

Formation Optimisation et Profiling des applications

JAVA avec JProfiler

Description de la formation Java optimisation

Cette formation Java Profiling et Optimisation avec JProfiler, est destinée à des développeurs ayant une bonne connaissance du langage Java. C'est une personnalisation de la formation standard JPOPJ en effectuant la plupart des ateliers avec l'outil JProfiler et en faisant un focus sur les facilités apportés par l'outil.

Objectifs

Objectifs Opérationnels :

Savoir monitorer et profiler les applications Java tout en explicitant les techniques permettant de les optimiser

Objectifs pédagogiques :

Concrètement, à l'issue de cette formation Java optimisation vous aurez acquis les compétences nécessaires pour :

- Maitriser les concepts fondamentaux du langage (gestion de la mémoire, mécanisme d'exécution, chargement de classes...)
- Connaître les impacts de l'algorithmie sur les performances
- Maitriser les techniques classiques d'optimisation
- Connaître les particularités des services back-end écrits en Java

À qui s'adresse cette formation ?

Public :

Cette formation Java optimisation s'adresse à tout développeur, chef de projet ou architecte impliqués dans la réalisation d'applications Java et plus particulièrement J2EE.

Prérequis :

Connaissance de Java

Contenu du cours Java optimisation

Introduction

L'optimisation de performance :

Méthodologie, recommandations, benchmarking, métriques observés

Concepts fondamentaux de Java :

JVM, Zones mémoires et GC, Class Loader, Multithreading, JIT, JVMTI, Outils de base

Les causes de mauvaises performances

CPU, mémoire, IO

Le ramasse miettes

Introduction

Pools de la HotSpot

Les différents algorithmes

Tuning du garbage collector

Atelier

Observation des collectes avec JProfiler, tuning des zones mémoire et de l'algorithme

Boîtes à outils

Introduction et complémentarité des outils

Outils systèmes

JMX et les Mxbeans de la JVM

Agents JVMTI, exemples

Outils fournis par la distribution

Commandes en ligne Java

jVisualVM et jmc

JProfiler

Distributions, Facilités proposés

Temps d'exécution : Enregistrements des appels, CPU profiling,

Mémoire : Utilisation, collectes

Multithreading :

Serveurs back-ends, JavaEE

Atelier

Monitoring via des consoles JMX, Exemple d'un agent JVMTI

Ateliers avec les commandes en lignes

Utilisation de JProfiler pour différents cas de détection de mauvais fonctionnement

Optimisation de code Java

Généralités

Limitation d'instances temporaires, boucles et récursivité, les streams

Utilisation des chaînes de caractères, switch

Exceptions et stacktrace

Gestion des I/O, Bufferisation, le package java.nio, gestion des traces

Les collections et tableaux

Choisir les bonnes implémentations en fonction de l'algorithme

Collection synchronisées ou non

Coût des allocations/désallocations

Réutilisation d'instance

Pattern pool, singleton, Thread Local

Les Weak references

Applications multithreadées

Cas d'usage des threads, les problèmes de synchronisation, mécanismes de base

Le package java.util.concurrent

Utilisation de pool de threads

Java8 et l'asynchronisme

Reactive programming

Atelier

Optimisation d'application en utilisant les techniques présentées

Services Back-end

Particularités JavaEE et services back-end

Différents Pool de l'architecture, les caches, scalabilité

Outils de simulation de charge

Intégration des serveurs JProfiler

Persistance et JPA

Pools de connexions

Optimisation du schéma

Caches

Métier

Modèle stateless/stateful

Transactions

Http et REST

Sérialisation/désérialisation

Optimisation des transferts

Ateliers

Diagnostic de problèmes sur une application web complète, sur une API Rest

Travaux Pratiques

Java 6/7 - Utilisation de Netbeans, Netbeans Profiler ou JProfiler