

Introduction to Java Performance Tuning

Course Summary

Description

The [Introduction to Java Performance Tuning](#) training course examines the philosophies, tools, and processes required to begin tuning a Java application.

Assuming most students have not experimented with performance tuning, the JVM Tuning training course begins with an introduction of performance tuning concepts, exploring what is and is not considered performance tuning.

To ensure a successful learning of tuning concepts, the Introduction to Java Performance tuning course also examines the Java platform and the Java Virtual Machine architecture. With the appropriate foundation laid, the course examines common tuning tools and their applications.

Objectives

At the end of this course, students will be able to:

- Understand the performance tuning process
- Establish performance characteristics and measurement strategies
- Identify and determine potential causes for hot spots, areas of latency, and bottlenecks in your application
- Design a performance tuning strategy for your application based on identified underperforming and established characteristics
- Work with common tuning tools such as profilers, pstat, top, visual vm, etc.
- Tune the JVM, including GC algorithms, memory profiles, etc.

Topics

- Introduction to Performance Tuning
- Java Tuning Process
- Garbage Collection Tuning

Prerequisites

ProTech's [Introduction to Linux System Administration](#) course.

Duration

Three days

Introduction to Java Performance Tuning

Course Outline

- I. Introduction to Performance Tuning**
 - A. Introduction to Performance Tuning
 - 1. Background on tuning
 - 2. Why do we need to tune
 - 3. What are symptoms of a system that needs tuning
 - 4. What outcomes should I expect?
 - B. Java Virtual Machine Architecture
 - 1. Overview of Java as a platform
 - 2. Java Virtual Machine Architecture
 - 3. Key Components of the Java Virtual Machine
 - 4. Byte-code verifiers, class loaders, and security
 - 5. Memory heaps and stacks
 - 6. Garbage collection
 - C. Java Tuning Tools
 - 1. Basic tools
 - 2. Timing tools
 - 3. Load Testing Tools and Techniques
 - 4. JVM Profiling Tools (Profilers)
 - 5. Memory Management and Monitoring
 - 6. Operating System and Network Monitoring tools
- II. Java Tuning Process**
 - A. Java Tuning Process
 - 1. Java Runtime Environment and System constraints
 - 2. Calculating & Measuring JVM / JRE overheads
 - 3. Identifying "startup" components of JVM / JRE
 - 4. Calculating and Measuring Hardware and OS Constraints
 - B. Performance planning
 - 1. Defining performance specifications, variations, and objectives
 - C. Defining and Capturing a benchmark (a baseline)
 - D. Tuning from the benchmark to the objectives
 - E. Measuring improvements with Profiling and Monitoring tools
- III. Garbage Collection Tuning**
 - A. Fundamentals of Garbage Collection and Heap Tuning
 - 1. Understanding JVM memory and lifecycles
 - 2. Following an object through a lifecycle
 - 3. GC algorithms: Which one, when – tuning GC algorithms
 - 4. JVM Command line switches
 - 5. Measuring improvements
 - B. Tuning Best Practices
 - 1. Overview of the tuning methodology
 - 2. Understanding the 80/20 rule
 - 3. Can you be smarter than the JIT?
 - 4. Getting big gains quick
 - 5. Understanding the overhead of Logging, Exceptions, Strings, and IO
 - C. Tuning Java EE Applications
 - 1. Review of Java EE and Java EE Application Server platforms
 - 2. Identifying and tuning common bottlenecks in Java EE applications
 - 3. Identifying and tuning resource pools, external resources, and dependencies
 - D. Best practices relating to EE tuning