redhat. | JBoss by Red Hat

Java. Cloud. Leadership.

The future of the Cloud

Dr Mark Little
Red Hat, Inc.
24/10/2012

Overview

- Where are we today and why?
- Mobile and Cloud
  - Ubiquitous computing in the large
- Hardware and software forces in action
- What does this mean for today's middleware offerings?
- The future of the JVM, Java and middleware

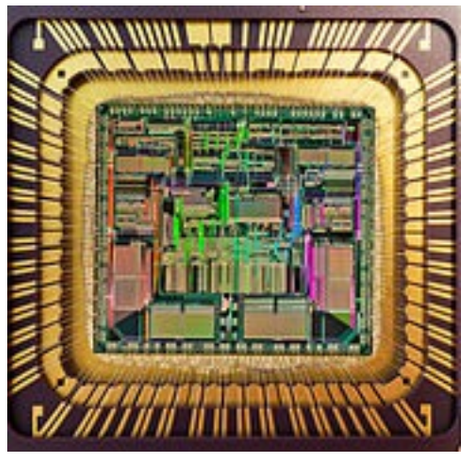
30 years ago …

- 16K was considered a lot of memory
- 140K floppy disks were the standard
- 10 mbps ethernet was decadent
- 8 bit 6502 processor was king for personal computing
- Wireless was what people listened to when there was nothing on TV

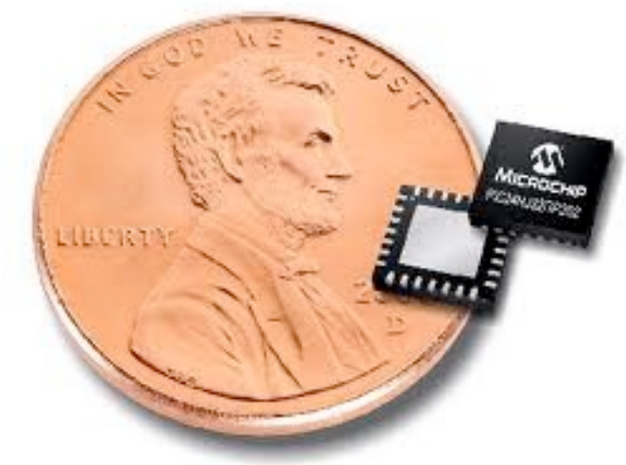
Today …

- 512Meg memory is standard on smart phones, 64Gig storage
- 256Gig USB sticks are the norm
- 100Gig ethernet at work and 30mbps to the home
- 64 bit quad core processors in laptops, 1GHz ARM in iPhone
- WiFi throughout many cities

The times have changed



- Motorola 68040 (1990)

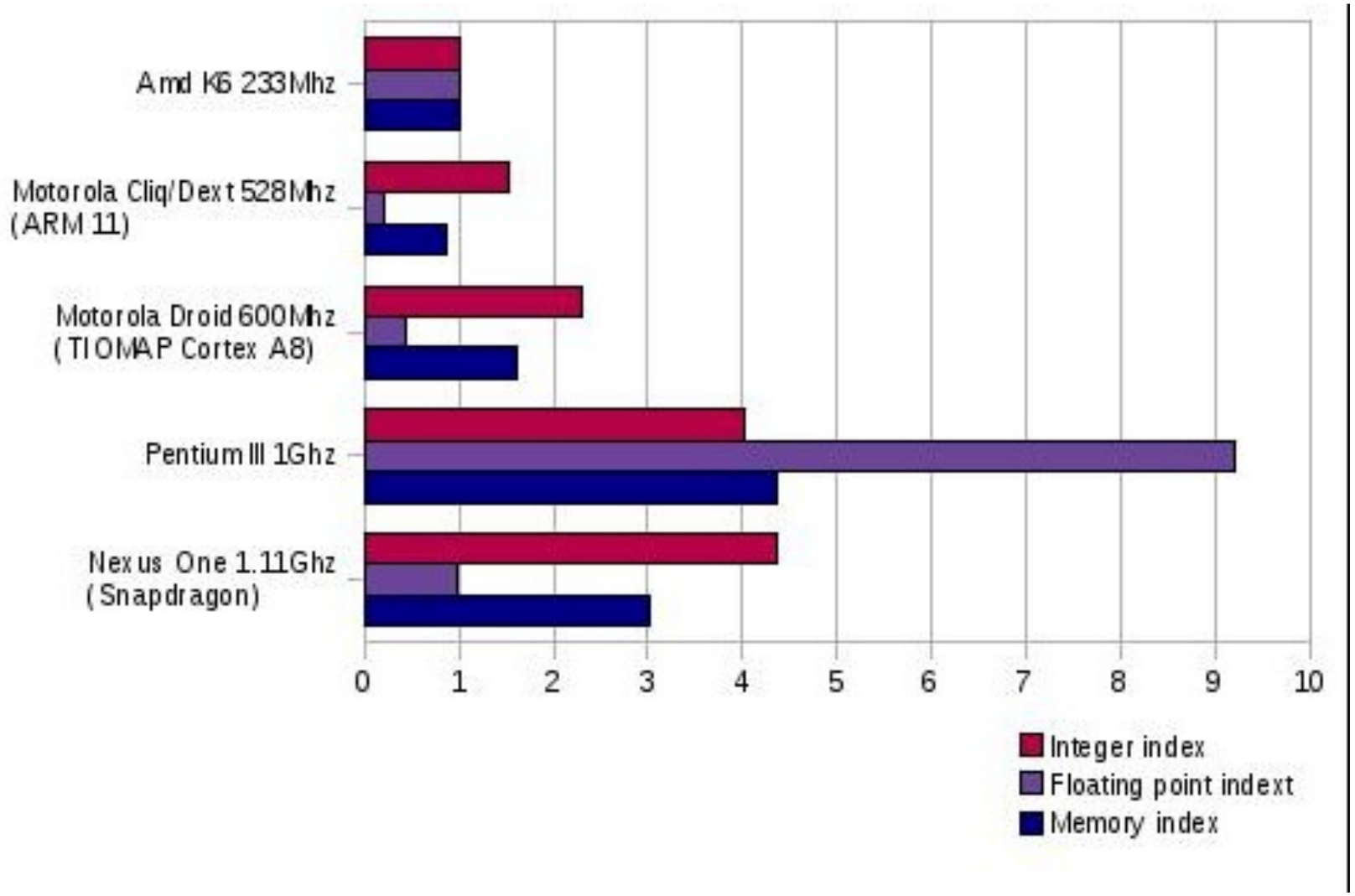


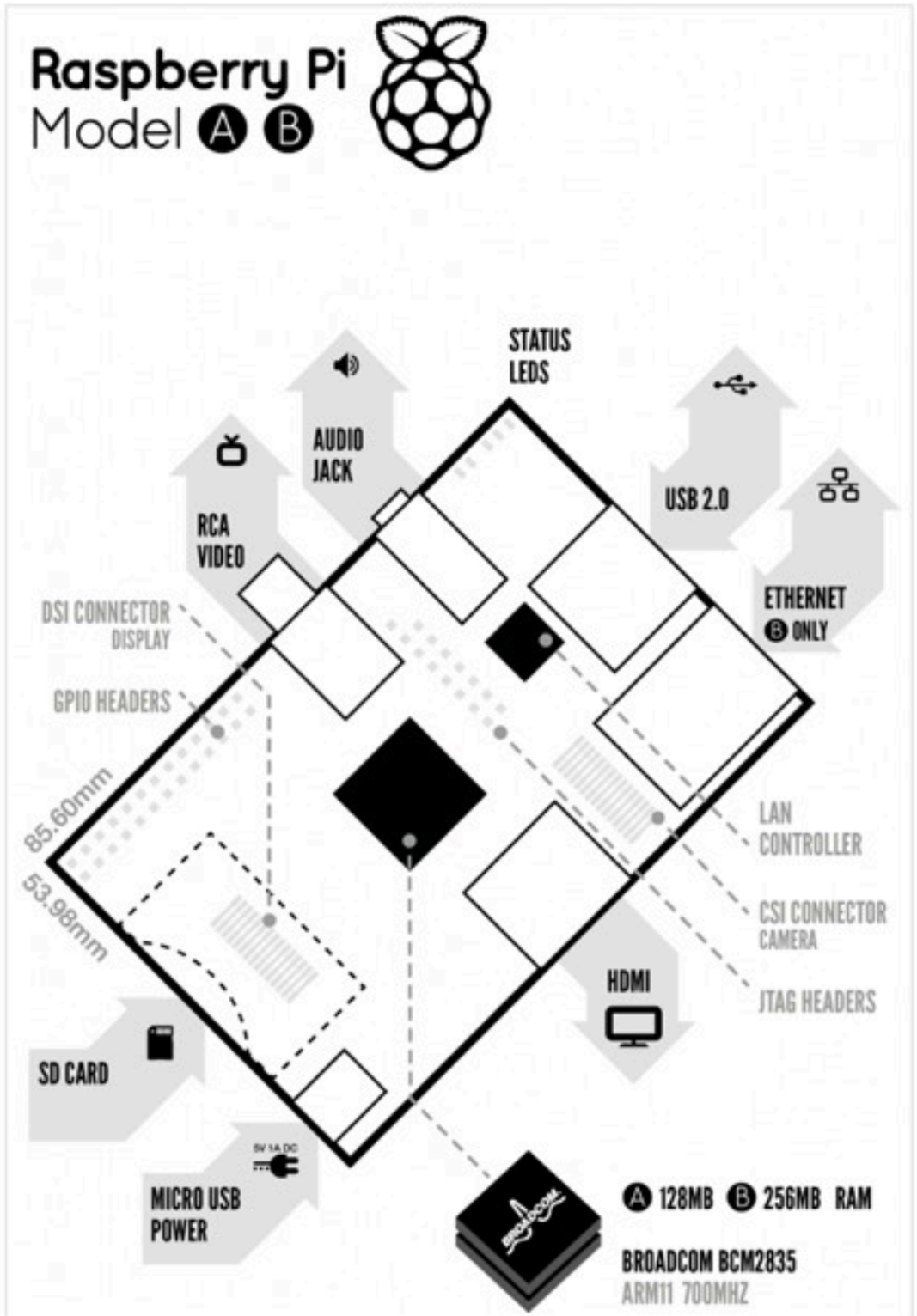
- dsPIC33FJ12GP 16-bit microcontroller (2007)

Java in 2000

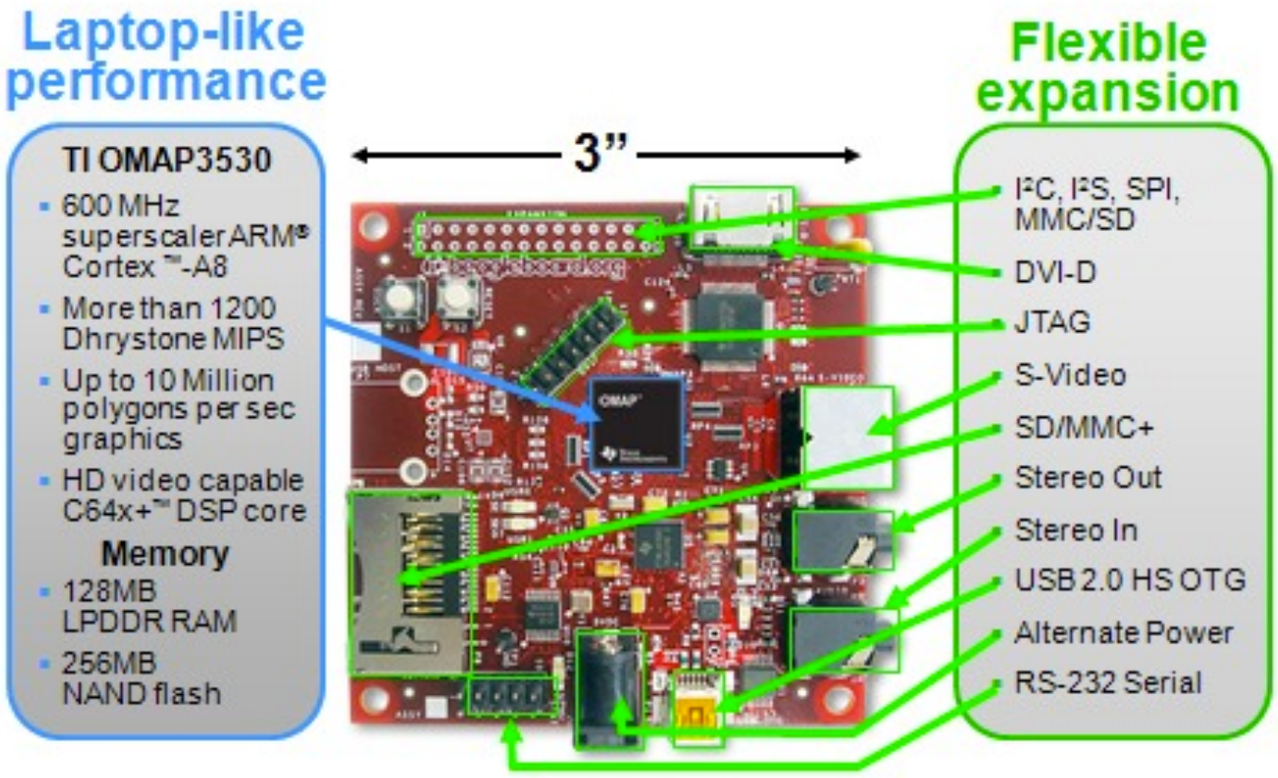
- J2SE 1.3 and J2EE 1.2 to 1.3
- Typical laptop configuration
  - 1 GHz Pentium III with 512 Meg and 20 Gig disk
  - Sufficient to run full J2EE stack
- Limited mobile devices
  - HP Jornada 720, 206MHz StrongArm SA1110 32-bit processor, 32 MB SDRAM, Windows CE
    - ChaiVM JDK 1.2
  - Sufficient to run some components of J2EE

2010 (mobile) versus 2000





Beagleboard



redhat. | JBoss by Red Hat

Java. Cloud. Leadership.

Java in 2012

- Java SE 7 and Java EE 6
- Typical laptop configuration
  - 2.2 GHz Quad Core i7 with 4 Gig and 500 Gig disk (SSD?)
  - Sufficient to run full EE6 stack (on several different OS concurrently)
- Smartphones
  - Samsung Galaxy S2, Dual Core 1.2 GHz Cortex-A9
    - Android 4.x
  - Sufficient to run many (all?) components of EE6

Cloud meets Mobile

- Public Clouds important
- Private Clouds probably more important
- But ***Personal Clouds*** may become the norm
  - Many devices are predicted to have processors/ sensors in the future
  - Cheaper to give them off-the-shelf processors
  - Tapping into this local cloud will be possible
- Thin clients are not sufficient
  - Shannon's Limit

Variant of Parkinson's Law

- "Work expands to use the power available"
  - Basic word processors on first PCs
  - Games from Pong through Space Invaders to CoD
  - Mobile devices contain more and more personal data
  - Disconnected operation is the normal situation

Application realities

- Types of application increasing in complexity
  - Online purchases
  - Distributed peer-to-peer interactions
- More requirements becoming a necessity
  - Security and identity
  - High performance, low latency, reliable messaging
  - Database updates with transactions
  - Workflows as inter-app interactions increase

Mobility and “enterprise” apps

Transactional invocation

Ad-hoc auction
Peer-based social networking
Decentralised calendar
Gaming

Mobiles replace consoles?

Sounds familiar?

- Many commonalities with (Java) middleware
  - Enterprise requirements for all but trivial apps
    - Messaging
    - Transactions
    - Security
    - ...
- Obviously Java is not the only application language
  - Lots of popular JVM-based languages
  - Polyglot JVM and stacks

So what does this mean?

- Middleware is needed whatever the deployment environment
  - Mainframes, servers, laptops etc.
- Middleware isn't defined by an implementation
- Mobile and Cloud should not be new silos for developers!
  - It's inefficient
  - It's expensive
  - It's time consuming

Present and future directions

- Built on existing middleware
  - Evolution rather than revolution
- Cloud is not the final solution, it's just another step
- Today "Cloud" means "servers"
  - It should be wider
  - More processors outside of "servers" than inside
- But there are many issues we need to resolve
  - "Cloud" exacerbates the situations …
  - Really scaling

Java Enterprise Edition 6

- Turns out that EE6 has many of the required capabilities
    - Standards based too!
- EE6 represents a great evolution for 40 years of work!
- New capabilities (e.g., JAX-RS, CDI, BeanValidation)
- Input from wider open source communities and users

“Java EE is too bloated”

- Differentiate the standard from implementations!
  - Bloatware should be a thing of the past
- It is possible to be lightweight and enterprise ready

The Open Source Java application server *reignited*

Designed for flexibility.
Amped with electrifying speed.
Launch your Java EE applications in a flash!

Lightning Fast... *start-up / deployment / configuration*

redhat. | JBoss by Red Hat

Java. Cloud. Leadership.

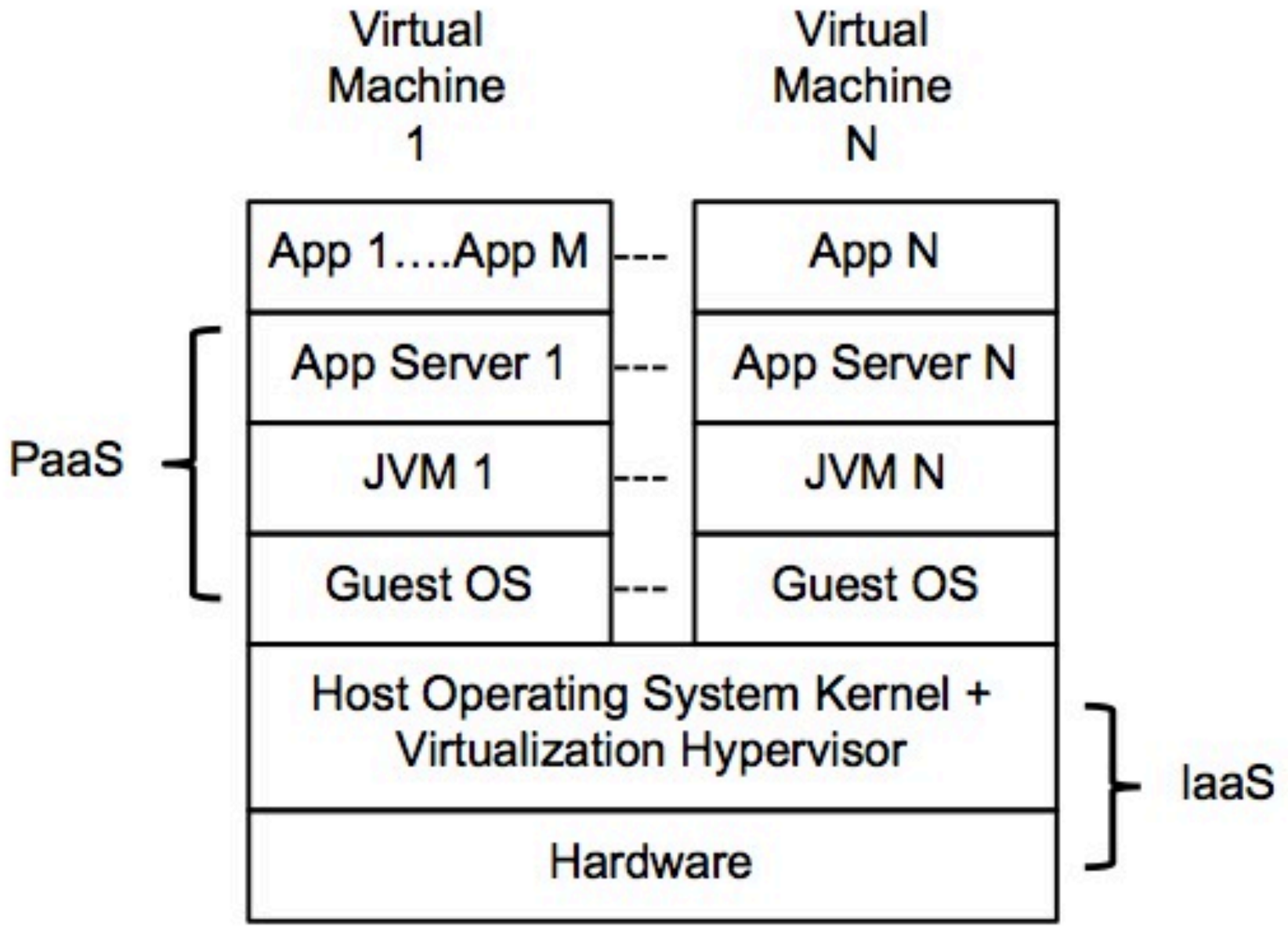
Not quite perfect ...

- The next generation of Java middleware needs to address ...
  - Coordination
  - ***Multi-tenancy***
  - ***Large-scale data storage***
  - Negotiation and enforcement of SLAs
  - Accountability
- http://2011.middleware-conference.org/fome/fome2011

Multi-tenancy

- PaaS and mobile requires high density applications
  - Multiple applications running in each VM instance
- Java middleware stacks are typically not architected for multi-tenancy
  - JVM and application server run one application at a time
  - We forgot some of the good things we learnt in OS classes!

Middleware stack



What's needed?

- The JVM needs to change
  - Multiple processes per JVM or slimmer footprint
  - Red-zone protected
    - Independent failures
    - Real-time
    - Deterministic scheduling
- Dalvik does a good job but ...
- Improvements in OpenJDK
  - Java SE 1.8 modularity, OSGi modularity

Large scale data storage

- RDBMS are great generalist solutions
  - ACID transactions
  - Strong consistency
  - Replication to reduce single point of failure
- But they were not designed for large scale data
  - Quantity
  - Distributed
- Resulted in NoSQL and NewSQL efforts

NoSQL

- Weak consistency replication
- Gossip protocols for disseminating updates
- Typical (topical?) to ignore transactions
  - CAP theorem cited but sometimes an excuse
- But weak consistency requires applications to be aware
  - Complicated for developers
  - Requirements may change over time

What's needed?

- RDBMS and SQL aren't going away
- Transactions or extended transactions in NoSQL
  - Relaxing ACID properties may help
- Explore the trade-offs between strong and weak consistency
  - Memory is becoming the new disk
  - Impact on applications, new frameworks, design guidelines
- JSR 347 (Data Grids); other JSRs and efforts in open source

Conclusions

- The next decade will be defined by ubiquitous computing
- Enterprise capabilities will be a necessity
- Java and the JVM should play a leading role
- There are still areas that need to be addressed for this to happen
  - Great time to be a Java (JVM) developer!

Richard Hamming, 1968 Turing speech

- Whereas Newton could say, "If I have seen a little farther than others, it is because I have stood on the shoulders of giants," I am forced to say, "Today we stand on each other's feet." Perhaps the central problem we face in all of computer science is how we are to get to the situation where we build on top of the work of others rather than redoing so much of it in a trivially different way.