

Spis treści

Przedmowa	13
Przedmowa	15
Wstęp	17
Podziękowania	21
O autorach	23
1. Strategie, podejścia i metodologie	25
Zasadnicze czynniki	26
Dwa podejścia: z góry do dołu oraz z dołu do góry	29
Metoda z góry do dołu	29
Metoda z dołu do góry	30
Wybór odpowiedniej platformy i ocena systemu	31
Wybór właściwej architektury procesora	32
Ocena wydajności systemu	33
Bibliografia	34
2. Monitorowanie wydajności systemu operacyjnego	35
Definicje	36
Wykorzystanie CPU	36
Monitorowanie wykorzystania CPU w systemach Windows	37
Monitorowanie wykorzystania CPU w systemach Windows za pomocą komendy typeperf	40
Monitorowanie wykorzystania CPU w systemach Linux	41
Monitorowanie wykorzystania CPU w systemach Solaris	42
Monitorowanie wykorzystania CPU w systemach Linux i Solaris za pomocą narzędzi wiersza poleceń	45

Kolejka uruchamiania planisty krótkoterminowego	48
Monitorowanie kolejki uruchamiania planisty krótkoterminowego	
w systemach Windows	49
Monitorowanie kolejki uruchamiania planisty krótkoterminowego	
w systemach Solaris	50
Monitorowanie kolejki uruchamiania planisty krótkoterminowego	
w systemach Linux	51
Wykorzystanie pamięci	51
Monitorowanie wykorzystania pamięci w systemach Windows	52
Monitorowanie wykorzystania pamięci w systemach Solaris	53
Monitorowanie wykorzystania pamięci w systemach Linux	54
Monitorowanie rywalizacji o blokady w systemach Solaris	55
Monitorowanie rywalizacji o blokady w systemach Linux	57
Monitorowanie rywalizacji o blokady w systemach Windows	57
Izolowanie gorących blokad	58
Monitorowanie minowolnego przełączania kontekstu	58
Monitorowanie migracji wątków	59
Wykorzystanie we/wy sieci	59
Monitorowanie wykorzystania we/wy sieci w systemach Solaris	60
Monitorowanie wykorzystania we/wy sieci w systemach Linux	61
Monitorowanie wykorzystania we/wy sieci w systemach Windows	61
Względy dotyczące poprawy wydajności aplikacji	62
Wykorzystanie we/wy dysku	63
Dodatkowe narzędzia wiersza poleceń	65
Monitorowanie wykorzystania CPU w systemach z procesorami SPARC T	66
Bibliografia	69
3. Przegląd JVM	71
Wysokopoziomowa architektura HotSpot VM	72
HotSpot VM Runtime	73
Opcje wiersza poleceń	74
Cykl życia maszyny wirtualnej	75
Ładowanie klas maszyny wirtualnej	78
Weryfikacja kodu bajtowego	80
Udostępnianie danych klas	81
Interpreter	82
Obsługa wyjątków	84
Synchronizacja	84
Zarządzanie wątkami	85
Zarządzanie sterą C++	89
Java Native Interface	90
Obsługa błędów krytycznych VM	91
Mechanizmy odzyskiwania pamięci HotSpot VM	93
Pokoleniowy mechanizm odzyskiwania pamięci	93
Młode pokolenie	95
Szybka alokacja	97
Mechanizmy odzyskiwania pamięci, czyli osiółkowi w żłoby dano	97

Szeregowy mechanizm odzyskiwania pamięci	98
Równoległy mechanizm odzyskiwania pamięci: przepustowość ma znaczenie!	99
Przeważnie-równoczesny mechanizm odzyskiwania pamięci: opóźnienie ma znaczenie!	99
Mechanizm odzyskiwania pamięci najpierw-kosz: następca CMS	102
Porównanie	102
Generowanie pracy mechanizmu odzyskiwania pamięci	103
Perspektywa historyczna	103
Kompilatory JIT HotSpot VM	104
Analiza hierarchii klas	105
Polityka kompilacji	106
Deoptymalizacja	107
Kompilator JIT dla aplikacji klienckich	108
Kompilator JIT dla aplikacji serwerowych	108
SSA — wykres programowo zależny	108
Nadchodzące poprawki	110
Regulacja adaptacyjna HotSpot VM	111
Wartości domyślne dla Java 1.4.2	111
Ergonomiczne wartości domyślne dla 5. wersji Javy	111
Zaktualizowane domyślne wartości ergonomiczne dla Java 6 Update 18	113
Adaptacyjne ustalanie rozmiaru sterty Java	115
Nie tylko ergonomia	115
Bibliografia	116
4. Monitorowanie wydajności maszyny wirtualnej Javy	117
Definicje	118
Odzyskiwanie pamięci	118
Dane istotne dla procesu odzyskiwania pamięci	119
Raportowanie danych procesu odzyskiwania pamięci	119
Analiza offline danych z procesów odzyskiwania pamięci	129
Narzędzia graficzne	132
Kompilator JIT	150
Ładowanie klas	151
Monitorowanie aplikacji Java	153
Szybkie monitorowanie rywalizacji o blokady	154
Bibliografia	156
5. Profilowanie aplikacji Java	157
Terminologia	159
Terminy związane z profilowaniem	159
Pojęcia związane z programem Oracle Solaris Studio Performance Analyzer	159
Pojęcia związane z programem NetBeans Profiler	160
Oracle Solaris Studio Performance Analyzer	161
Obsługiwane platformy	161
Pobieranie i instalacja programu Oracle Solaris Studio Performance Analyzer	162
Przechwytywanie profilu Oracle Solaris Studio Performance Analyzer	163
Przeglądanie zebranej próby	166
Prezentacja danych	174

Filtrowanie danych profilu	177
Narzędzie wiersza poleceń <code>er_print</code>	178
Program NetBeans Profiler	185
Obsługiwane platformy	186
Pobieranie i instalacja NetBeans Profiler	186
Rozpoczynanie sesji profilowania metod	186
Controls, czyli przyciski sterujące	193
Status	193
Profiling Results, czyli rezultaty profilowania	193
Saved Snapshots, czyli zapisane zrzuty ekranu	193
View, czyli podgląd	194
Basic Telemetry, czyli podstawowa telemetria	194
Przeglądanie aktualnych rezultatów	195
Wykonywanie zrzutu ekranu rezultatów	195
Rozpoczynanie sesji profilowania pamięci	196
Przeglądanie bieżących rezultatów	198
Wykonywanie zrzutu ekranu rezultatów	200
Izolowanie wycieków pamięci	201
Analiza zrzutów sterty	201
Bibliografia	202
6. Profilowanie aplikacji Java — porady i sztuczki	203
Potencjalne obszary poprawy wydajności	203
Wykorzystanie CPU przez jądro lub system	204
Rywalizacja o blokady	212
Użycie słowa kluczowego <code>volatile</code>	221
Zmiana rozmiaru struktur danych	222
Zmiana rozmiaru klas <code>StringBuilder</code> lub <code>StringBuffer</code>	223
Zmiana rozmiaru klas <code>Java Collections</code>	226
Zwiększanie współczynnika równoległości	230
Wysokie wykorzystanie CPU	232
Inne użyteczne wskazówki dotyczące programu Performance Analyzer	233
Bibliografia	235
7. Regulacja JVM krok po kroku	237
Metodologia	238
Założenia	240
Wymagania dotyczące infrastruktury testowania	240
Wymagania systemowe aplikacji	241
Dostępność	241
Zarządzalność	241
Przepustowość	241
Opóźnienie i czas reakcji	242
Zużycie pamięci	242
Czas uruchamiania	242
Ranking wymagań systemowych	242
Wybór modelu wdrożenia JVM	243
Model wdrożenia pojedynczej instancji JVM	243

Model wdrożenia wielu instancji JVM	243
Wskazówki ogólne	244
Wybór środowiska uruchomieniowego JVM	244
Środowisko uruchomieniowe typu klienckiego czy serwerowego	244
32-bitowa czy 64-bitowa maszyna wirtualna Javy	245
Mechanizmy odzyskiwania pamięci	246
Podstawy regulowania mechanizmu odzyskiwania pamięci	246
Atrybuty wydajności	246
Podstawowe zasady	247
Opcje wiersza poleceń oraz rejestrowanie zdarzeń dla mechanizmu odzyskiwania pamięci	247
Określanie zużycia pamięci	251
Ograniczenia	251
Układ sterty HotSpot VM	252
Punkt początkowy rozmiaru sterty	254
Obliczanie rozmiaru żywych danych	256
Konfiguracja początkowego rozmiaru sterty	257
Dodatkowe uwagi	259
Regulowanie opóźnienia/czasu reakcji	259
Przesłanki	260
Precyzyjna regulacja rozmiaru przestrzeni młodego pokolenia	261
Precyzyjna regulacja rozmiaru przestrzeni starego pokolenia	264
Precyzyjna regulacja opóźnień dla równoczesnego mechanizmu odzyskiwania pamięci	267
Objaśnienie przestrzeni ocalałych	269
Objaśnienie progu zatrudnienia	271
Monitorowanie progu zatrudnienia	272
Ustalanie rozmiaru przestrzeni ocalałych	274
Bezpośrednio wywoływane procesy odzyskiwania pamięci	280
Równoczesne odzyskiwanie pamięci dla stałego pokolenia	281
Regulacja czasu przestojów CMS	282
Kolejne czynności	283
Regulowanie przepustowości aplikacji	283
Regulacja przepustowości dla CMS	284
Regulacja przepustowościowego mechanizmu odzyskiwania pamięci	285
Regulacja rozmiaru przestrzeni ocalałych	287
Regulacja liczby wątków przepustowościowego mechanizmu odzyskiwania pamięci	290
Wdrożenie na systemach NUMA	291
Kolejne czynności	291
Przypadki skrajne	291
Dodatkowe opcje wiersza poleceń do regulowania wydajności	292
Najnowsze i największe optymalizacje	292
Analiza ucieczki	292
Blokada przeciągana	293
Duże strony pamięci	294
Bibliografia	296

8. Testy porównawcze aplikacji Java	297
Wyzwania dotyczące benchmarków	298
Rozgrzewka	298
Odzyskiwanie pamięci	300
Zastosowanie metod Time interfejsów Java API	301
Usuwanie martwego kodu w wyniku optymalizacji	302
Wplatanie	307
Deoptymalizacja	311
Porady dotyczące przygotowywania mikrobenchmarków	315
Projektowanie eksperymentów	317
Zastosowanie metod statystycznych	318
Obliczanie średniej	318
Obliczanie odchylenia standardowego	319
Określanie przedziału ufności	319
Zastosowanie testów hipotezy	321
Porady dotyczące stosowania metod statystycznych	323
Literatura	324
Bibliografia	324
9. Testy porównawcze aplikacji wielowarstwowych	325
Wyzwania dotyczące benchmarków	326
Rozważania na temat benchmarków typu enterprise	328
Definiowanie systemu testowego (SUT)	328
Przygotowywanie mikrobenchmarków	329
Definiowanie modelu interakcji z użytkownikiem	329
Definiowanie metryk wydajności	333
Skalowanie benchmarku	337
Weryfikacja za pomocą prawa Little'a	338
Czas na zastanowienie	340
Analiza skalowalności	343
Przeprowadzanie benchmarku	343
Monitorowanie serwera aplikacji	347
Monitorowanie za pomocą serwera aplikacji GlassFish	347
Monitorowanie podsystemów	352
Wydajność systemów zewnętrznych	356
We/wy dysku	359
Monitorowanie i regulacja pul zasobów	361
Profilowanie aplikacji enterprise	362
Bibliografia	363
10. Wydajność aplikacji internetowych	365
Testy porównawcze aplikacji internetowych	366
Komponenty kontenera webowego	367
Konektor HTTP	368
Silnik serwetu	369
Monitorowanie i regulowanie wydajności kontenera webowego	369
Tryb deweloperski i tryb produkcyjny kontenera	370
Menedżer bezpieczeństwa	371

Regulacje JVM	371
HTTP service i kontener webowy	373
Nasłuchiwaniec HTTP	373
Najlepsze praktyki	386
Najlepsze praktyki dla serwletów i stron JSP	386
Buforowanie pamięci podręcznej zawartości	395
Trwałość sesji	400
Buforowanie plików serwera HTTP	401
Bibliografia	405
11. Wydajność usług internetowych	407
Wydajność XML	408
Cykl przetwarzania XML	408
Parsowanie/unmarshalling	409
Dostęp	412
Modyfikacja	412
Serializacja/marshalling	413
Weryfikacja	413
Rozwiązywanie encji zewnętrznych	415
Częściowe przetwarzanie dokumentów XML	417
Wybór odpowiedniego interfejsu API	420
Stos modelowej implementacji JAX-WS	423
Testy porównawcze usług internetowych	425
Czynniki, które wpływają na wydajność usługi internetowej	428
Wpływ rozmiaru wiadomości	428
Charakterystyki wydajności różnych typów schematów	430
Implementacja punktu końcowego	433
Wydajność programu obsługi	434
Najlepsze praktyki dotyczące wydajności	436
Przetwarzanie binarnego bloku danych	436
Praca z dokumentami XML	441
Wykorzystanie MTOM do wysyłania dokumentów XML w postaci załączników	441
Korzystanie z interfejsu provider	444
Fast Infoset	446
Kompresja HTTP	448
Wydajność klienta usługi internetowej	449
Bibliografia	450
12. Wydajność Java Persistence oraz Enterprise Java Beans	451
Model programowania EJB	452
Interfejs Java Persistence API i jego implementacja modelowa	453
Pamięć podręczna drugiego poziomu	453
Monitorowanie i regulacja kontenera EJB	456
Pula wątków	457
Pule beanów i pamięci podręczne beanów	459
Pamięć podręczna sesji EclipseLink	464
Poziom izolacji transakcji	465

17	Najlepsze praktyki stosowane w Enterprise Java Beans	466
171	Benchmark do testów porównawczych EJB wykorzystany w przykładach	466
172	EJB 2.1	466
182	EJB 3.0	478
183	Najlepsze praktyki stosowane w Java Persistence	481
184	Zapytania języka zapytań JPA	482
186	Pamięć podręczna wyników zapytań	484
188	FetchType	485
189	Tworzenie puli połączeń	486
190	Aktualizacje zbiorcze	488
192	Wybierz właściwą strategię blokowania bazy danych	489
193	Odczyty bez transakcji	490
196	Dziedziczenie	490
197	Bibliografia	490

A. Wybrane opcje wiersza poleceń HotSpot VM 493

Skorowidz 511