AZUL
S Y S T E M S®

Zing Quick Start

A Professional Services package to get your Zing deployment started right

ENGAGEMENT PROFILE

In partnership with Azul, your team delivered a successful Zing® pilot with one of your application systems. The knowledge gained during the pilot along with configuration options should now be migrated to your staging and production environments. Your operations team needs to become familiar with running an application system using Zing; and your architects and development team need to increase their knowledge of Zing:

· Analyzing, Configuring and Validating the deployment environment
· Installing Zing and continuing knowledge transfer
· Leveraging Best Practices regarding testing and deploying Zing
· Learning about Zing's monitoring and profiling tools and how to use them throughout the application's life cycle
· Getting support and help from Azul, should you ever need it

Azul is offering a five day package of consulting and training delivered by our Zing Professional Services team to assist customers in getting the most value from their Zing investment. The package includes a mix of both off-site and on-site consulting and training.

A Zing Consultant is engaged to work with your operations and development team to deploy your Java application on Zing. Working together using best practices, the application will be configured and deployed to achieve the best results.

Azul recognizes that any change to a production environment requires careful planning and change control; therefore the first part of the package is delivered remotely, which allows your operations team to follow your existing processes.

HOW DOES IT WORK?

Days 1 & 2   The initial two days are to define requirements and evaluate and configure a suitable server for the application. These tasks are performed remotely over a number of days depending on your staff availability and the duration of your change control processes. As a key first step, the Consultant assists your operations team to run Azul tools to examine the hardware and operating system environment of the server platform on which you intend to deploy the application with Zing. Next, the Consultant holds a questionnaire-based workshop with your development team to understand the latency requirements of the application and to document those details. Subsequently, the Consultant evaluates the outputs from the Azul tools and the questionnaire; and then reviews them with your team. The Consultant recommends a set of best practices configuration changes; and your operations team follows your usual change control processes to implement the recommendations and install and license the Zing software. The Azul tools are run again to evaluate how the reconfigured environment is behaving. This might require one or more steps to be repeated.

Day 3   A one day, structured training course designed to give your team the information they need to successfully test, monitor, deploy and operate your Java application on Zing. (See left sidebar on page 2 for more details.)

Days 4 & 5   On-site, hands-on consulting, working alongside your team's developers and architects as required answering questions and delivering knowledge transfer relevant to the application. The goal is to enhance your team's understanding of Zing and its tools, and how they can be put to practical use within your environment.

Zing Quick Start Training

WHAT YOU WILL LEARN

How to successfully test and deploy your Java application on Zing in order to reap its unique benefits.

The Azul Systems Consultant will teach you what is different about Zing compared with other JVMs; how to use the tools provided; best practices for configuring your environment; and how to deliver even more predictable, more consistent and lower '100% worst case' latency outliers than Zing's already remarkable out of the box performance.

Introduction to Zing

· How Zing is different from other JVMs.
· Understanding the Azul Systems C4 Garbage Collector.
· Testing practices and strategies with Zing.
· System Requirements and installing Zing.
· How to use jHiccup Idle Test to visualize the jitter in your OS environment.
· Using Azul Inspector to examine your server's environment for configuration issues.

Using Zing

· Initial JVM configuration best practices.
· How to run tests with Zing so that the benefits that Zing provides are readily observable.
· Detecting coordinated omission in testing systems and using HdrHistogram and jHiccup to combat it.
· Monitoring tests with jHiccup and Azul Inspector Monitor Mode.
· Compilers, optimization, deoptimization – Zing Directives for controlling compilation and optimizations.
· How to work in partnership with Azul Support to track down performance, configuration and code issues.

Introduction to Zing Tools

· Azul supplied tools: Zing PS; ZVision; ZVRobot; jHiccup; gcLogAnalyser; ZRC; Azul Inspector Monitor Mode.
· Using Third-party tools and agents with Zing.
· The richness of system and application performance information available from Zing and how to use the supplied tools to measure, monitor, troubleshoot and tune the system to run your application.
· How to use Zing Vision (ZVision) to dynamically view ARTA information.
· Using Zing Vision Robot (ZVRobot) to collect and save selected ARTA information.
· How to visualize performance with jHiccup and gcLogAnalyser.

Deploying an application on Zing

· How to achieve a reduction in the number and duration of statistical outliers in your application system, together with consistent latencies.
· Best Practices for system configuration and system tuning; accumulated from real world experience with Zing customers.
· Using a Deployment Checklist for the Best Practices covered.

Azul Real Time Analysis (ARTA)

The no-overhead, built-in Zing Java system interface for application profiling and code examination – and how to use it in test, QA and Production.

How to dynamically view the ARTA information; allowing you to see snapshots of the operational behavior of your code threads in the Zing VM: see hot methods, drill down into hot methods' calling stacks, and view the assembly code; see stack traces for your threads and lock contention; profile live objects, see total size in bytes, counts and growth rate; view system threads as well as application threads, filter by state.

How to collect selected ARTA information over a period of time and store it in a directory structure for browsing offline.

How to use jHiccup and the gcLogAnalyser tools to visualize the performance of your application and the runtime environment. These tools allow you to make decisions that will improve the performance of your application as well as pinpoint and troubleshoot issues.

What becomes relevant to latency consistency once the JVM no longer pauses for Garbage Collection and understanding what else needs to be changed when tuning for consistent latency.

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/zing