"Java Performance Tuning"

Duration: 5 half-days

Pre-requisite: Minimum **ONE** year of java experience

Course Objectives:

Understand JVM Architecture

Understand Garbage Collection

Understand Heap memory

Understand Thread Dump

Working with Collections

Working with Java References

Working with String effectively, String Deduplication

Working with Threads, Locks

Working with Reactive Streams

Working with JMC, Jstack, JMap tools

Course Outline:

Day-1

1. Performance Overview

Performance Principles

Common Performance Problems

Performance Methodology

Development and Performance

2. JVM Architecture

Class Loader Subsystem

Runtime Data Area

Execution Engine

JIT Compiler

Lab : Generating JIT bytecode & assembly code, JIT Watch

3. Memory Management & Garbage Collection Concepts

HotSpot Generations (young, old and permanent)

Garbage Collection Types

- ✓ Serial GC
- ✓ Parallel GC
- ✓ Concurrent Mark-Sweep GC
- ✓ Garbage First GC (G1)

Ergonomics -- Automatic Selections and Behavior Tuning

Lab : Analyzing GC Logs manually (all GC Algorithms from Parallel till G1) & GC Viewer demo

DAY-2

4. Key Options Related to Heap & Garbage Collection

- ✓ Heap and Generation Sizes

  -Xms,-Xmx,-XX:NewSize,-XX:MaxNewSize,-XX:ThreadStackSize,

  -XX:SurvivorRatio,-XX:MetaspaceSize,-XX:MaxMetaspaceSize,

  -XX:ReservedCodeCacheSize,-XX:-UseAdaptiveSizePolicy

- ✓ Garbage Collector Options

  –XX:+UseSerialGC,–XX:+UseParallelGC,

  –XX:+UseConcMarkSweepGC,-XX:+UseG1GC

- ✓ Garbage Collector Statistics

  –XX:+PrintGC,–XX:+PrintGCDetails,–XX:+PrintGCTimeStamps

  -XX:+HeapDumpOnOutOfMemoryError,-XX:+PrintCommandLineFlags

- ✓ Other GC options

  –XX:ParallelGCThreads,–XX:MaxGCPauseMillis,–XX:GCTimeRatio

  -XX:ConcGCThreads,-XX:InitiatingHeapOccupancyPercent

  -XX:MaxGCPauseMillis,-XX:G1HeapRegionSize,

  -XX:G1HeapRegionSize,-XX:G1ReservePercent,

  -XX:+AlwaysPreTouch,-XX:+UnlockDiagnosticVMOptions

  XX:+UseBiasedLocking, -XX:+UseStringDeduplication

Lab : JVM process memory monitor and optimization

5. Code Optimization, Monitor & Analyze Java Heap, Threads & CPU Usage

jmap, jhat, jcmd

jmc (JFR), jProfiler

Eclipse Memory Analyser

Lab: Java code execution & fixing the problems

Day-3

6. Performance Tuning at the Language Level

The best practices for Object Allocation

String-efficient Java Applications

Collection Classes

Using Threads, Locks & Java References types

Understanding the overhead of Logging, Exceptions, Strings, and IO

Lab : Java code optimization (String, Collection, Thread & Locks)

Day-4

7. Managing the application at Runtime

JMX Architecture

Core Service MBeans

Custom MBeans

Lab : Exposing Custom MBeans and run-time management

8. Performance Tuning at web application Level

Http Sessions, NIO Connector & Thread pool

Servlet Filters. listeners optimization

Datasource Connection Pooling

Lab : Datasource, Connector optimization

Day-5

9. Reactive Streams - non-blocking asynchronous stream processing

Reactive Streams Specification

Flow, Publisher, Subscriber

Subscription & Processor

10. Best Practices – Spring & JPA
Spring – Bean Post Processor
JPA – Fetching Strategies
Spring Boot Actuator