



Le clustering d'applications Java avec Terracotta

Référence : TERRACOTTA-01-02

Durée : 2 jour(s)

Présentation

Cette formation détaille l'utilisation de Terracotta, solution de clusterisation d'applications agissant directement sur les machines virtuelles Java. Vous apprendrez à rendre vos applications Java hautement disponibles et déployables à grande échelle sans effort particulier.

La formation couvre les différents aspects de Terracotta, dont le fait de s'appuyer directement sur les JVM, et son positionnement par rapport aux technologies de clusterisation existantes. Nous couvrirons les cas d'utilisation suivants : sessions http, caches distribués, POJO clusterisés, singleton-Beans/Context Events Spring, coordination de JVM et la clusterisation d'autres frameworks de développement.

Objectifs

- Appliquer les cas d'utilisation basiques de la clusterisation
- Utiliser la clusterisation de sessions et de POJO
- Ecrire un TIM
- Coordonner les clusters
- Gérer les clusters
- Configurer le cache distribué
- Maîtriser le tuning pour la performance
- Déployer pour la production

Répartition: 50% Théorie, 50% Pratique

Public: Développeur, Architecte

Pré-requis: Connaissance de Java, et avoir lu **The Definitive Guide to Terracotta: Cluster the JVM for Spring, Hibernate, and POJO Scalability** serait un plus.

Programme

Introduction

- La société Terracotta
- Les services Terracotta
- La technologie phare de Terracotta : les objets partagés et distribués
- Philosophie, Concepts, Architecture
- Installation de Terracotta
- Exemples

- Cache Aside
- Data Grid partitionné
- Scalabilité, haute disponibilité, caractéristiques de chaque configuration
- Exemple d'application

Cas d'utilisation

- Clusterisation de sessions
- Clusterisation de POJO
- Coordination de JVM
- Gestion des clusters
- Topologies des Caches distribués :
- Cache Aside
- Cache Write through
- Cache Write behind
- Data-Grid réparti
- Autres configurations

Tuning

- Tuning de la mémoire
- Garbage Collection
- Gestionnaire de mémoire virtuelle
- Garbage Collection distribué
- Tuning des verrous
- Striping
- Granularité des verrous
- Types de verrous
- Verrouillage pessimiste/optimiste
- Instrumentalisation
- tc-config.xml , tc.properties
- Console d'administration (profilier de verrous, DGC...)
- Visualisation des statistiques
- Options de débogage (tc-config.xml)
- Travaux pratiques : améliorer les performances d'une application

Clusterisation de sessions

- Exemple d'applications clusterisée
- Scalabilité, haute disponibilité, Précision

Clusterisation de POJO

- Concepts
- Exemple de clusterisation de POJO

Déploiement en production

- Options de déploiement
- Analyse des échecs
- Surveiller le cluster en production
- Mise à jour en minimisant les temps d'indisponibilité
- Bonnes pratiques

Le De-Container

- Concept de TIM
- Terracotta TIM pour EHCACHE, Cache Hibernate niveau 2, Wicket, Struts, Quartz, Lucene, Compass
- Exemples avec Quartz et Compass
- Ecrire un TIM

Coordination de clusters

- Wait-Notify distribué
- Invocation de méthodes distribuées
- Master/Worker

Gestion de clusters

- Événements JMX
- Exemple d'application

Cache distribué et configurations de cache

- Cache distribué
- Lecture seule
- Write Through
- Write Behind

OFFERT EN INTER-ENTREPRISE

- ☎ Le petit déjeuner croissants, jus d'orange, café)
- ☎ Le déjeuner
- ☎ Une qualification téléphonique si nécessaire avec l'un de nos consultants

Tel: +33(0)1 45 26 19 15
Fax : +33(0)1 75 43 49 92
Email : training@zenika.com

Auteur du cours



Ari Zilka est le fondateur et le Directeur Technique de Terracotta.

