

Oracle WebLogic Server Deep Dive WebLogic Server Wissenwertes für Fortgeschrittene

**Michael Fuhr
Oracle Deutschland B.V. & Co.KG
Geschäftsstelle Frankfurt**

und

**Stephan Rudolph
rudolph ag
Bern/Schweiz**

Schlüsselworte

Oracle WebLogic Server, WorkManager, WebLogic Scripting Tool, WLST, Deployment Parameter Injektion, Deployment Plan, Deployment-Descriptor, Flexibility, Disaster Recovery, Hochverfügbarkeit, Whole Server Migration, Virtual-/Floating-IP, Java-based, Script-based Nodemanager, Crash Recovery, AutoKillIfFailed, AutoRestart, Consensus Leasing, Split-Brain

Einleitung

Der Oracle WebLogic Server, als Plattform der Fusion Middleware Produkte, verlangt für die Verwendung dieser Infrastrukturkomponente die gleich grossen Kenntnisse, ob sie nun als Grundlage für den Enterprise Manager oder als SOA Plattform eingesetzt wird. Genauso bei den Themen Verfügbarkeit, Skalierbarkeit und Performance ist tiefes Wissen des Produktes wichtig.

In diesem Vortrag werden drei Punkte, die für die Anwendungsentwicklung eine große Rolle spielen, tiefer behandelt und Best Practices zur Verwendung aufgezeigt.

Zum Einen wird es um Injektion zusätzlicher Deployment Descriptor Parameter, also das Anpassen und Modifizieren der Deployment Descriptor zur Laufzeit, gehen. Zum Anderen wird die Verwendung von WorkManagern tiefer beleuchtet, um stabile und optimierte verteilte Anwendungen betreiben zu können. Und zum Schluss betrachten wir noch die „Whole Server Migration“ des Hochverfügbarkeitsmechanismen des WebLogic Servers.

WorkManager, das unbekannte Wesen

Der WebLogic Server verarbeitet alle seine Anfragen über einen „Self Tuning Thread Pool“. Dieser Thread Pool kann aber, durch die Verwendung von WorkManagern, unterteilt und reglementiert werden.

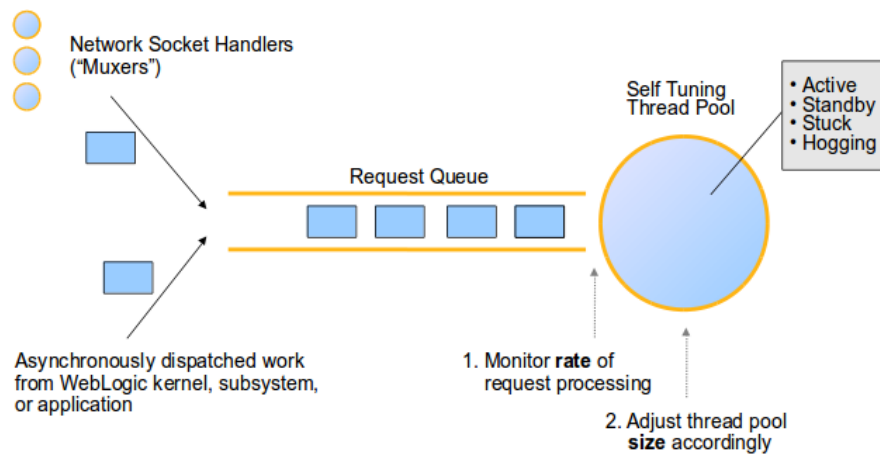


Abb. 1: Self Tuning Thread Pool im WebLogic Server

Mithilfe von WorkManager können nun die ankommenden Anfragen optimal zur Abarbeitung organisiert werden. Eine Priorisierung erfolgt auf Basis der definierten Regeln auf Domain- oder Anwendungs-Ebene (Default WorkManager, Global WorkManager, Application-scoped WorkManager) sowie der Laufzeit-Metriken inklusive der aktuell benötigten Zeit zur Abarbeitung einer Anfrage.

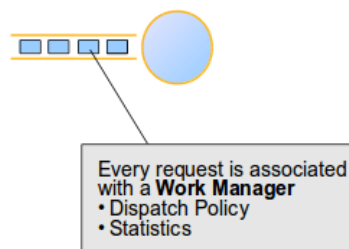


Abb. 1: Request Queue mit einem eignen WorkManager

Dabei spielt der WorkManager auch eine wichtige Rolle bei der Erkennung und Behandlung überlasteter und blockierter WebLogic-Server-Knoten.

Dead-lock Situation können automatisch erkannt werden und verhindert werden, eine strikte Verarbeitungsreihenfolge kann organisiert werden, Parallelisierung kann erreicht werden und ein optimales Ressourcen Management der einzelnen Anwendungen innerhalb einer Domain kann durch die Verwendung von WorkManager einfach konfiguriert werden.

Mit Hilfe des WebLogic Scripting Tools können WorkManager automatisiert konfiguriert und verwaltet werden.

Injektion zusätzlicher Deployment Descriptor Parameter

Die Situation kommt Ihnen möglicherweise bekannt vor. Sie bekommen eine gepackte Java EE Anwendung. Diese enthält eine große Anzahl EJB's, jedoch sind deren Tuning Parameter weder deklariert noch konfiguriert. Sie möchten das Laufzeitverhalten, umgebungsspezifisch, durch gezieltes Tuning einzelner EJB Parameter (max-beans-in-free-pool, etc.) gezielt optimieren. Oder,

Sie bekommen eine gepackte Java EE Anwendung. Diese nutzt Bibliotheken der Laufzeitumgebung, z.B. für das Logging. Sie möchten bestimmte Bibliotheken gezielt z.B. durch neuere Versionen dieser ersetzen.

In jedem der Fälle möchten Sie die Anpassung umgebungsspezifisch (Test, Integration, Produktion), effizient, und ohne Modifikation der gepackten Java EE Anwendung auführen.

Der Weblogic Deployment Plan (Implementierung des JSR-88) unterstützt sowohl das Überschreiben bereits in den Deployment Deskriptoren definierte Property Werte, als auch das Hinzufügen noch nicht deklarerter Properties.

Wir zeigen Ihnen am praktischen Beispiel, wie sie noch nicht deklarierte Deployment Descriptor Parameter Hinzufügen und damit das Laufzeitverhalten Ihrer Anwendung beeinflussen können ohne diese selbst zu modifizieren.

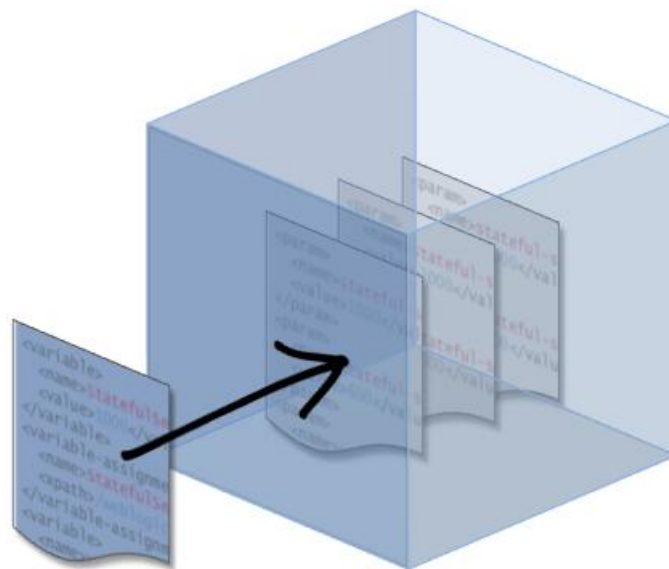


Abb. 1: Injektion zusätzlicher Deployment Descriptor Parameter

Hochverfügbarkeitsmechanismen des Oracle Weblogic Server

Eine wesentliche Funktion des WebLogic Server zur Erzielung von Hochverfügbarkeit stellt die Whole Server Migration dar.

Was gilt es zu beachten? Welche Rolle spielt der Cluster Master? Wie funktioniert Consensus Leasing?

Wir zeigen Ihnen die Konfiguration und die Funktionsweise von Whole Server Migration am praktischen Beispiel für:

- Ausfall einzelner Instanzen (Status FAILED)
- Ausfall einzelner Knoten (z.B. Unterbrechung der Stromversorgung)
- Split-Brain (z.B. Ausfall einer NIC)

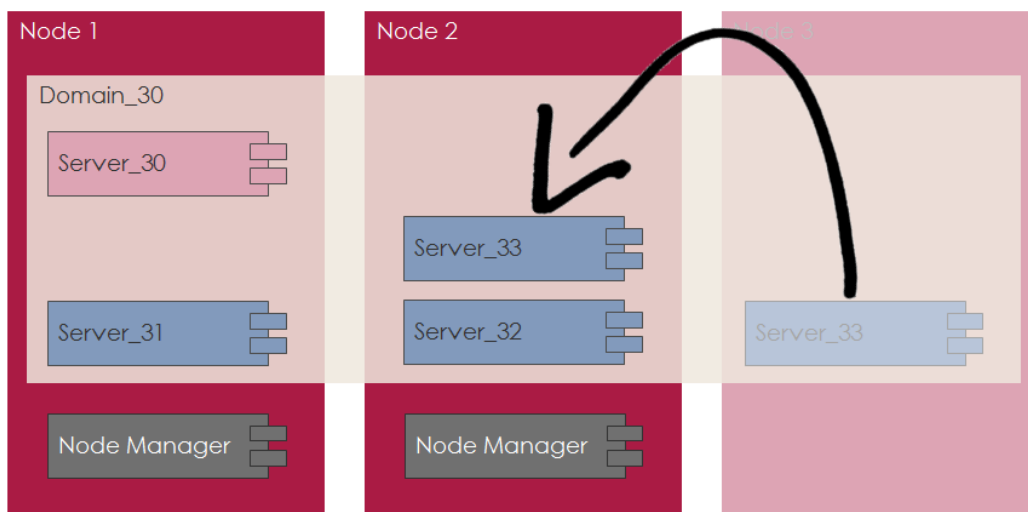


Abb. 2: Whole Server Migration

Kontaktadresse:

<p>Michael Fuhr Oracle Deutschland B.V. & Co.KG Geschäftsstelle Frankfurt Robert-Bosch 5 D-63303 Dreieich +49 6103-397773 michael.fuhr@oracle.com www.oracle.com/de</p>	<p>Stephan Rudolph rudolph ag Kramgasse 68 CH-3011 Bern +41 31 312 1920 stephan.rudolph@rudolph-ag.ch www.rudolph-ag.ch</p>
--	---