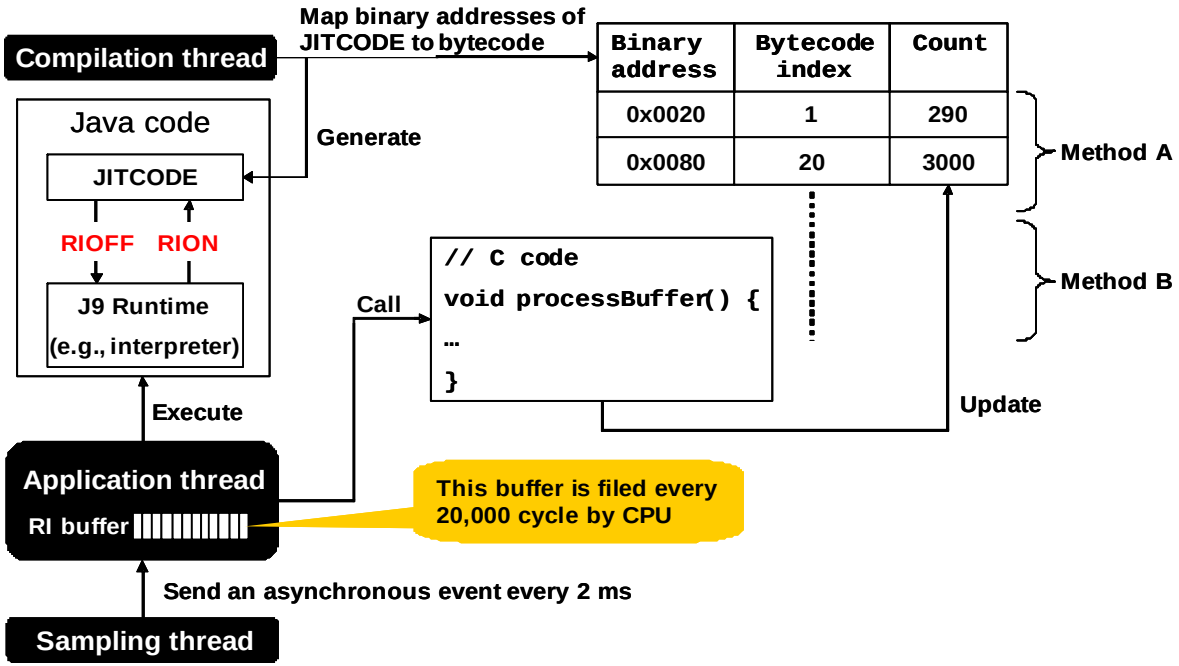
Tomorrow:

- Deeper synergy with micro-architecture
- Providing separation of interests to hedge risk from large changes to h/w
- Automatic parallelism (thread, data, etc)
- Hybrid systems, accelerators, fit-for-purpose
- Adaptive, dynamic and specialized
- JITing for light weight scripting environments

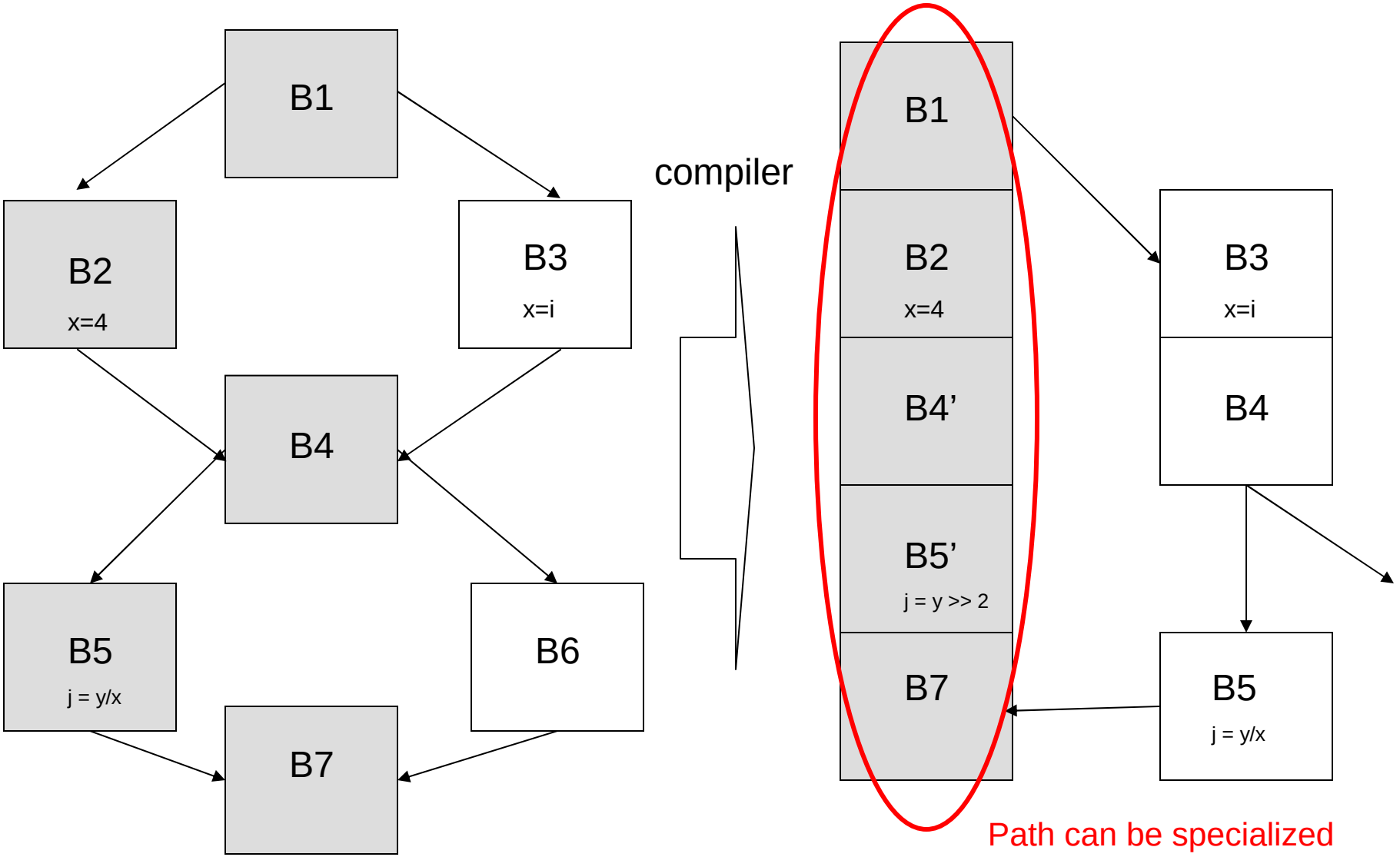


zEC12 Runtime-Instrumentation: H/W for Managed Runtimes

- Highly configurable trace-sampling mechanism
  - Events: Data/instruction cache miss information, register values
  - Paths: Last N taken branches
  - Correlated value, event and path profiling
- Integrating into IBM JVM
  - Java Runtime Environment is designed to be adaptive and self-tuning
  - RI enhances JRE decision-making by providing real-time feedback



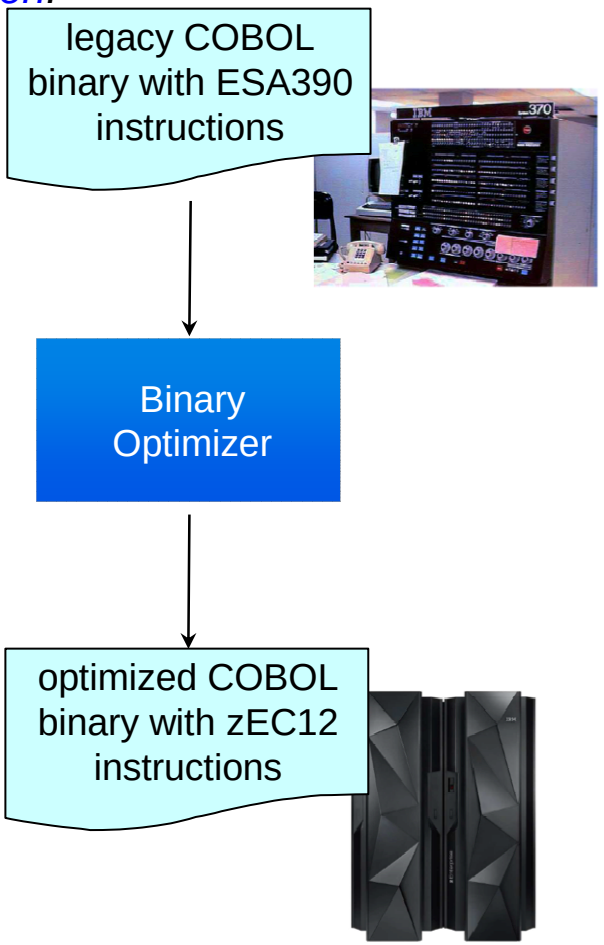
RI Example: Path Splitting/Specialization



Binary Optimization

Technology that enables re-optimization of legacy COBOL binaries on the latest System z without requiring source-level re-compilation.

- Binary Optimization technology being built by IBM Research (Tokyo)
  - Solution for the "Dusty Deck Problem": Can re-optimize the existing large body of legacy code that the customers are unable and/or unwilling to recompile
  - Built on top of IBM Testarossa Optimizer
- Value
  - Upgrade the binary from really old ISA to the exact ISA of the machine the code is running on
  - Inject the latest COBOL optimization technology into legacy programs
  - Ability to start utilizing profiling and RI-based feedback into to optimize the program
- Experimental Results*
  - up-to ***4.62x*** and average ***1.89x*** over the original binary on z196
  - up-to ***3.31x*** and average ***1.94x*** over the original binary on zEC12
- Alpha-level prototype available on developerWorks



* Measured using twelve benchmarks in the internal testsuite, used by the COBOL compiler dev. team.

© 2013 IBM Corporation

Concurrency and Parallelism in the Enterprise

Traditionally a play in niche spaces (eg. HPC)

With industry focus on business intelligence, analytics and optimization, **stakes have reached a new high**

SPSS

ILOG An IBM Company

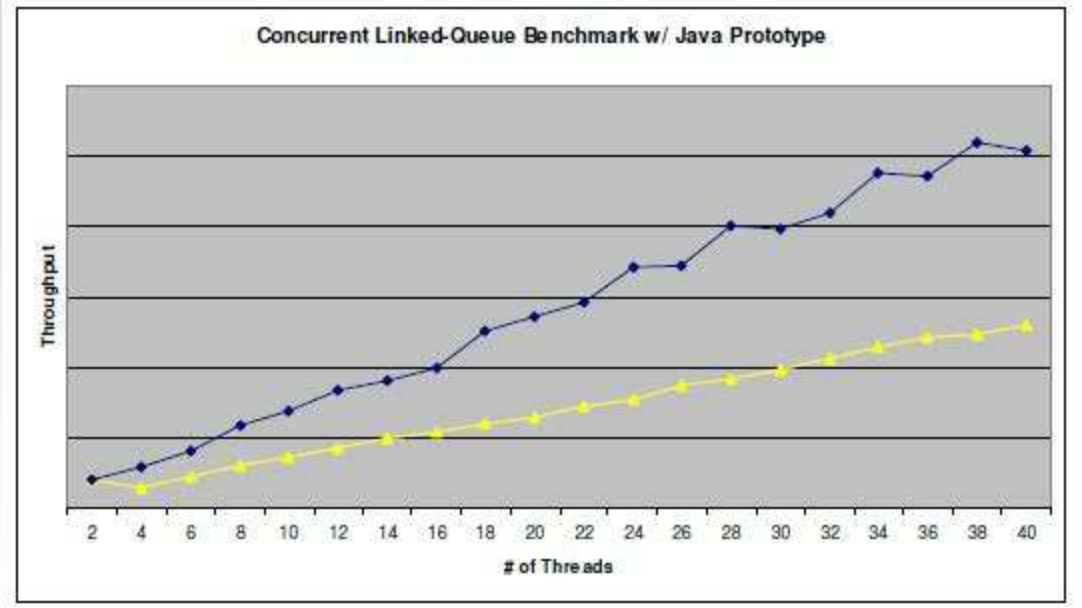
Algorithmics an IBM Company

COGNOS AN IBM® COMPANY

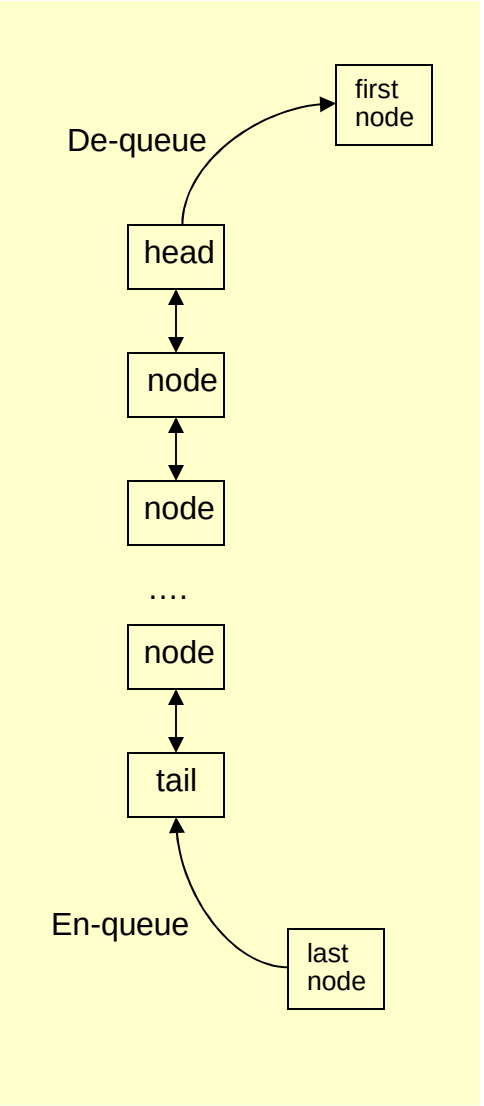
- 'Programming models for parallelism becoming mainstream
  - Eg. java/util/concurrent, fork/join and Lambdas
- Traditional Programming models evolving to meet the needs of enterprise computing
  - Eg. OpenMP adds tasks
- Hardware transactional memory… it's here!
- Auto-parallel remains elusive!

Transactional Execution: Concurrent Linked Queue

- ***~2x improved scalability of juc.ConcurrentLinkedQueue***
- ***Unbound Thread-Safe LinkedQueue***
  - First-in-first-out (FIFO)
    - Insert elements into tail (en-queue)
    - Poll elements from head (de-queue)
  - No explicit locking required
- ***Example usage: a multi-threaded work queue***
  - Tasks are inserted into a concurrent linked queue as multiple worker threads poll work from it concurrently



(Controlled measurement environment, results may vary)

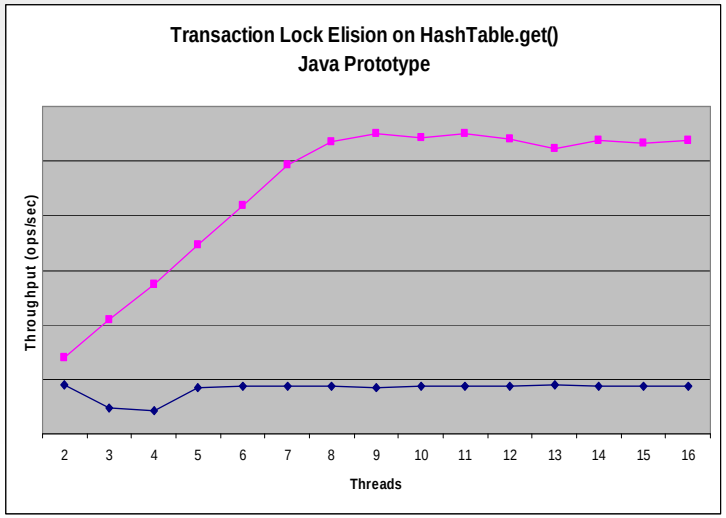


HTM Example: Transactional Lock Elision (TLE)

e·lide [ih-lahyd] Show IPA

verb (used with object), **e·lid·ed, e·lid·ing.**

1. to omit (a vowel, consonant, or syllable) in pronunciation.
2. to suppress; omit; ignore; pass over.
3. *Law*. to annul or quash.

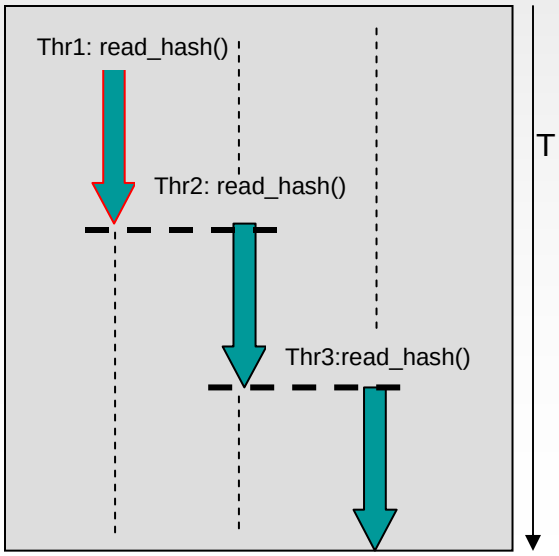


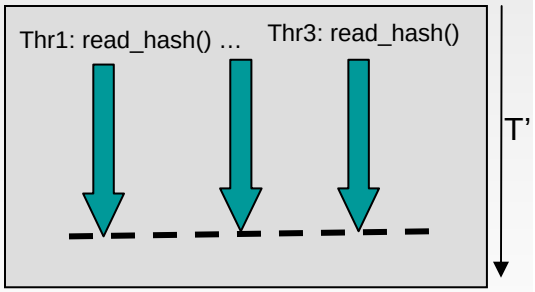
Threads must serialize despite only reading… just in-case a writer updates the hash

```
read_hash(key) {
  Wait_for_lock();
  read(hash, key);
  Release_lock();
}
```

Lock elision allows readers to execute in parallel, and safely back-out should a writer update hash

```
read_hash(key)
  TRANSACTION_BEGIN
  read hash.lock;
  BRNE serialize_on_hash_lock
  read (hash, key);
  TRANSACTION_END
```





Java8: Language Innovation – Lambdas and Parallelism

New syntax to allow concise code snippets and expression

- Useful for sending code to java.lang.concurrent
- On the path to enabling more parallelisms

```
Collections.sort(people, new Comparator<Person>() {
    public int compare(Person x, Person y) {
        return x.getLastName().compareTo(y.getLastName());
    }
});
```

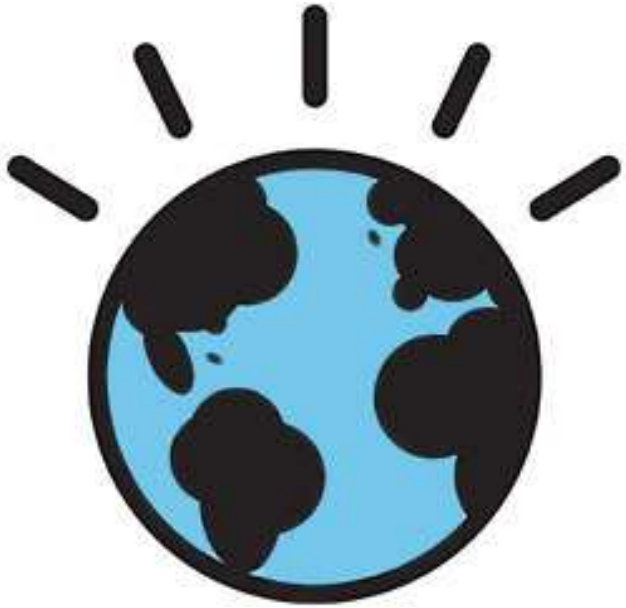
```
people.sort(comparing(Person::getLastName));
```

http://www.dzone.com/links/presentation_languagelibraryvm_coevolution_in_jav.html

So what about auto-parallelism?

A MOUTHFUL to chew on:

- Will the convergence of analytics/optimization and enterprise in the context of the end of the single-thread-performance-roadmap on the cloud provide enough momentum/focus to see some real-world (eg. Java) breakthroughs in auto-parallel technology?



Concluding Remarks

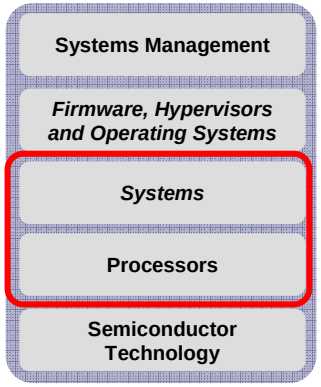
Lots to be excited about

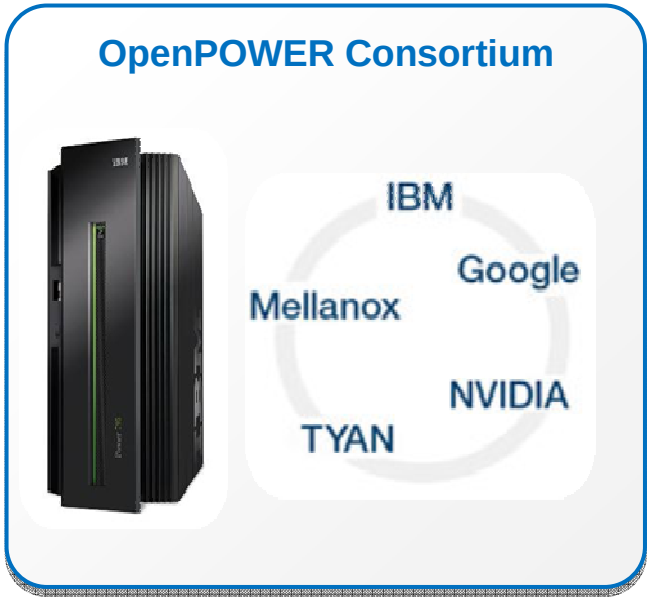
- Significant acceleration in innovation in the hardware/software interface
- It's never been a better time to be a runtime/compiler developer

Creating an Open Community to drive full stack innovation for the cloud: **OpenPOWER**

OpenPOWER Consortium – IBM POWER server technology creates an open community to drive innovation for the Cloud

- Industry's first open system design for cloud data centers
- Custom development group for hyperscale servers including hardware designs, firmware and software.
- Addresses need for industry-based innovation across processors, network and storage I/O
- OpenPower will create an ecosystem for Power Systems
  - IBM will contribute OpenSource software
  - IBM will enable industry participation through open documentation
  - IBM will license chip design intellectual property (IP) to allow customization





Co-Location by GC