



## Programme détaillé formation 'Performance JAVA'- 2j

**Objectifs** : fournir connaissances théoriques et pratiques permettant la et l'optimisation des performances d'une application java

**Audience** : Développeurs, chefs de projets, Architectes, Administrateurs .

**Prérequis** : pratique langage de programmation JAVA

**Moyens pédagogiques** : 1 ordinateur/stagiaire. Supports cours. Travaux pratiques. Vidéoprojecteur. Tests

**Durée** : 2 jours (total 14 h)

### Vous allez apprendre à :

- ▶ Mesurer les performances d'une application JAVA / J2EE
- ▶ Réaliser l'optimisation de la performance
- ▶ Mettre en oeuvre méthodologie de tuning

### Programme détaillé cours :

#### Rappels J2EE

- ▶ Architecture et impact sur performance
- ▶ Services : JMX, JNDI, JTA, JDBC, Web, EJB
- ▶ Livrables : WAR, EAR, Jar Ejb, Jar client, RAR
- ▶ Fichiers web.xml, application.xml, ejb-jar.xml, application-client.xml, ra.xml
- ▶ Chargeur de classes (ClassLoader)

#### Introduction Performance Java

- ▶ Définir la performance d'une application
- ▶ Service Level Agreement (SLA)
- ▶ Critères de performance
- ▶ Temps de réponse
- ▶ Transactions par secondes
- ▶ Difficultés associés à l'optimisation performances

#### Méthodologie Tuning

- ▶ Tuning Matériel
- ▶ Tuning Os
- ▶ Tuning JVM
- ▶ Tuning Serveur d'application
- ▶ Tuning Applicatif

#### Monitoring avec JMX

- ▶ Spécification JMX
- ▶ Qu'est ce qu'un Mbean ?
- ▶ Couche Instrumentation
- ▶ Couche Agent (MBean Server)
- ▶ Couche Client : connecteurs (RMI, HTTP, SNMP)
- ▶ Monitoring local / distant

- ▶ Fichier jmxremote.password
- ▶ Fichier jmxremote.access
- ▶ JMX pour l'administrateur / le développeur
- ▶ Ouverture explicite d'un port JMX par l'administrateur
- ▶ limiter l'accès distant JMX à une seule machine
- ▶ propriété système : -Dcom.sun.management.jmxremote
- ▶ propriété système : -Dcom.sun.management.jmxremote.authenticate
- ▶ propriété système : -Dcom.sun.management.jmxremote.ssl
- ▶ propriété système : -Dcom.sun.management.jmxremote.authenticate
- ▶ propriété système : -Dcom.sun.management.jmxremote.password.file
- ▶ propriété système : -Dcom.sun.management.jmxremote.access.file
- ▶ Adaptation à Tomcat, Jboss, GlassFish
- ▶ Création d'un Mbean par le développeur

#### **Outil : jvisualvm**

- ▶ Jvisualvm dans JDK / en standalone
- ▶ Plugins MBeans et VisualGC
- ▶ Connection locale / distante
- ▶ Auto découverte Processus java (jps)
- ▶ Première Analyse rapide : cpu, heap, threads
- ▶ Instrumentation : Profiling Mémoire, profiling CPU
- ▶ Identifier fuites de mémoire (objets occupant le plus de place)
- ▶ Identifier goulots d'étranglement (méthode occupant le plus de cycle cpu)
- ▶ Snapshots, Dump (HeapDump, ThreadDump)
- ▶ Mise en oeuvre avec JStatd + fichier policy
- ▶ fichiers hprof

#### **Mesure & Instrumentation**

- ▶ Mesure manuelle : System.currentTimeMillis()
- ▶ Mesures du Garbage Collector
- ▶ -verbosegc , GCViewer, visualGC
- ▶ Mesures de temps de d'appels de méthode
- ▶ Profiling de création d'objets
- ▶ Mesure de l'allocation mémoire (young, old)
- ▶ Mesure communication client / serveur
- ▶ -Djava.rmi.server.logCalls=true
- ▶ CheckList outils performance bas niveau
- ▶ Monitoring JMX : comprendre les MBeans
- ▶ Jconsole, JVisualVM

#### **Tuning applicatif**

- ▶ Création d'Objets
- ▶ Chaînes (Strings)
- ▶ Collections (tableau, vecteurs)
- ▶ Boucles, Switch, Récursivité
- ▶ Entrées/sorties fichier, réseau
- ▶ Multithread
- ▶ complexité
- ▶ Utilisation Patterns
- ▶ Logs

#### **Tuning Serveur d'application**

- ▶ Tuning Pool de connexion
- ▶ Tuning Pool de thread
- ▶ Tuning services J2EE (HTTP(S), JDBC, JMS, JPA)

#### **Tuning matériel**

- ▶ Tuning carte réseau
- ▶ Tuning mémoire
- ▶ Tuning cpu

#### **Tuning conteneur Web**

- ▶ Tuning JSP
- ▶ Tuning Servlets

## **Tuning conteneur Ejb**

- ▶ Tuning EJB

## **Tuning JVM**

- ▶ Comprendre le fonctionnement mémoire
- ▶ Young, old, PermGen, Survivor space
- ▶ Choix -Xmx, -Xms en fonction RAM
- ▶ Choix PermSize, MaxPermSize
- ▶ Choix -client / -server
- ▶ Ratios Young/old
- ▶ Paramètres GC
- ▶ Simuler et analyser fuites mémoire
- ▶ **Outil VisualGC**
- ▶ Exploitation des résultats