Zing ReadyNow!

Faster Java execution at application startup – and beyond

ZING'S READYNOW! FEATURES

- Designed for Java-based commercial applications
- Allows Java applications to start up faster and stay fast
- Eliminates the mass of slow transactions that typically accompany any new code rollout
- Gives developers fine-grained control over Java compilation
- Greatly improves Java application responsiveness to demand spikes
- Included as part of Zing, Azul's "pauseless" JVM

WHAT'S THE JAVA WARM-UP PROBLEM?

Until today, Java developers and operations teams working with commercial applications have faced a daily challenge – overcoming Java's warm-up issues. For instance, whenever a DevOps team using a continuous development methodology does a code rollout, every Java instance will suffer from thousands of slow transactions while Java JIT compilers optimized the new code.

Warm-up is a problem for every Java runtime (aka the JVM) except one – Zing. We've taken the folklore and guess-work out of Java, and given you control over how Java behaves when it matters most – when a new feature is launched, a retailer launches a flash sale or in extreme cases even when a trading system needs peak performance with the first few trades.

WHAT CAUSES JAVA WARM-UP ISSUES?

Java was designed to start up quickly, then improve performance over time by repeatedly compiling and optimizing frequently used code. The JVM's Just-in-Time (JIT) compilers use constantly-updated profile data that describes which parts of the application are called the most (the "hot" code). JIT compilation allows the JVM to optimize performance, but it takes time, and every time conditions change the JVM falls back to interpreted code until it identifies which methods (and even which code branches) are now most frequently used.

In DevOps use cases that do multiple code rollouts per day (or per week) the operational impact is extremely visible – every Java instance "learns" which sections of code need to be optimized, and while it is doing that, thousands of transactions execute slowly.

Today there is only one solution to manage this warm-up problem: ReadyNow technology from Azul Systems – a key feature of the Zing runtime for Java.

INTRODUCING ZING WITH READYNOW! TECHNOLOGY

Originally designed for extreme use cases like low-latency trading or messaging platforms, ReadyNow has evolved to become an essential component for your DevOps toolkit.

ReadyNow technology is built into every Zing release and every Zing subscription.

With ReadyNow, the blizzard of slow transactions caused by Java warm-up issues can be a thing of the past. With each new code rollout, ReadyNow enables the runtime optimizations generated by a single server to be applied on the rest of the servers running your application. You can see the performance impact with each new release – instead of 20 (or 100, or 1000) servers providing sluggish performance for thousands of transactions, one server makes the optimization decisions and shares them with every other machine running the same application. The impact on operating metrics is immediate, and the performance impact as new code is deployed is diminished by orders of magnitude – every time.

Note: For developers with uncommon and extremely latency-sensitive performance requirements, ReadyNow also provides a set of robust (Xtreme) APIs and compiler directives that provide more control over Java's JIT compilation.

Zing ReadyNow! Features

- Improves the effectiveness of warm-up strategies and gets applications running at acceptable performance levels faster in continuous development environments
- Delivers more consistent Java application performance, especially in the presence of demand spikes throughout the day
- Allows operations teams and DevOps to use compiled code profiles across runs
- Provides developers with APIs that control compilation policy, reduce de-optimization, and direct startup processes that eliminate the need to warm-up the JVM
- Optimized for 64-bit Linux on x86
- Works with Zing's Azul-optimized LLVM-based Falcon JIT compiler

SUPPORTED PLATFORMS

Processor

- Intel: Xeon server class processors released 2009 and later
- AMD: Opteron server class processors released 2010 and later

Memory and CPU Cores Recommended

- 1 GB to 8 TB heap
- 2 cores or more

Supported Operating Systems

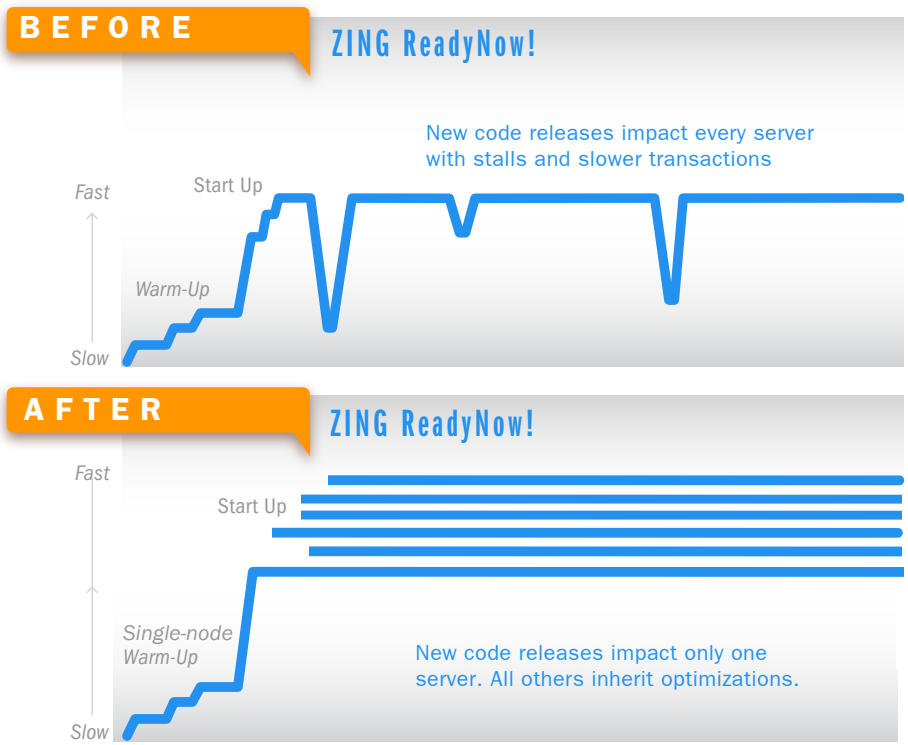
- 64 bit Linux (x86)
- Red Hat Enterprise Linux/CentOS 5.2 or later
- Red Hat Enterprise Linux/CentOS 6.0 or later
- Red Hat Enterprise Linux/CentOS 7.0 or later
- Red Hat Enterprise MRG Realtime 2.3 or later
- SUSE Linux Enterprise Server 11 sp1, sp2, sp3
- Oracle Linux 5.x or 6.x (kernel specific)
- Ubuntu 10.04 LTS (Lucid Lynx)
- Ubuntu 12.04 LTS (Precise Pangolin)
- Ubuntu 14.04 LTS (Trusty Tahr)
- Debian Wheezy,Jesse,Stretch
- Amazon Linux
- Other Linux distributions supported via Dynamic Kernel Module System

JDK Versions

- Java 8, 7 and 6

Selected Zing ReadyNow! use cases

- Social media monitoring
- In-memory Big Data Analytics
- Online Retail systems with dynamic offers
- Stream processing
- Large portals and other web-scale applications
- Real-time advertising networks
- Large-scale online and social gaming
- SLA-driven complex event processing
- Real-time messaging applications
- Low-latency financial trading applications



Contact Azul Systems:

385 Moffett Park Drive
Suite 115
Sunnyvale, CA
94089 USA

T + 1.650.230.6500
F + 1.650.230.6600

www.azul.com/solutions/zing/readynow