

# ***Java Performance Tuning***

## ***Seminarunterlage***

***Version: 5.06***

Dieses Dokument wird durch die ORDIX AG veröffentlicht.

Copyright ORDIX AG. Alle Rechte vorbehalten.

Alle Produkt- und Dienstleistungs-Bezeichnungen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Firmen und beziehen sich auf Eintragungen in den USA oder USA-Warenzeichen.

Weitere Logos und Produkt- oder Handelsnamen sind eingetragene Warenzeichen oder Warenzeichen der jeweiligen Unternehmen.

Kein Teil dieser Dokumentation darf ohne vorherige schriftliche Genehmigung der ORDIX AG weitergegeben oder benutzt werden.

### **Adressen der ORDIX AG**

Die ORDIX AG besitzt folgende Geschäftsstellen

ORDIX AG  
Karl-Schurz-Straße 19a  
D-33100 Paderborn  
Tel.: (+49) 0 52 51 / 10 63 - 0  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
An der alten Ziegelei 5  
D-48157 Münster  
Tel.: (+49) 02 51 / 9 24 35 – 00  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Welser Straße 9  
D-86368 Gersthofen  
Tel.: (+49) 08 21 / 507 492 – 0  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Kreuzberger Ring 13  
D-65205 Wiesbaden  
Tel.: (+49) 06 11 / 7 78 40 – 00  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Wikingerstraße 18-20  
D-51107 Köln  
Tel.: (+49) 02 21 / 8 70 61 – 0  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Südwestpark 67/2  
D-90449 Nürnberg  
Tel.: (+49) 0 52 51 / 10 63 - 0  
Fax.: (+49) 01 80 / 1 67 34 90

Internet: <http://www.ordix.de>

Email: [seminare@ordix.de](mailto:seminare@ordix.de)

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>6</b>
1.1	Agenda	7
1.2	Performance Grundlagen	8
1.2.1	Definition von Performance	8
1.2.2	Performance Technologie und Messgrößen	9
1.2.3	Messmethoden	10
1.3	Strategien und Techniken	12
1.3.1	Stufen	12
1.3.2	Tuning Strategie und –Techniken	13
1.4	Java Virtual Machine	19
1.4.1	Runtime Environment	19
1.4.2	Speicherbedarf	27
1.4.3	Class Loading	29
1.4.4	Garbage Collection	33
<b>2</b>	<b>Monitoring- und Profiling- Tools</b>	<b>34</b>
2.1	Agenda	35
2.2	Monitoring vs. Profiling	36
2.3	Schichtenmodell der Performance-Analyse	37
2.3.1	Schichtenmodell – Betriebssystem	38
2.3.1.1	Betriebssystem: Tools	39
2.3.1.2	Task Manager	40
2.3.1.3	Performance Monitor	43
2.3.1.4	UNIX Monitoring	44
2.3.1.5	Übersicht Betriebssystem-Tools	46
2.3.2	Schichtenmodell – JVM	47
2.3.2.1	JVM: Tools	48
2.3.2.2	Jps	49
2.3.2.3	jstatd / jstat	50
2.3.2.4	JConsole	51
2.3.2.5	VisualVM – Überblick	52
2.3.2.6	VisualVM – Features	53
2.3.2.7	VisualVM – Plugins	54
2.3.2.8	VisualVM – Monitor	55
2.3.2.9	VisualVM – Threads	56
2.3.2.10	VisualVM – Profiler	57
2.3.2.11	VisualVM – Visual GC	58
2.3.2.12	GC-Profiling	59
2.3.3	Schichtenmodell – Java Anwendung	60
2.3.3.1	Java Profiler	61
<b>3</b>	<b>JVM: Garbage Collection</b>	<b>63</b>
3.1	Agenda	64
3.2	Einführung	65
3.2.1	Altersverteilung von Objekten	66
3.3	Speicherbereiche/Generationen	67
3.3.1	Altersverteilung von Objekten	67
3.3.2	Heap: Aufteilung in Generationen	68
3.3.3	Speicherbereinigung nach Generationen	69
3.3.4	Perm Space	70
3.3.5	Aufteilung der Young Generation	71
3.3.6	Die Old Generation	72
3.4	GC Algorithmen	73
3.4.1	Young Generation	73
3.4.1.1	Mark and Copy	73
3.4.1.2	Ausgabe des Heap vor und nach GC	75
3.4.1.3	Mark and Copy	76
3.4.1.4	Intergenerational References	77

3.4.1.5	Parallelisierung der GC .....	79
3.4.1.6	Parallele Speicherallokation .....	80
3.4.2	Old Generation .....	81
3.4.2.1	Mark and Compact .....	81
3.4.2.2	Mark and Compact – Kompaktierung .....	82
3.4.2.3	Mark and Compact – Phasen .....	83
3.4.2.4	Mark and Compact – Markierung .....	84
3.4.2.5	Mark and Compact – Berechnung neuer Speicherorte .....	85
3.4.2.6	Mark and Compact – Anpassung der neuen Referenzen .....	86
3.4.2.7	Mark and Compact – Verschiebung der Objekte .....	87
3.4.2.8	Mark and Compact – Fazit .....	88
3.4.2.9	CMS – Concurrent Mark and Sweep .....	89
3.4.2.10	CMS – Initial Marking .....	90
3.4.2.11	CMS – Concurrent Marking .....	91
3.4.2.12	CMS – Remarking .....	92
3.4.2.13	CMS – Concurrent Sweep .....	93
3.4.2.14	CMS – Preclean .....	94
3.4.2.15	CMS – Parallelisierung .....	95
3.4.2.16	CMS – Überblick .....	96
3.4.2.17	CMS – Fazit .....	97
3.4.3	G1 – Garbage-First Garbage Collector .....	98
3.4.3.1	G1 – Ziel .....	100
3.4.3.2	Parallelität und Nebenläufigkeit .....	101
3.4.3.3	G1 – Aufbau des Heap .....	102
3.4.3.4	G1 – Garbage Collection .....	105
3.4.3.5	G1 – Glossar .....	106
3.4.3.6	G1 – Phasen der GC .....	109
3.4.3.7	G1 – 1. RS maintenance .....	110
3.4.3.8	G1 – 2. Concurrent Marking .....	112
3.4.3.9	G1 – 2. Concurrent Marking – Initial Marking .....	114
3.4.3.10	G1 – 2. Concurrent Marking – Remarking .....	115
3.4.3.11	G1 – 2. Concurrent Marking – Cleanup .....	116
3.4.3.12	G1 – 3. Evacuation and Collection .....	117
3.4.3.13	G1 – Tuning .....	119
3.4.3.14	G1 – GC-Ausgaben .....	121
3.5	ZGC .....	127
3.6	Shenandoah .....	129
3.7	Tuning der JVM .....	131
3.7.1	Strategien und Ziele .....	131
3.7.2	Durchsatz .....	132
3.7.3	Pausenzeit .....	133
3.7.4	Speicherverbrauch .....	134
3.7.5	Auswahl des „optimalen“ Garbage Collectors .....	135
3.8	Young Generation Collectors .....	136
3.9	Old Generation Collectors .....	137
3.10	Kombinationen von Garbage-Collectors .....	138
3.10.1	HotSpot VM Adaptive Tuning .....	141
3.10.2	Konfiguration des Speichers: Gross heap tuning .....	142
3.10.3	Konfiguration des Speichers: Fine heap tuning .....	143
3.10.4	Weitere nützliche JVM-Parameter .....	144
3.10.5	Garbage Collector JVM-Parameter .....	147
3.11	Monitoring der Garbage Collection .....	149
3.11.1	Überwachung der GC-Aktivitäten .....	149
<b>4</b>	<b>Tuning der Applikation .....</b>	<b>153</b>
4.1	Agenda .....	154
4.2	Mehr Performance durch „besseren“ Code .....	155
4.2.1	System.arraycopy() .....	156
4.2.2	Doppelte Initialisierung von Member-Variablen .....	157
4.2.3	Iterator bei For-Each-Schleife .....	158

4.2.4	StringBuffer und StringBuilder .....	159
4.2.5	Datentypen .....	160
4.2.6	Exception-Handling .....	161
4.2.7	Collections in Java .....	162
4.2.8	Maps in Java .....	163
4.3	Micro Benchmarking .....	164
4.4	Speicherverbrauch .....	166
4.4.1	Object Reuse / Pooling.....	166
4.4.2	Memory Leaks .....	168
4.4.3	WeakReferences.....	171
4.4.4	OutOfMemoryError-Analyse.....	173
4.5	Multithreading.....	174
4.5.1	Synchronization-Overhead.....	177
4.5.2	Dead-Lock-Beispiel .....	178
4.5.3	Dead-Lock Phänomene.....	179
4.5.4	Dining Philosopher's Problem .....	180
4.5.5	Dead-Locks finden .....	181
<b>5</b>	<b>Tuning von JEE-Applikationen .....</b>	<b>186</b>
5.1	Agenda.....	187
5.2	JEE-Architekturen .....	188
5.2.1	Skalierung – HTTP-Skalierung.....	189
5.2.2	Skalierung – EJB-Skalierung .....	190
5.3	Performance Faktoren .....	191
5.3.1	Einfluss auf die Performance .....	191
5.3.2	Application Server .....	192
5.3.2.1	Netzwerk-Kommunikation – Service Locator .....	192
5.3.2.2	Service-Locator .....	193
5.3.2.3	Service Locator – Implementierung.....	194
5.3.2.4	Netzwerk-Kommunikation – Session Facade .....	195
5.3.2.5	Session-Facade.....	196
5.3.2.6	Session Facade – Implementierung.....	197
5.3.2.7	Netzwerk-Kommunikation – Web-Anwendungen.....	198
5.3.3	Datenbank .....	199
5.3.3.1	Netzwerk-Kommunikation – JDBC-Optimierung .....	199
5.3.3.2	JDBC-Optimierung – Batch Updates.....	200
5.4	Konkurrierende Prozesse.....	204
5.4.1	Datenbank .....	204
5.4.1.1	Parallele Verarbeitung auf der Datenbank .....	204
5.4.2	Application Server .....	211
5.4.2.1	Parallele Verarbeitung - Application Server .....	211