



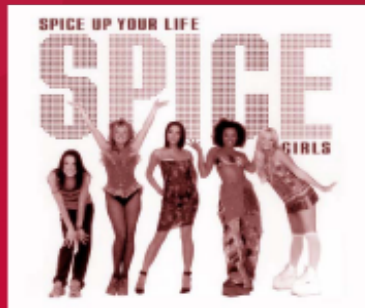
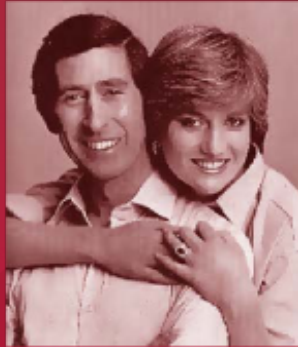
waratek

Achieving genuine Java multitenancy

Waratek The Cloud VM for JAVA

1996

1996



Java is

Launched

Java is launched

Used by 80% of the top 5000 IT users worldwide.

World's leading computer language.

More than 1/3 of all installed enterprise applications.



9 million programmers

An eco - system of JVM based languages has sprung up around it.

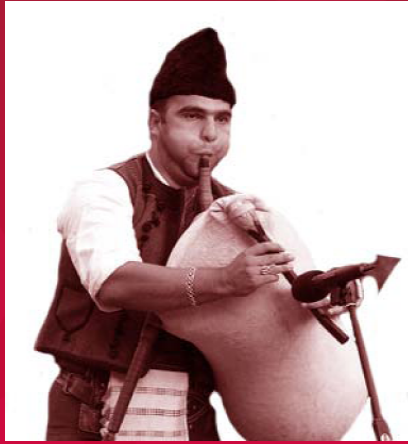
2 trillion dollars

Sunk investment.



As a language

More speakers than Swedish, Bulgarian and Hebrew.



The world moves on...

87 billion emails are sent daily.

The digital universe doubles every 18 months.

Network attached storage devices double every two years.

80 to 85% of IT budgets go to operational maintenance.

The world moves on...

Increase capacity without investing in infrastructure, training or licensing software.



The Cloud... a tsunami

It could overrun all pre-existing paradigms.

Unprecedented advantages:
multitenancy / instant scalability / granular elasticity.



Transition to The Cloud is the new revolution

But JVMs that give Java its power are architecturally stuck in the server capabilities of 1990s.

The Cloud... a tsunami

Could wipe out the equivalent GNP of the state of California.

Or the entire Russian economy.

A technology breakthrough

Introducing Waratek The Cloud VM for JAVA





The Problem with Java

- Java is 15 years old, predates multitenant cloud computing
- J2EE app servers were designed to host multiple webapps, but rarely do in practice because there's no isolation between webapps
- Today's Java/J2EE apps are deployed as "one app per app-server per JVM", which leads to "app-server sprawl"
- Hosting one webapp per JVM means provisioning fixed cpu/memory quotas which can't elastically fluctuate on demand.



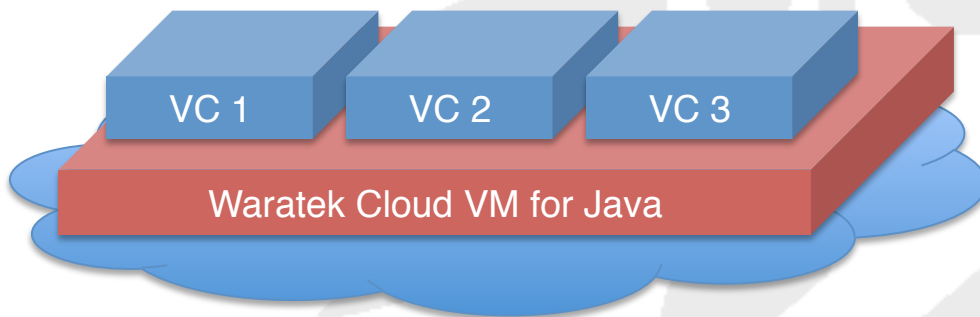
The Problem with Java

- Hosting one webapp per fixed cpu/memory quota means there is unused cpu/memory capacity being paid for: this is a waste
- Today's Java/J2EE apps are perfectly suited to a multitenant "pay exactly what you consume" model were there a multitenant JVM
- A multitenant JVM would be able to charge computing power like a utility: gigahertz-hours for cpu, gigabyte-hours for memory



Genuine multitenant JVM for the Cloud

- The Waratek Cloud VM introduces a feature called Virtual Containers (VC)
- A VC is a metacircular VM within the VM which shares the host VM environment (heap, classes, JIT) with other VCs
- A VC is extremely lightweight: < 1 MB overhead per VC





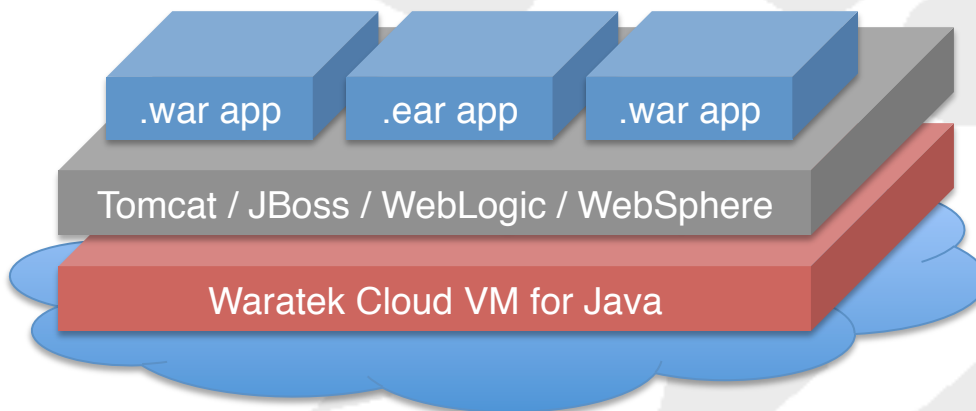
Fine-grained resource control

- A single VM can host thousands of VCs:
a VC can be any size (from 1MB up to many GBs)
- Every aspect of VC execution is isolated:
cpu priority, memory limits, bandwidth, etc
- Every aspect of VC use is metered:
cpu use in Hz, memory/IO use in Bytes
- VCs provide computing power like electricity kW-h:
cpu consumed in GHz-h, memory consumed in GB-h



Waratek Cloud VM delivers Java-as-a-Service

- Run existing Java/J2EE platform software as a multitenant cloud service without code change: binary compatible with existing apps and platforms
- Every .war/.ear app gets its own VC





Open framework for extensible VCs

- VC framework is extensible and customizable: enhanced VCs can be created using Waratek APIs
- Special-purpose VCs already in dev or planning: Servlet VC, Scala VC, Clojure VC, Python VC, Ruby VC
- Waratek is adding a range of premium VC features: VC mirroring for disaster-recovery, live snapshot, live migration, etc.



Waratek Cloud VM for Java Summary

- ✓ Genuine multitenancy
- ✓ Precise metering of compute usage, by application
- ✓ Allocation and prioritization of resources in real time, by application
- ✓ Instant scalability
- ✓ Granular elasticity
- ✓ Binary compatibility

Thank You

Test Drive the Waratek
Cloud VM for Java TODAY!

Visit: www.waratek.com
