

## ZING

### Zing is a better JVM. Period.

With Zing, Java applications simply show better runtime behavior and help you meet interactive and machine-machine SLAs.

### C4 Garbage Collector

Continuous, concurrent, and consistent operation.

### Falcon JIT Compiler

Modern, modular server tier compiler leveraging LLVM.

### ReadyNow! Technology

Solves Java warm-up problems.

### Zing Vision (ZVision)

A zero-overhead, always-on, Azul production monitoring tool for analyzing and troubleshooting Java applications running on the Zing JVM.

## ZING ADVANTAGES

- › Takes full advantage of the large memory and multiple CPU cores available in today's servers
- › Supports Java heaps up to 8 TB, reducing or eliminating the need for off-heap technologies
- › Scales rapidly, with sustained memory allocation rates beyond 30GB/sec
- › Removes Java Garbage Collection pauses as a factor limiting your business
- › Eliminates response time and latency outliers
- › Delivers a better user experience
- › Minimizes JVM and application tuning for faster time to market
- › Starts fast and stays fast – solves “warm up” issues
- › 50% reduction in TCO vs. other Java platforms
- › Optimized for x86/64 servers and all major Linux distros

# Zing®: The best JVM for Java Workloads

## INTRODUCING ZING

Today Java is ubiquitous across the enterprise. Flexible and powerful, Java is the ideal choice for development, DevOps and operations teams worldwide.

Zing builds upon Java's advantages by delivering a robust, highly scalable Java Virtual Machine (JVM) that matches the needs of today's business, from newly-developed microservice-based applications to long-running legacy systems. Zing is the best JVM choice for all Java workloads, including online retail, SaaS or Cloud-based deployments, insurance portals, multi-user gaming platforms, or Big Data. Zing has even been widely deployed in low-latency systems for capital markets – and ensures smooth operation anywhere predictable, glitch-free Java is essential. Zing works great in Docker and LXC containers, too.

Zing enables Java developers to make efficient and effective use of server resources or VM instances – without the random stalls, pauses and jitter that have been part of Java's heritage. Zing also solves Java “warm-up” problems that can degrade performance at the start of trading or other time-sensitive operations. With improved memory-handling and a more stable, consistent runtime platform, Java developers can rely upon Zing as they build and deploy richer applications, driving new revenue streams and supporting new innovations.

Whether your application requires human-scale response times or is machine-scale, measuring peak response time in microseconds, Zing allows you to meet even the most demanding service level agreements without re-architecting or depending upon JVM tuning experts – helping you take advantage of new business opportunities faster and with lower operating costs.

## INSIDE ZING

At its core, Zing uses Azul's Continuously Concurrent Compacting Collector (C4) instead of the garbage collectors built into Oracle's Hotspot. The C4 collector eliminates the “stop-the-world” garbage collection (GC) pauses that limit the scalability of all legacy JVMs. Zing includes the Falcon server tier compiler, leveraging industry efforts in the LLVM compiler engine, for rock solid performance. Struggling with Java warmup issues? Azul's ReadyNow! technology is built into Zing, allowing Java applications to start fast and stay fast, even across reboots. For production instance control and issue resolution, Zing adds Zing Vision, a zero-overhead, always-on management tool for production-time diagnostics. Zing readily integrates with industry-leading third party application performance management suites.

Zing is simple to install and requires no coding changes to existing applications. You don't even need to recompile. Because Zing has been optimized for today's servers, configuration and setup are typically reduced to just a few parameters, instead of the myriad of JVM tuning flags necessary to reach peak performance that characterize many Java-based production environments. Simply point your application or startup scripts to Zing, and you're running on the most robust, scalable JVM with the fastest time-to-market for any business application.

## Zing Features

- Ships as part of a complete JDK – easy to get started, easy to use
- Zing distributions support environments using Java SE LTS releases, e.g. 8, 7 or 6
- Feature preview builds available for Java SE 11
- Eliminates the stalls, jitter, and latency outliers caused by Java Garbage Collection
- Unique garbage collector technology: Azul C4 (Continuously Concurrent Compacting Collector)
- Optimized for 64-bit Linux on x86
- Choose the Java memory configuration and heap size you need, from 1GB to 8TB
- Optimistic Thread Concurrency for CPUs supporting hardware transactional memory
- Continuous delivery of new Linux kernel support
- Quarterly and as-needed security updates
- Azul-optimized Falcon server-tier JIT compiler
- ReadyNow! technology solves Java "warm-up" problems
- Improves operating metrics even when used in zero-GC environments and frameworks
- Supports JVM languages beyond Java (e.g. Scala, jRuby and others)
- Try Zing hassle-free – download from [www.azul.com/zingtrial](http://www.azul.com/zingtrial)

## 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 or more
- 2 cores or more

## Supported Operating Systems

- 64 bit Linux (Intel x86-64)
- Red Hat Enterprise Linux/CentOS 7.0 or later
- Red Hat Enterprise Linux/CentOS 6.0 or later
- CoreOS 4.13.16
- SUSE Linux Enterprise Server 12 SP1, 12 SP2, 11 SP4
- Oracle Linux 5.x or 6.x (kernel specific)
- Ubuntu 16.04 LTS
- Ubuntu 14.04 LTS
- Ubuntu 12.04 LTS
- Debian Wheezy, Stretch and Jessie
- Amazon Linux
- Other Linux distributions supported via Dynamic Kernel Module System

## Supported Java Versions

- Java 8, 7 and 6
- Java 11 feature preview

## Zing Management and Diagnostic Tools:

### Zing Vision

- Always-on, zero-overhead visibility into production workloads
- Fast diagnosis of issues when and where they happen
- Both granular and broad visibility into the details of instance level performance
- Thread-level analysis
- Code hotspot detection
- Lock contention detection
- Method profiling
- Runtime memory leak detection

### Zing Robot

- Automate collection of Zing Vision data

### Zing MxBean

- Standard and extended C4-aware JMX metrics

## Selected Zing use cases:

- Online retail
- SaaS deployments
- Cloud-based solutions (AWS, Azure, Google) plus private and hybrid Clouds
- Big Data
- Risk analytics
- Web-scale IT
- Advertising networks
- Large-scale online and social gaming
- Low-latency trading
- Complex event processing
- Real-time messaging
- In-memory data grids, including Hazelcast, JBoss Data Grid and GridGain
- Search – Elasticsearch, Apache Lucene & Solr
- Java-based infrastructure: Cassandra, Kafka, Zookeeper, Hadoop, Spark, Neo4J and many more

Contact Azul Systems:  
385 Moffett Park Drive  
Suite 115  
Sunnyvale, CA  
94089 USA  
T + 1.650.230.6500  
F + 1.650.230.6600  
[azul.com/products/zing](http://azul.com/products/zing)

## Monotype™

Copyright © 2018 Azul Systems, Inc. 385 Moffett Park Drive Suite 115, Sunnyvale, CA 94089-1306. All rights reserved. Azul Systems, the Azul Systems logo, Zulu and Zing are registered trademarks, and ReadyNow! is a trademark of Azul Systems Inc. Java and OpenJDK are trademarks of Oracle Corporation and/or its affiliated companies in the United States and other countries. Monotype is a trademark of Monotype Imaging Inc. registered in the United States Patent and Trademark Office and may be registered in certain other jurisdictions. The Monotype logo is a trademark of Monotype Imaging Inc. and may be registered in certain jurisdictions. Other marks are the property of their respective owners and are used here only for identification purposes. Products and specifications discussed in this document may reflect future versions and are subject to change by Azul Systems without notice.

10-18v1