

2.3	Problem Patterns	70
2.3.1	Falsche GC-Strategie und -Konfiguration	70
2.3.2	Falsch dimensionierter Heap	71
2.3.3	Memory Leaks	72
2.3.4	Zu hoher Speicherverbrauch	76
2.3.5	Classloader-Probleme	78
2.3.6	OutOfMemory-Fehler	80
2.3.7	Zu viel transaktionaler Speicherverbrauch (GC Thrashing)	81
2.3.8	Große temporäre Objekte	82
2.3.9	Fehlerhafte Implementierungen von Operatoren	83
<b>3</b>	<b>Die Datenbankschicht</b>	<b>85</b>
3.1	Architektur und Laufzeitverhalten	85
3.1.1	Die Rolle des Datenbank-Layers	85
3.1.2	Architektur einer Datenbankanwendung	87
3.1.3	Architekturüberblick	87
3.1.4	Die JDBC-Schicht	87
3.1.5	O/R Mapper	90
3.1.6	Caching	97
3.1.7	Speicherverhalten	99
3.2	Performanceanalyse	100
3.2.1	Statistischer Überblick	101
3.2.2	Transaktionale Analyse	102
3.2.3	Statement-Erzeugung	103
3.2.4	Speicherverhalten	104
3.2.5	JMX- und PMI-Metriken	104
3.3	Problem Patterns	106
3.3.1	Prepared-Statement-Verwendung	106
3.3.2	Transaktionsdauer und -verhalten	107
3.3.3	Ineffizientes Abfrageverhalten	109
3.3.4	Anwendungsdesign vs. Datenbankdesign	110
<b>4</b>	<b>Verteilung und Kommunikation</b>	<b>111</b>
4.1	Architektur und Laufzeitverhalten	111
4.1.1	Architektur eines Remoting-Protokolls	113
4.1.2	Zustandsmanagement	114
4.1.3	Laufzeitverhalten	115

4.1.4	Serialisierung	117
4.1.5	Latenz und Bandbreite	121
4.2	Remoting-Protokolle	122
4.2.1	Remote Method Invocation (RMI)	123
4.2.2	Java Messaging Service (JMS)	124
4.2.3	Web Services	126
4.2.4	REST	128
4.2.5	Enterprise Service Busses	129
4.2.6	CICS	129
4.3	Performanceanalyse	130
4.3.1	Analyse des Aufrufverhaltens	130
4.3.2	Serviceelastanalyse	131
4.3.3	Statistischer Überblick	132
4.3.4	Überwachung von Infrastrukturmetriken	134
4.3.5	Detailanalyse des Kommunikations-Stacks	135
4.3.6	Problemisolation	136
4.4	Problem Patterns	137
4.4.1	Falsches Protokoll	137
4.4.2	Gesprächige Anwendungen	137
4.4.3	Zu große Nachrichten	139
4.4.4	Schlechte Serviceinterfaces	140
4.4.5	Falsches Deployment	140
4.4.6	Konfigurationsfehler	142
4.4.7	Schlechtes Zustandsmanagement	142
<b>5</b>	<b>Web- und AJAX-Performance</b>	<b>143</b>
5.1	Architektur und Laufzeitverhalten	143
5.1.1	Das HTTP-Protokoll	143
5.1.2	DOM-Manipulation	153
5.1.3	Memory Leaks in JavaScript	157
5.2	Performanceanalyse	158
5.2.1	Netzwerkanalyse	159
5.2.2	Caching-Analyse	162
5.2.3	Analyse von Serveranfragen	163
5.2.4	JavaScript-Analyse	166
5.2.5	Analyse im Echtbetrieb	168
5.2.6	Technologien zum Sammeln von Performancedaten	169

5.3	Problem Patterns	172
5.3.1	Zu viele Requests	172
5.3.2	Langsame externe Inhalte	173
5.3.3	Senden zu vieler Daten	174
5.3.4	Security Overhead	175
5.3.5	Unterschätzen des Browsers	175
5.3.6	Ungewolltes Layouting	176
5.3.7	Schlechtes State Management	177
5.3.8	Falsche Seitenstruktur	177
<b>6</b>	<b>Performance und Testen</b>	<b>179</b>
6.1	Einleitung	179
6.2	Dynamische Architekturvalidierung	179
6.3	Performance in Continuous Integration	182
6.3.1	Arten von Tests	182
6.3.2	Aufsetzen einer Testumgebung	184
6.3.3	Messungen durchführen	184
6.3.4	Auswertung der Messdaten	186
6.3.5	Regressionsanalyse	188
6.4	Last- und Performancetests	190
6.4.1	Definition eines Testprozesses	191
6.4.2	Aufsetzen eines Lasttests	192
6.4.3	Durchführung eines Lasttests	198
6.4.4	Interpretation eines Testlaufs	202
6.5	Problem Patterns	212
<b>7</b>	<b>Performanceanalyse und -optimierung</b>	<b>215</b>
7.1	Einleitung	215
7.2	Überwachung	216
7.2.1	Die richtige Datensammlung	217
7.2.2	Überwachung einer Anwendung	221
7.2.3	Überwachung aus Endnutzersicht	229
7.2.4	Business Transaction Management	229
7.3	Erkennen von Problemen	230
7.3.1	Was ist ein Problem?	231
7.3.2	Service Level Monitoring	231
7.3.3	Key Performance Indikatoren (KPIs)	232

## Inhaltsverzeichnis

---

7.3.4	Trending	232
7.3.5	Incidents und Alerts	233
7.4	Isolation der Ursache	235
7.4.1	Langsame Antwortzeiten, schlechter Durchsatz	237
7.4.2	Steigende Ressourcenauslastung	240
7.4.3	Anwendungsfehler	241
7.5	Detaildiagnose	242
7.5.1	Typische Auslöser	242
7.5.2	Diagnose typischer Probleme	245
7.6	Problembehebung	249
7.6.1	Problemdokumentation	249
7.6.2	Testfall	250
7.6.3	Überprüfung der Problembehebung	250
7.6.4	Zurück zur Überwachung	250
7.7	Optimierung	251
7.7.1	Kontinuierliche Optimierung	252
7.7.2	Ein Prozess zur dauerhaften Performancesteigerung	253
7.8	Problem-Patterns	255
7.8.1	Datenflut	255
7.8.2	Blackbox-Monitoring	256
7.8.3	Teile und herrsche ... nicht	256
7.8.4	Vermutungen statt Fakten	256
7.8.5	Top Ten Reports	257
7.8.6	Unzureichende Verifikation	258