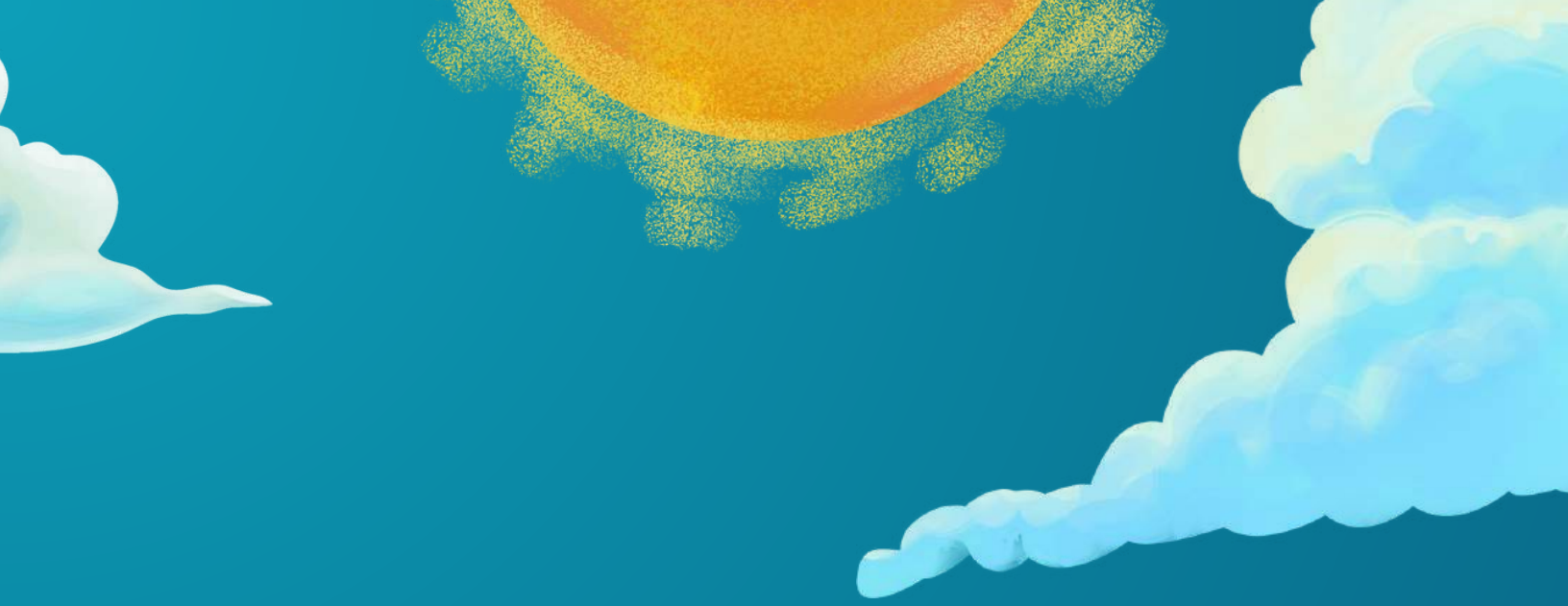


azul

UTILIZING A HIGH-PERFORMANCE JAVA PLATFORM

INTRODUCTION

The following information is about Azul Platform Prime.

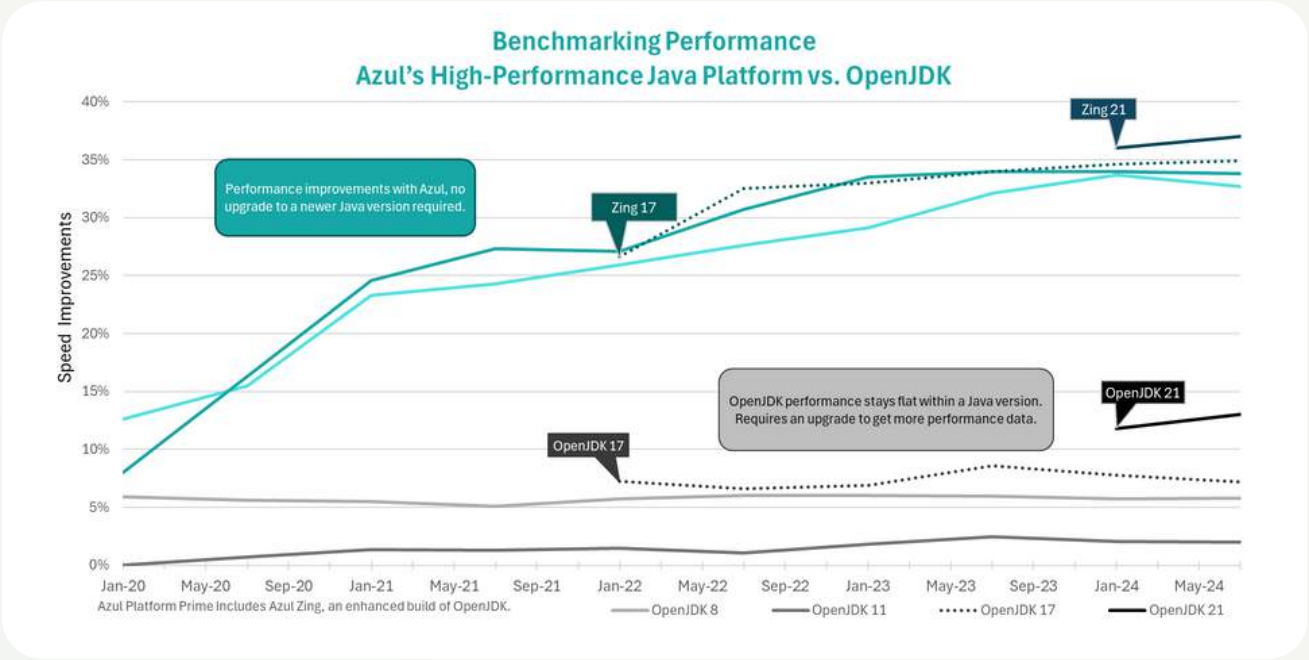
This brief report summarizes initial cost-savings methods, challenges organizations face with the lift-and-shift approach, and various methods to optimize cloud performance. It also highlights how Azul's high-performance Java Platform can reduce costs, save time, and enhance value to your end customers.

1. Introduction
   A. **Savings methods:** Include cheaper storage, discounts, turn off instances, and cheaper compute platforms.
   B. **Lift and shift to cloud** (without app re-architecture for cost optimization) usually results in higher costs.
   C. **Cloud Optimization focuses on:** (1) Right-sizing (only provision resources you need), (2) Cloud-native architecture (smaller stateless microservices), and (3) Elastic scaling (add/remove servers as traffic changes)

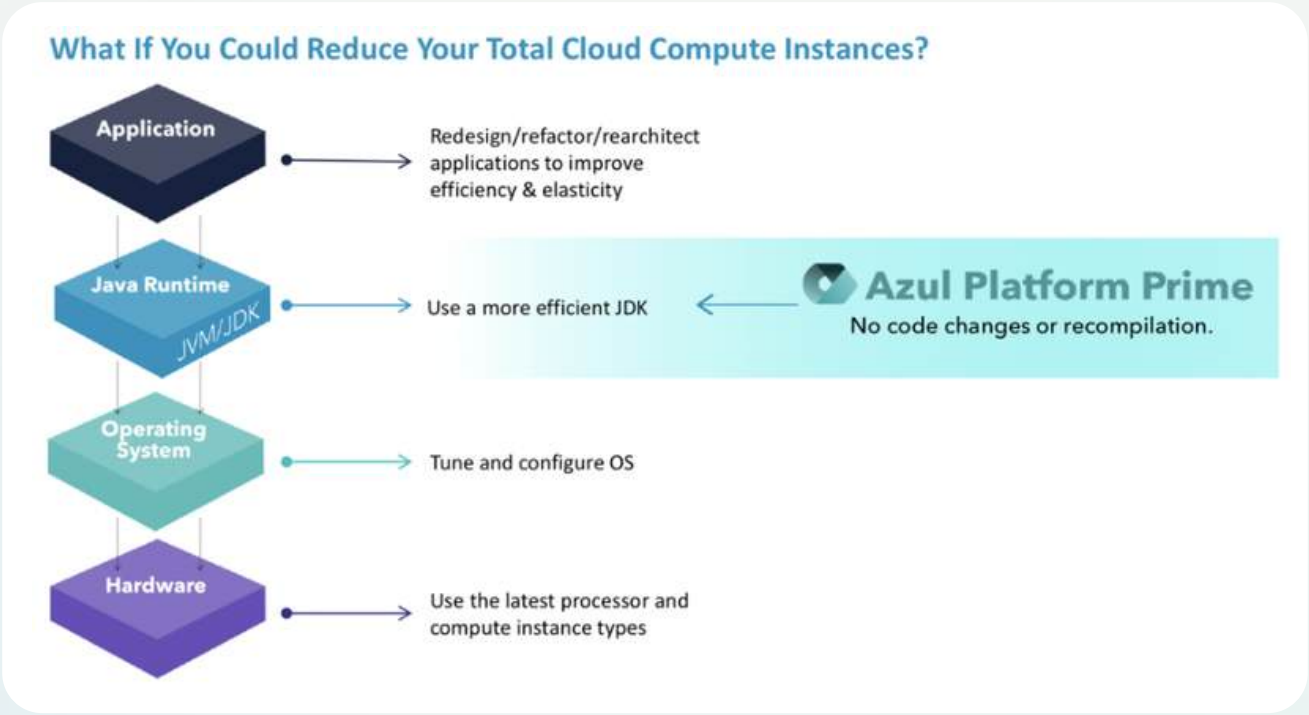
2. High-Performance Java platform: Definition and value
   A. **Vanilla OpenJDK Builds** are unmodified from the OpenJDK source code and are easy to migrate between. Examples include Oracle Java SE and Azul Zulu Builds of OpenJDK.
   B. **High-performance Java platforms** deliver enhanced speedy and efficienc better performance with performant versions of OpenJDK components (like Azul Zing Builds of OpenJDK) or different implementations of the JDK (like Oracle GraalVM and IBM Semeru Runtime)
      - **Figure 12** provides more details:

VANILLA OPENJDK VS. A HIGH-PERFORMANCE JAVA PLATFORM	
Vanilla OpenJDK Builds. These JDKs are unmodified builds of the main OpenJDK source code. Their entire purpose in life is to behave exactly the same as any other build of OpenJDK, to ensure easy migration from one JDK to another.	• Oracle Java SE • Azul Zulu Builds of OpenJDK • Amazon Corretto • Microsoft Build of OpenJDK • Bellsoft Liberica • Red Hat build of OpenJDK • Eclipse Temurin
High-Performance Java Platforms. These JDKs differentiate themselves from vanilla builds of OpenJDK by providing better performance. They either build on OpenJDK by replacing and adding more performant versions of standard OpenJDK components (like Azul Zing Builds of OpenJDK) or are completely different implementations of the JDK (like GraalVM or IBM Semuru).	• Azul Zing Builds of OpenJDK • Oracle GraalVM • IBM Semeru Runtime

   C. **Azul's performance benchmark comparing Azul Zing Builds of OpenJDK vs. vanilla JDK** demonstrates that Zing consistently delivers significantly better performance across all Java releases. Also, Zing's performance improves with each release along with security and patch updates, whereas OpenJDK does not improve between Java versions. Zing 21 is 3x performant as OpenJDK 21.
      - **Figure 13** provides more data:



3. **Maximizing efficiency in cloud-native Java applications**
   - A. **Most people don't know** how important the JDK is for cost optimization. In many cases, developers can cut their costs in half.
   - B. **The Java Runtime** (JVM/JDK) is where Azul Platform Prime helps you reduce your total cloud compute instances/cost.
     - **The other areas** for Cost Optimization include Application Redesign (elasticity), Operating System (tune & configure), and Hardware (latest processor).
     - **Figure 14** provides more information and context:



C. JDK Problems:

- **Right-sizing:** JDKs have garbage collection (GC) issues and deoptimization storms (repeated optimization/deoptimization leads to performance issues).
- **Elastically scaling loads:** Framework initialization (warm-up) times.
- **Rearchitecting to cloud-native:** Cheap resources are bad for latency requirements, there are issues with scaling microservices, and so on.

D. Azul's high-performance Java platform:

- **Right-sizing:** Mitigate GC and deoptimization outliers, increasing CPU utilization.
- **Garbage Collection (GC):** Pauseless GC cleans unused memory without pausing thread execution.
- **Elastic Scaling:** Minimize startup and warmup by front-loading optimizations.
- **Cloud-Native Architecture:** Azul increases transactions per server to reduce total servers needed to manage traffic.

4. Maximize efficiency in legacy lift-and-shift

A. **Problem:** Most apps are monoliths and the developers are long-gone, so the cost is often higher to (1) lift and shift them to the cloud (using too many resources) or (2) to then rearchitect them in the cloud for elasticity and microservices.

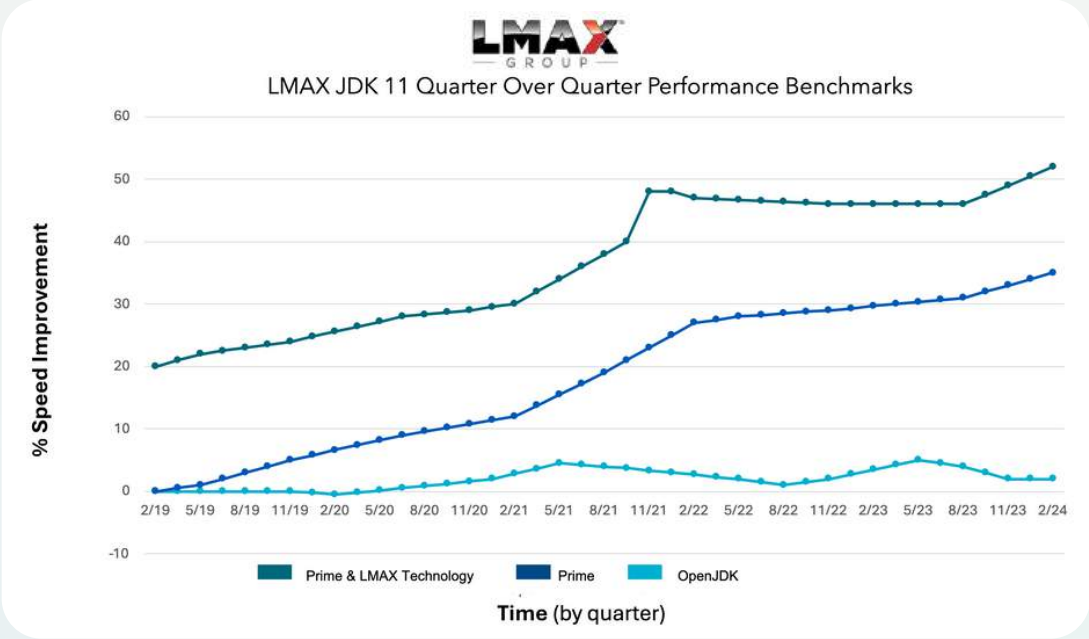
- It's often less expensive to leave the apps on a server.

B. **Solution:** A high-performance Java platform (Azul Platform Prime) optimizes the lift and shift (faster code, faster warmup, and pauseless GC) without changing a line of code or upgrading to a new version of Java.

5. Examples of customer success on Azul Platform Prime

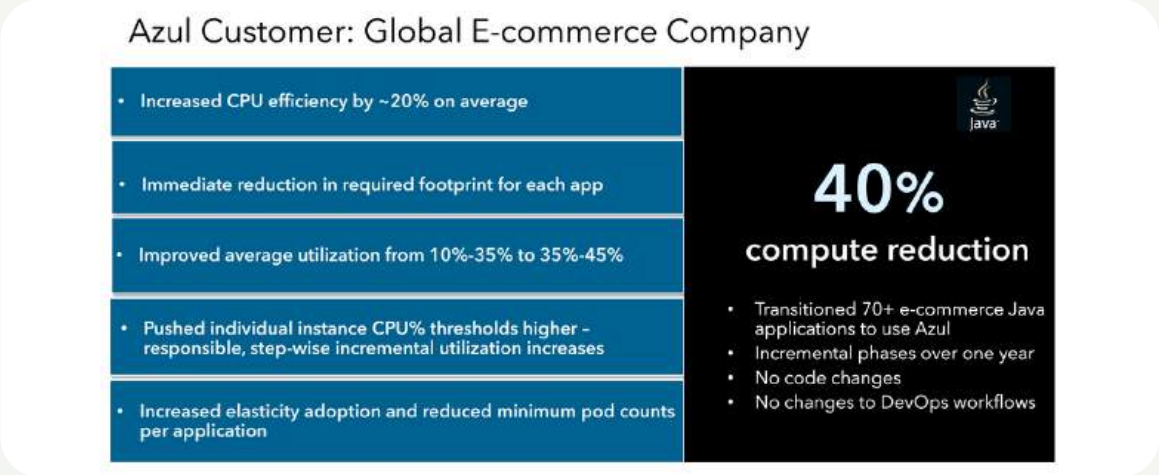
A. **LMAX is 48% faster** with Azul Zing. This provides a better fill ratio, fewer rejected orders, and lowered opportunity cost.

- **Figure 15** provides more data (the two darker lines show Prime):

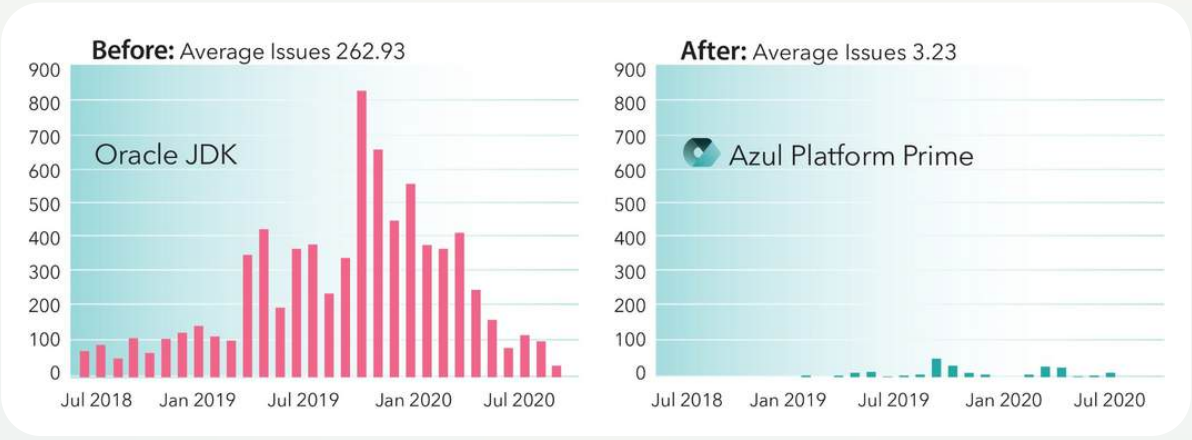


B. Global Retailer reduced compute by 40% and transitioned over 70 Java apps to Azul.

- **Figure 16** provides more details:

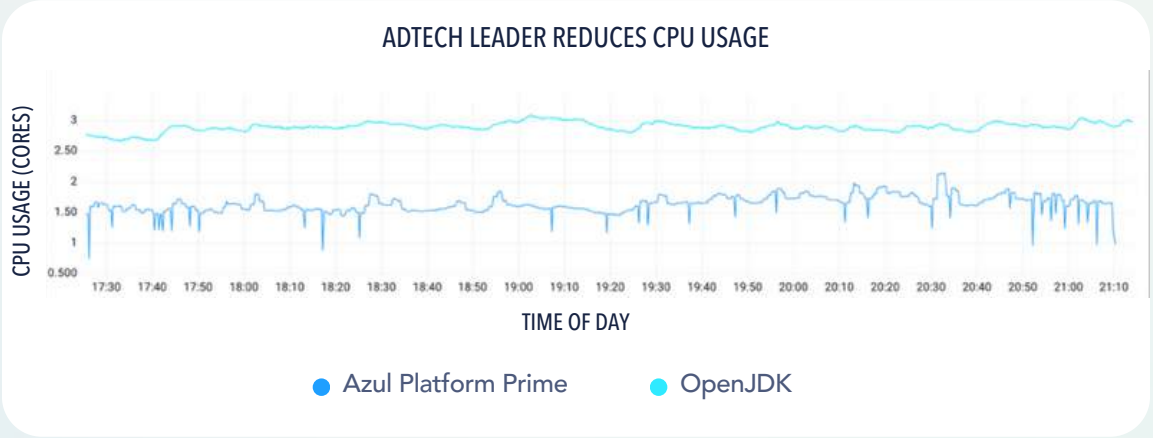


- **Eliminated 95%** of operational issues
- **Figure 17** provides more comparative information about the operational issues:



C. AdTech Company flattened 95 percentile of SLA.

- **Reduced database server footprint** by 50% and front-end server footprint by 30% (removed hundreds of servers).
- **Figure 18** shows the drop in CPU usage:



azul

CHECK OUT THE FULL GUIDE

Get the Full Guide

GETTING STARTED WITH AZUL'S HIGH-PERFORMANCE JAVA PLATFORM

Azul's high-performance Java platform can help your organization maximize efficiencies and minimize costs by delivering pauseless garbage collection, faster code, and faster warmup times. Azul Platform Prime is TCK-verified using Oracle's Technology Compatibility Kit, a test suite that verifies that the Java implementation is compatible with the Java SE specifications, and includes Azul Zing Builds of OpenJDK, enhanced for superior performance, consistency and efficiency.

Download Azul Platform Prime for Free

LMAX Group is a global financial technology company that is shaping the future of the global FX and the crypto marketplace. LMAX Group uses Azul Platform Prime to increase market order fill ratios and processes more than 700,000 risk calculations/second.

Read the Case Study

Learn how your organization can improve performance while reducing infrastructure costs by 20%+.

Speak with a Java Expert Today

Azul Platform Prime: Highest-rated New Product or Service of the Year

Azul Platform Prime Gold Award: Software Development Solution of the Year

Azul Platform Prime Gold Award: Cloud Application/Service of the Year