

» Idź do

- Spis treści
- Przykładowy rozdział

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

- Fragmenty książek online

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl
© Helion 1991-2008

Oracle Database 11g. Nowe możliwości

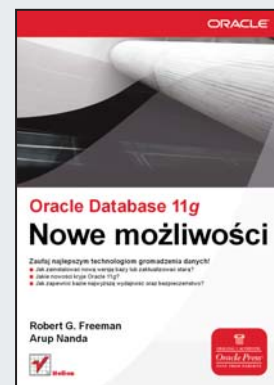
Autorzy: [Robert G. Freeman](#), Arup Nanda

Tłumaczenie: Kamila Primke

ISBN: 978-83-246-1857-6

Tytuł oryginału: [Oracle Database 11g
New Features \(Osborne Oracle Press\)](#)

Format: 168×237, stron: 344



Zaufaj najlepszym technologiom gromadzenia danych!

- Jak zainstalować nową wersję bazy lub zaktualizować starą?
- Jakie nowości kryje Oracle 11g?
- Jak zapewnić bazie najwyższą wydajność oraz bezpieczeństwo?

Być może będzie Ci trudno w to uwierzyć, ale baza danych firmy Oracle ma już blisko trzydziestkę! Pierwsze wydanie tej prawdopodobnie najlepszej bazy danych na świecie datuje się na 1979 rok. Przez te trzydzieści lat rozwiązanie to zdobyło sobie niezliczonych zwolenników, wyznaczyło standardy i zostało wdrożone wszędzie tam, gdzie wymagana jest największa niezawodność i wydajność. Synonimem tych dwóch słów może być właśnie Oracle!

W tej książce poza opisem nowych funkcji znajdziesz szereg informacji na temat sposobu instalacji Oracle Database 11g lub uaktualnienia istniejącej wersji do najnowszej. Dzięki „Oracle Database 11g. Nowe możliwości” dowiesz się, jak wykorzystać funkcje automatycznego zarządzania pamięcią. Ponadto poznasz dostępne narzędzia doradcze (Oracle Database Advisors), a także nauczysz się zapewniać bazie najwyższą wydajność i bezpieczeństwo. Jeżeli masz jakiegokolwiek wątpliwości co do tego, czy warto zacząć używać nowej wersji bazy Oracle, ta książka z pewnością je rozwieje!

- Instalacja Oracle 11g
- Sposoby aktualizacji do wersji 11g
- Charakterystyka nowych własności zarządzania
- Wykorzystanie Automatic Memory Management
- Zastosowanie mechanizmu AutoTask
- Zapewnienie dostępności oraz sposoby przywracania danych
- Wykorzystanie narzędzi doradczych (Oracle Database Advisors)
- Sposoby użycia narzędzia SQL Performance Analyzer
- Zapewnienie bezpieczeństwa danym zgromadzonym w bazie
- Zestawienie najlepszych własności Oracle 11g

Wykorzystaj nowe możliwości bazy Oracle!

Spis treści

O autorze	11
O współautorze	13
O redaktorze technicznym	15
Przedmowa	17
Podziękowania	19
Wstęp	21
Rozdział 1. Oracle 11g, zaczynamy	23
Instalacja Oracle Database 11g	24
Asystent Konfiguracji Bazy Danych	25
Uaktualnienie do Oracle Database 11g	25
Oszczędzanie czasu przy uaktualnianiu	26
Wspierane ścieżki aktualizacji	29
Wspierane metody aktualizacji	29
Aktualizacja z DBUA	30
Ręczna aktualizacja	36
Eksportowanie i importowanie w celu aktualizacji bądź jej wycofania	41
Zmiany w parametrach Oracle	43
Nowe parametry	43
Parametry niezalecane	45
Parametry przestarzałe	45
Dziwactwa parametru undo_management	46
Zmiany w widokach słownikowych	46
Słowo na koniec	46
Rozdział 2. Nowe własności zarządzania	47
Zmiany i nowe własności w ASM	47
Nowa dokumentacja ASM	48
Atrybuty grup dyskowych ASM	48
ASM Fast Disk Resync	49
Opcje kompatybilności ASM	50
Preferowane odczyty lustrzane ASM	51
ASM rolling upgrade	52
Wsparcie ASM dla jednostek alokacji o zmiennym rozmiarze	53
Nowa rola SYSASM	53
Nowe polecenia narzędzia asmcmd	53

Automatic Memory Management	56
Przegląd Automatic Memory Management	56
Nowe funkcje w Memory Advisor i nowe widoki	59
W jaki sposób Oracle zarządza pamięcią?	60
Automatic Memory Management i OEM	60
Przejście na Automatic Memory Management	60
Nowe własności ADDM	61
Nowe widoki ADDM	61
Współistnienie RAC i ADDM	62
Zarządzanie ADDM poprzez dbms_addm	63
Klasyfikacja znalezisk	64
Dyrektywy	65
Nowe własności AWR	67
Domyślna retencja migawek AWR	67
Nowe własności linii bazowych AWR	67
Adaptacyjne progi metryk	71
AutoTask — planowanie zautomatyzowanych zadań utrzymania	72
Architektura AutoTask	72
Widoki danych słownikowych dla AutoTask	73
Zarządzanie zadaniami AutoTask poprzez OEM	75
Ręczne zarządzanie zadaniami AutoTask	78
Okna utrzymania AutoTask	78
Zmiany i nowe własności w zarządzaniu plikiem parametrów	79
Obsługa błędów odczytu/zapisu do SPFILE	79
Prostsze przejście na wykorzystywanie plików SPFILE	80
Zabezpieczenie przed ustawianiem przez użytkowników nieprawidłowych wartości w plikach SPFILE	80
Zmiany i nowe własności w Resource Managerze	80
Kalibracja wejścia/wyjścia	81
Domyślny plan utrzymania	82
Wbudowane plany zasobów	83
Statystyki Resource Managera w AWR	83
Nowe własności dyrektyw planu Resource Managera	83
Zależności drobnoziarniste	85
Obecnie domyślna opcja DDL WAIT	85
Nowa funkcjonalność dodawania kolumn	87
Słowo na koniec	87
Rozdział 3. Nowe własności dostępności i odtwarzania	89
Infrastruktura diagnozowania błędów	89
Automatic Diagnostic Repository (ADR)	90
Dziennik alertów	91
Pliki śladu, pliki rzutu, pliki core	92
Support Workbench	92
Nowe własności RMAN	105
Zrównoleglenie interplikowej archiwizacji	105
Szybsza kompresja archiwizacji	107
Aktywna duplikacja bazy danych	107
Udoskonalona obsługa długoterminowych archiwizacji	110
Przełączenie awaryjne kopii zapasowej zarchiwizowanych plików dziennika powtórzeń	111
Ulepszona polityka usuwania zarchiwizowanych plików dziennika powtórzeń	111
Ulepszenia w katalogu odtwarzania	112
Optymalizacja archiwizacji wycofania	113

Zwiększenie wydajności odtwarzania pojedynczych bloków danych (Block Media Recovery)	113
Pozostałe nowe własności RMAN	114
Nowe własności powiązane z mechanizmem Flashback	115
Oracle Flashback Transaction Backout	115
Oracle Flashback Data Archives	118
Nowe własności rezerwowych baz danych (Standby Database)	123
Detekcja utraconego zapisu	123
Kompresja zarchiwizowanych plików dziennika powtórzeń	124
Możliwości zapytań czasu rzeczywistego w fizycznej rezerwowej bazie danych	124
Migawkowe bazy danych	124
Nowe własności Oracle Data Pump	125
Ustalenie narzędzia exp niezalecanym	126
Kompresja zbiorów plików zrzutu	126
Udoskonalenie szyfrowania w Data Pump	127
Remapowanie danych w Data Pump (zaciemnianie)	127
Zmiana nazwy tabeli z Data Pump	128
Data Pump i tabele partycjonowane	128
Nadpisywanie plików zrzutu	129
Parametr data_options w Data Pump	129
Parametr transportable	130
Słowo na koniec	130
Rozdział 4. Narzędzia doradcze (Oracle Database Advisors)	131
Data Recovery Advisor	131
SQL Repair Advisor	141
SQL Access Advisor	142
Streams Performance Advisor	146
Widoki narzędzi doradczych w Oracle Database 11g	147
Słowo na koniec	148
Rozdział 5. Zarządzanie zmianami	149
Database Replay	149
Używanie Oracle Database Replay	150
Przegląd Database Replay	150
Możliwości i ograniczenia przechwytywania obciążenia przez Database Replay	151
Database Replay — przechwytywanie obciążenia	152
Database Replay — przetwarzanie wstępne przechwycenia obciążenia	163
Database Replay — odtwarzanie obciążenia	166
SQL Performance Analyzer	177
Wprowadzenie do SQL Performance Analyzer	178
SQL Performance Analyzer w OEM	178
SQL Performance Analyzer z wykorzystaniem procedur PL/SQL	184
Słowo na koniec	189
Rozdział 6. Bezpieczeństwo	191
Monitorowanie	191
Własności hasel	192
Ustawienia hasel i profil domyślny	192
Złożoność hasel	193
Wrażliwość na wielkość liter	193
Zapobieganie włamaniom poprzez opóźnienie logowania po nieudanych próbach	194
Zmiany w haszowaniu hasel	196
Używanie hasel domyślnych	196

Drobnoziarnista kontrola dostępu (FGAC) w usługach sieciowych	197
Tworzenie ACL oraz definiowanie powiązanych z listą uprawnień	197
Przypisywanie ACL do hostów w sieci	198
Widoki słownikowe dla ACL	198
Szyfrowanie przestrzeni tabel	199
Wprowadzenie do szyfrowania przestrzeni tabel	199
Przygotowanie bazy danych do szyfrowania przestrzeni tabel	199
Tworzenie szyfrowanej przestrzeni tabel	201
Szyfrowanie a wydajność systemu bazy danych	202
TDE a Log Miner, Oracle Streams i logiczne rezerwowe bazy danych	203
Oracle SECUREFILE LOB	203
Słowo na koniec	204
Rozdział 7. Inteligencja biznesowa i hurtownie danych	205
Partycjonowanie	205
Partycjonowanie przedziałowe	205
Rozszerzone partycjonowanie złożone	209
Partycjonowanie referencyjne	210
Partycjonowanie systemowe	212
Indeksy domenowe zarządzane systemowo	213
Kolumny wirtualne	214
O kolumnach wirtualnych	214
Tworzenie tabel z kolumnami wirtualnymi	215
Partycjonowanie tabel z kolumnami wirtualnymi	217
Data Pump — importowanie pojedynczych partycji	217
Widoki zmaterializowane i przepisywanie zapytań	218
Redefiniowanie online tabel z dziennikami widoku zmaterializowanego	219
Przepisywanie zapytań podczas odświeżania	219
Odświeżanie Partition Change Tracking dla widoków zmaterializowanych z klauzulą union all	219
Nowe i ulepszone widoki katalogowe widoków zmaterializowanych	219
Udoskonalenie przepisywania zapytań	221
Klauzula pivot i klauzula unpivot	223
Klauzula pivot	223
Klauzula unpivot	225
Kompresja tabel	227
Słowo na koniec	228
Rozdział 8. Rozwój aplikacji	229
SQL*Plus	229
Nowe polecenia set	229
Zdarzenia Fast Application Notification w bazie danych RAC	230
Utrzymanie i aktualizacja online aplikacji	231
Nowy parametr lock table	231
Mniej blokad wyłącznych zakładanych podczas operacji online	231
Indeksy niewidzialne	231
SQL	233
Tabele tylko do odczytu	233
SQL Query Result Cache	234
Pamięć podręczna wyników po stronie klienta	238
Ulepszenie wyrażeń regularnych	238
Notacja nazwana i mieszana w SQL	239
PL/SQL	239
Tworzenie wyzwalaczy jako włączonych bądź wyłączonych	240
Klauzula follows polecenia create trigger	240

Wyzwalacze złożone	241
Wbudowywanie kodu (inlining)	244
Typ danych simple_integer	245
Pamięć podręczna wyników funkcji PL/SQL	246
Dynamiczny SQL	246
Zmienne REF CURSOR i Dynamiczny SQL	247
Ostrzeżenie PLW-06009	248
Ulepszenia sekwencji w PL/SQL	248
Instrukcja continue w PL/SQL	248
Słowo na koniec	250
Rozdział 9. Dostrajanie wydajności	251
Ulepszone monitorowanie procesów Oracle	251
Statystyki	253
Statystyki oczekujące i opublikowane	253
Przywracanie poprzednich statystyk	255
Statystyki rozszerzone	256
Natywna kompilacja PL/SQL	262
SQL Plan Management	263
Przegląd SQL Plan Management	263
Pochwycenie planu	263
Wykorzystanie linii bazowych planów SQL	265
Odpytywanie linii bazowych planów SQL	266
Ewolucja linii bazowych planów SQL	267
Zarządzanie liniami bazowymi planów SQL	268
Automatic SQL Tuning	270
Przegląd Automatic SQL Tuning	271
Automatic SQL Tuning w OEM	273
Ręczne zarządzanie Automatic SQL Tuning	276
Ręczne tworzenie i używanie zbiorów SQL Tuning Set	279
Stworzenie zadania	279
Stworzenie SQL Tuning Set	280
Załadowanie SQL Tuning Set	280
Połączenie SQL Tuning Set z zadaniem	281
Ustawienie parametrów zadania	281
Wykonanie zadania	281
Przejrzenie wyników	282
Inteligentne współdzielenie kursorów (bind-aware peeking)	282
O bind-aware peeking	283
Widoki związane z bind-aware peeking	283
Uruchamianie systemu z bind-aware peeking	284
Własności tymczasowych przestrzeni tabel	284
Kurczenie tymczasowych przestrzeni tabel	285
Widok DBA_TEMP_FREE_SPACE	285
Monitorowanie SQL czasu rzeczywistego	286
Przegląd Real-Time SQL Monitoring	286
Widoki Real-Time SQL Monitoring	286
Raport Real-Time SQL Monitoring	287
Kontrolowanie Real-Time SQL Monitoring	287
Kontrolowanie użycia OEM Management Packs	287
Słowo na koniec	288

Rozdział 10. Inne nowe własności i ulepszenia	289
Real Application Clusters	289
Równoważenie obciążenia połączeń OCI w czasie trwania	290
Używanie transakcji XA w RAC	290
RAC Configuration Assistants	291
Network Configuration Assistant (NetCA)	291
Uaktualnienie bazy danych typu rolling upgrade	292
Zapytanie równoległe uznaje umiejscowienie usługi	293
Direct NFS	293
Nowe własności XMLDB	294
Składowanie binarne XML	294
Wsparcie partycjonowania dla XMP	299
Ulepszenia XQuery	299
Natywne Web Services w bazie danych	300
Udoskonalenia XML DB Repository	301
Zestaw narzędzi dewelopera XML	302
Java	303
Własności Oracle związane z JVM	303
Udoskonalenia istniejących narzędzi	305
Narzędzie ojvmtc	306
Wsparcie JDBC 4.0	307
Wsparcie JDK w Oracle Database 11g	308
Nowe procedury i pakiety dostarczone w Oracle	308
Dodatek 10 najlepszych własności według Arupa	311
Najlepsza dziesiątka Arupa: #1 — Database Replay	312
Najlepsza dziesiątka Arupa: #2 — SQL Performance Analyzer	313
Najlepsza dziesiątka Arupa: #3 — partycjonowanie	314
Najlepsza dziesiątka Arupa: #4 — przezroczyste szyfrowanie przestrzeni tabel	315
Najlepsza dziesiątka Arupa: #5 — Flashback Data Archive	316
Najlepsza dziesiątka Arupa: #6 — SQL Plan Management	318
Najlepsza dziesiątka Arupa: #7 — statystyki prywatne	319
Najlepsza dziesiątka Arupa: #8 — większa współbieżność	320
Najlepsza dziesiątka Arupa: #9 — pamięć podręczna wyników	321
Najlepsza dziesiątka Arupa: #10 — wyższa jakość kodu PL/SQL	322
Skorowidz	325

Rozdział 2.

Nowe własności zarządzania

Mówi się, że rosyjskie zimy są mroźne i długie. Muszę przyznać, że ten rozdział będzie jak rosyjska zima. Jest długi i może bardziej suchy niż mroźny, ale za to cały na temat bardzo ważnych spraw dotyczących Oracle Database 11g. Wersja ta dostarcza wielu nowych własności i ulepszeń ułatwiających zarządzanie bazą danych. Są wśród nich:

- ◆ Automatic Storage Management (ASM),
- ◆ Automatic Memory Management,
- ◆ Automatic Database Diagnostic Monitor (ADDM),
- ◆ Automatic Workload Repository (AWR),
- ◆ AutoTask — planista zautomatyzowanych zadań utrzymania,
- ◆ zmiany i nowe własności w pliku parametrów,
- ◆ Resource Manager,
- ◆ bardziej drobnoziarniste zależności,
- ◆ DDL WAIT,
- ◆ dodawanie kolumn z wartościami domyślnymi.

Rozpocznijmy od zmian i nowych własności związanych z ASM.

Zmiany i nowe własności w ASM

ASM (Automatic Storage Management) zawiera teraz wiele nowych własności, zaprojektowanych tak, aby używanie ASM stało się łatwiejsze i wydajniejsze. W skład nowości w ASM wchodzi:

- ◆ nowa dokumentacja,
- ◆ atrybuty grup dyskowych ASM,
- ◆ ASM Fast Disk Resync,
- ◆ atrybuty kompatybilności ASM,
- ◆ preferowane odczyty lustrzane ASM,
- ◆ aktualizacja ASM typu rolling upgrade,
- ◆ inne udoskonalenia ASM,
- ◆ nowa rola SYSASM.

Nowa dokumentacja ASM

Do Oracle Database 11g została dodana dodatkowa dokumentacja poświęcona zarządzaniu przestrzenią. Przewodnik *Oracle Database Storage Administration Guide* jest teraz częścią całkowitej dokumentacji bazy danych, a omawia w szczególności instalację, konfigurację oraz użycie ASM.

Atrybuty grup dyskowych ASM

Dostępna nowa klauzula `attribute` ASM pozwala przypisać atrybuty bezpośrednio do grup dyskowych ASM (w przeciwieństwie do sytuacji użycia szablonów). Niektóre z atrybutów istniały już w Oracle Database 10g, ale zostały dodane także nowe, które mogą zostać przypisane do grupy dyskowej (omówimy to w dalszej części rozdziału). Nowa jest także sama klauzula `attribute`. Oba polecenia: `create diskgroup` i `alter diskgroup`, pozwalają zdefiniować lub modyfikować ustawienia atrybutów adekwatnie do wymagań. Poniższa tabela zawiera listę atrybutów, które można ustawić za pomocą nowej klauzuli `attribute`.

Atrybut	Opis
<code>au_size</code>	Jest to rozmiar jednostki alokacji (<i>allocation unit</i>), który domyślnie wynosi 1 MB. Wartość tego atrybutu można ustawić tylko podczas tworzenia grupy dyskowej. Może nią być dowolna potęga liczby 2 (1, 2, 4, 8 itd.), od 1 M do 64 M. Przykładowo 4 M to 4194304.
<code>compatible.rdbms</code>	Więcej informacji o tym atrybucie znajduje się w dalszym punkcie tego rozdziału, zatytułowanym „Opcje kompatybilności ASM”. Jego przykładowa wartość to '11.0', a domyślna '10.1'. Raz ustawionej wartości atrybutu nie można z powrotem zamienić na niższą.
<code>compatible.asm</code>	Definiuje format danych na dyskach ASM. Więcej o tym atrybucie w dalszym punkcie tego rozdziału, „Opcje kompatybilności ASM”. Jego przykładowa wartość to '11.0', a domyślna to '10.1'.
<code>disk_repair_time</code>	Powoduje, że proces <code>resync</code> rozpoczyna śledzenie zmian w ekstentach należących do dysków offline. Więcej informacji o tym atrybucie znajduje się w dalszym punkcie tego rozdziału, „ASM Fast Disk Resync”. Jego dozwolone wartości mieszczą się w przedziale od 0 do 136 lat. Wsparcie dla tego atrybutu jest zapewnione, tylko jeśli oba atrybuty kompatybilności (<code>compatible.rdbms</code> i <code>compatible.asm</code>) mają ustawioną wartość 11.1 lub wyższą.

W celu sprawdzenia indywidualnych atrybutów przypisanych do każdej z grup dyskowych można zadać zapytanie do widoku V\$ASM_ATTRIBUTE. Oto przykład:

```
Select group_number, name, value
from v$asm_attribute order by group_number, name;
GROUP_NUMBER NAME VALUE
-----
1 au_size 1048576
1 compatible.asm 11.1.0.0.0
1 compatible.rdbms 11.1.0
1 disk_repair_time 600M
```

ASM Fast Disk Resync

W Oracle Database 11g wprowadzono ASM Fast Disk Resync, nową własność, która umożliwia zsynchronizowanie dysków ASM w ramach grupy dyskowej ASM z grupą dysków, która przetrwała awarię, po tym jak grupa, która jej uległa, jest niedostępna przez jakiś okres czasu (na przykład w związku z awarią kontrolera dysku). Jeśli awaria nie spowodowała uszkodzenia podłączonych mediów dyskowych, ASM Fast Disk Resync będzie mógł ponownie zsynchronizować brakujące dyski, gdy tylko przeszkody powodujące niedostępność tych dysków zostaną usunięte. Czas wykonania ponownej synchronizacji jest zależny od wielu czynników, ale często jest krótszy niż odbudowa całej grupy dyskowej. Podczas takiej synchronizacji nasze dyski ASM mogą być w pełni operacyjne. Należy wziąć pod uwagę to, że w trakcie naprawy dysku może się obniżyć wydajność. Trzeba także uwzględnić fakt, że czas całkowitego przestoju i ponownej synchronizacji niesie ze sobą utratę redundancji.

Zazwyczaj ASM usuwa dysk w niedługim czasie od jego przejścia w stan offline. Aby włączyć możliwość szybkiej ponownej synchronizacji dysków (i zapobiec przejściu dysku w stan offline), należy dla danej grupy dyskowej ustawić atrybut `disk_repair_time`. Jeśli atrybut `disk_repair_time` jest ustawiony, proces resync rozpocznie śledzenie zmian w ekstentach dyskowych należących do dysku offline. Kiedy dysk zostanie przywrócony w tryb online, proces resync zsynchronizuje go i włączy z powrotem do grupy dyskowej.

Czas naprawy dysku definiujemy w jednostkach minutowych lub godzinowych (posługując się oznaczeniami m/M lub h/H w celu wskazania wybranej jednostki). Dozwolone jest używanie dziesiętnych wyrażeń ułamkowych (np. 3.5 H). Jeśli ustawimy atrybut `disk_repair_time` bez wskazania wartości, domyślną będzie 3.6 godziny. Dlatego, jeśli zdamy się na wartość domyślną, należy się bardzo spieszyć z wymianą uszkodzonych kabelków! Chociaż w przypadku gdy gromadzimy karty dyskowe, kable i tym podobne w naszym komputerowym pomieszczeniu i śpimy w pobliżu na łóżku polowym, to 3,6 godziny może okazać się w pełni wystarczające. Mówiąc poważnie, lepiej rozważyć zwiększenie wartości tego atrybutu, gdyż z reguły mieszkamy daleko, a awarie występują o takiej porze, że ciężko jest zdobyć zastępczy sprzęt w przeciagu zaledwie 3,6 godziny. Czas naprawy zostanie zresetowany po przywróceniu dysku w tryb online, dlatego w przypadku kolejnego przejścia w stan offline zegar zacznie liczyć od 0.

Jeśli dysk przejdzie w stan offline, a czas naprawy dysku upłynie, to taki dysk zostanie logicznie usunięty. Przed upływem zdefiniowanego czasu naprawy sami możemy usunąć dysk offline, wywołując polecenie `alter diskgroup ... disk offline` z użyciem klauzuli `drop after`.



Uwaga

Atrybutu `disk_repair_time` nie można ustawić dla dysków, które już są dyskami offline.

Oto kilka przykładów ustawiania atrybutu czasu naprawy dysku dla grupy dyskowej:

```
ALTER DISKGROUP DG1 SET ATTRIBUTE 'DISK_REPAIR_TIME'='18H';
ALTER DISKGROUP DG2 SET ATTRIBUTE 'DISK_REPAIR_TIME'='12.5H';
ALTER DISKGROUP DG3 SET ATTRIBUTE 'DISK_REPAIR_TIME'='600M';
```

Opcje kompatybilności ASM

Oracle Database 11g daje możliwość ziarnistej kontroli kompatybilności z uwzględnieniem grup dyskowych ASM. Służą temu dwa różne atrybuty ustawiane za pomocą polecenia `alter diskgroup`. Pierwszy z nich to atrybut **kompatybilności grup dyskowych Oracle**. Wartość atrybutu kompatybilności grup dyskowych definiuje format danych na dysku ASM i zawsze powinna być większa lub równa wartości parametru kompatybilności bazy danych Oracle mającej dostęp do tego dysku ASM. Atrybut `compatible.asm` musi być ustawiony na wartość wyższą bądź równą wartości atrybutu `compatible.rdbms`. Raz ustawionej wartości atrybutu nie można cofnąć do poprzedniej, niższej.

Wartość atrybutu kompatybilności grup dyskowych Oracle ustawia się za pomocą polecenia `alter diskgroup` i atrybutu `compatible.asm`, jak pokazano w poniższym przykładzie:

```
ALTER DISKGROUP DG1 SET ATTRIBUTE 'compatible.asm'='11.1.0';
```

Ustawienia wartości tego parametru można także dokonać podczas wywołania polecenia `create diskgroup`, jak pokazano niżej:

```
CREATE DISKGROUP dgroup4 EXTERNAL REDUNDANCY
DISK '/oracle/asmdata/asm_dgroup1_04.asm'
ATTRIBUTE 'compatible.asm' = '11.1';
```

Drugim ze wspomnianych atrybutów jest atrybut **kompatybilności bazy danych Oracle**, którego wartość definiowana jest przez atrybut grupy dyskowej ASM `compatible.rdbms`. Atrybut ten określa minimalną wersję bazy danych Oracle, która może zamontować daną grupę dyskową. Jego wartość powinna być równa najniższej wartości ustawień kompatybilności wśród wszystkich baz danych Oracle, które zamontują daną grupę dyskową. Raz ustawionej wartości atrybutu nie można cofnąć do poprzedniej, gdyż wartość atrybutu implikuje format wiadomości przesyłanych w obie strony pomiędzy instancją ASM a powiązаныmi instancjami baz danych. Oczywiście możliwe jest zwiększenie wartości atrybutu.

Zauważmy, że każda grupa dyskowa może posiadać swoje własne ustawienia kompatybilności, a w związku z tym różne wersje baz danych Oracle mogą łączyć do każdej danej instancji ASM. Poniżej prezentujemy przykład ustawienia atrybutu kompatybilności bazy danych:

```
ALTER DISKGROUP DG1 SET ATTRIBUTE 'compatible.rdbms'='11.1.0';
```

Przykład ilustrujący wzajemny wpływ atrybutów na siebie można zauważyć, przyglądając się związkowi pomiędzy atrybutem `disk_repair_time` a atrybutem `compatible.asm`. Jeśli zaakceptujemy domyślną wartość atrybutu `compatible.asm` i spróbujemy ustawić wartość `disk_repair_time`, otrzymamy następujący błąd:

```
ORA-15032: not all alterations performed
```

```
ORA-15242: could not set attribute DISK_REPAIR_TIME
ORA-15283: ASM operation requires compatible.rdbms of 11.1.0.0.0 or higher
```

Komunikat błędu wskazuje, że należy ustawić atrybut `compatible.asm` dla skonfigurowanej grupy dyskowej. Aby błędu uniknąć, zmienimy wartość atrybutu `compatible.asm` z domyślnej '10.1' na '11.1.0', tak jak pokazano w przykładzie poniżej:

```
ALTER DISKGROUP DG1 SET ATTRIBUTE 'compatible.asm'='11.1.0';
```

Parametr kompatybilności nie tylko kontroluje, które bazy danych mogą łączyć się do naszej grupy dyskowej ASM, ale także definiuje własności dostępne w ASM (analogicznie jak w przypadku parametru kompatybilności bazy danych). Ustawienia kompatybilności grup dyskowych można sprawdzić w widoku `V$ASM_ATTRIBUTE`.

Przykładem wzajemnego oddziaływania atrybutów `compatible.asm` i `compatible.rdbms` może być sytuacja, kiedy `compatible.rdbms` ma wartość 10.2, a `compatible.asm` — 11.1. W takim przypadku zarządzanie grupą dysków ASM zostanie ograniczone do wersji Oracle 11.1 lub wyższej, ale indywidualne grupy dyskowe mogą współpracować z instancjami bazy danych Oracle z parametrem kompatybilności ustawionym na wartość 10.2 lub wyższą.

Preferowane odczyty lustrzane ASM

Niektóre konfiguracje ASM angażują zdalne kopie lustrzane dysków znajdujących się w bardzo odległych lokalizacjach (mogą także angażować zdalne instancje). W takich przypadkach podstawowa grupa dyskowa może być nie najlepszym zbiorem dysków do odczytu przez daną instancję. Przykładowo, możemy posiadać dwie instancje RAC (*Real Application Cluster*) na urządzeniach oddalonych od siebie o 500 kilometrów. Dany zespół dysków może być przechowywany w określonym systemie RAC. W takiej sytuacji chcielibyśmy, aby lokalna instancja RAC mogła czytać z najbliższego dostępnego dysku ASM w celu zapewnienia najlepszej wydajności.

Preferowane odczyty lustrzane ASM dostępne są jedynie w konfiguracjach RAC i zazwyczaj są używane tylko z klastrowanymi instancjami ASM. Preferowane odczyty lustrzane ASM pozwalają na odczyt z ekstenów kopii lustrzanych zamiast z ekstenów dysku podstawowego. Aby skorzystać z zalet tej nowej własności, należy skonfigurować kopię lustrzaną ekstenów, lokalną dla danego odległego węzła klastra. W celu skonfigurowania w danej instancji listy nazw preferowanych awaryjnych grup dyskowych używanych przy dostępie do dysków ASM, należy ustawić nowy parametr `asm_preferred_read_failure_groups`. Format jego wartości przedstawia się następująco: *nazwaGrupyDyskowej.nazwaAwaryjnejGrupy*, gdzie *nazwaGrupyDyskowej* to nazwa grupy dyskowej, do której należy grupa awaryjna, a *nazwaAwaryjnejGrupy* to nazwa preferowanej grupy awaryjnej. W celu wskazania wielokrotnie nazw różnych par *grupaDyskowa.grupaAwaryjna* należy każdą ze zdefiniowanych preferowanych grup odczytu oddzielić znakiem przecinka, tak jak pokazano w poniższym przykładzie:

```
Asm_preferred_read_failure_groups=dgroup1.fdisk2, dgroup2.fdisk2
```

Jeśli ASM nie będzie w stanie dokonać odczytu z preferowanej awaryjnej grupy dyskowej, to będzie kontynuować odczyt z grupy preferowanej, a w następnej kolejności z dowolnej innej zdefiniowanej, dodatkowej awaryjnej grupy. W celu sprawdzenia, czy dany dysk należący do grupy dyskowej jest dyskiem preferowanego odczytu, należy podejrzeć kolumnę `PREFERRED_READ` z widoku `V$ASM_DISK`.

ASM rolling upgrade

Oracle Database 11g wspiera aktualizacje typu rolling upgrade, czyli aktualizowanie w czasie pracy systemu. Wykonywanie aktualizacji typu rolling upgrade z wersji Oracle Database 10g do Oracle Database 11g nie jest wspierane. Kiedy już dokonamy aktualizacji do 11g, to uaktualnienia do wyższych wersji będziemy mogli wykonywać jako te typu rolling upgrade. Częścią składową nowej własności rolling upgrade jest możliwość aktualizacji instancji ASM jako typu rolling upgrade. Przed wykonaniem uaktualnienia rolling upgrade musimy się upewnić, że zależne komponenty zostały uaktualnione wcześniej (przykładowo, przed aktualizacją ASM należy uaktualnić CRS — Cluster Ready Services).

W celu wsparcia aktualizacji rolling upgrade została dodana nowa składnia w Oracle. Polecenie `alter system` zostało wzbogacone o nowy parametr `start rolling migration`. Każda migracja wymaga dostarczenia zbioru różnych parametrów wraz z poleceniem `alter system start rolling migration`, włączając w to numer wersji, numer wydania, numer uaktualnienia, numer portu oraz numer portu uaktualnienia. W sprawie szczegółów dotyczących odpowiedniego formatu polecenia należy odnieść się do dokumentacji.

Po wydaniu polecenia `alter system start rolling migration` ASM najpierw spróbuje ustalić, czy w toku są jakieś operacje równoważenia ASM (*rebalancing*). Jeśli takie operacje trwają, wykonanie polecenia zakończy się błędem. W przypadku gdy w danym momencie nie występują operacje równoważenia, charakterystyka działania klastra ASM będzie wyglądać następująco:

- ◆ W przypadku grup dyskowych możliwe będzie wyłącznie ich, zamontowanie i odmontowanie (nie będzie można wykonywać operacji `alter`, `create`, `drop`).
- ◆ Możliwe będzie otwieranie i zamykanie plików w grupach dyskowych przez bazę danych.
- ◆ Możliwe będzie usuwanie i zmienianie rozmiarów plików.
- ◆ Dostęp do widoków i pakietów stałych będzie ograniczony. Widoki globalne na klastrowanej instancji ASM będą niedostępne.

Po zmodyfikowaniu instancji dla rozpoczęcia migracji należy zamknąć i zaktualizować każdą z instancji, jedną na raz. Kiedy aktualizacja zostanie zakończona, będziemy mogli ponownie uruchomić instancję ASM, a zostanie ona dołączona do klastra. Po zakończeniu wszystkich kroków aktualizacji musimy jeszcze zakończyć sam proces uaktualniania poleceniem `alter system`:

```
Alter system stop rolling migration;
```

Jeśli podczas aktualizacji napotkamy jakieś problemy, możemy po prostu odwrócić proces poprzez wycofanie. Każdorazowo należy zamknąć jeden węzeł ASM, wycofać go do poprzedniej wersji oprogramowania, a następnie uruchomić ponownie. Po zakończeniu wycofywania można zakończyć proces migracji. Zauważmy, że po zakończeniu procesu migracji nie można go już wycofać.



Uwaga

Jak można się było spodziewać, przed przystąpieniem do aktualizacji zalecamy przeczytanie instrukcji aktualizacji oraz upewnienie się, że możliwe jest wykorzystanie przedstawionej procedury.

Wsparcie ASM dla jednostek alokacji o zmiennym rozmiarze

Jak wspomniano w jednym z poprzednich punktów, „Atrybuty grup dyskowych ASM”, ASM wspiera teraz ekstenty (*extent*) o zmiennych rozmiarach, dzięki czemu możliwe jest zmniejszenie wymagań pamięciowych związanych z dużymi plikami dyskowymi ASM i jednocześnie polepszenie wydajności. Obecnie, wraz ze wzrostem rozmiaru plików, Oracle może automatycznie zaalokować wielokrotność ekstentów o zmiennym rozmiarze. Ponieważ maksymalny rozmiar pliku ASM jest zależny od rozmiaru jego ekstentu, oznacza to, że rozmiar naszych plików może przekraczać poprzedni limit 35 TB dla grup dyskowych zewnętrznej redundancji.

Możliwe jest także zdefiniowanie takich grup dyskowych, że pliki w nich utworzone zostaną stworzone z jednostkami alokacji o różnych rozmiarach. Rozmiar jednostki alokacji waha się od 1 do 64 MB. Więcej informacji na temat ustawiania rozmiaru jednostek alokacji można znaleźć w punkcie „Atrybuty grup dyskowych ASM”. Możliwość definiowania różnych jednostek alokacji pozwala grupom dyskowym ASM z redundancją zewnętrzną osiągnąć maksymalny rozmiar 128 TB, w przeciwieństwie do rozmiaru 35 TB osiąganego w Oracle Database 10g.

Nowa rola SYSASM

W Oracle została udostępniona nowa rola SYSASM, z której należy korzystać podczas łączenia się z instancją ASM jako administrator. Roli SYSASM powinno się zacząć używać zamiast SYSDBA, ponieważ połączenia SYSDBA do instancji ASM prawdopodobnie nie będą wspierane w przyszłości. Wymagania dotyczące uwierzytelniania są dla połączeń z rolą SYSASM takie same jak dla połączeń z rolą SYSDBA.

Poniżej prezentujemy przykład połączenia z instancją ASM z użyciem nowej roli SYSASM:

```
[oracle@localhost ~]$ sqlplus / as sysasm
SQL*Plus: Release 11.1.0.3.0 - Beta on Sat Feb 24 19:48:10 2007
Copyright (c) 1982, 2006, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.3.0 - Beta
With the Partitioning, OLAP and Data Mining options
SQL>
```

Można także użyć zwyczajnego polecenia połączenia, jak `connect sys/robert as SYSASM` i podobne.

Nowe polecenia narzędzia asmcmd

Zabawne, jak wielu ludzi, z którymi rozmawiałem, korzysta z ASM, a mimo to nigdy nie używało ani nie słyszało o `asmcmd`. Narzędzie `asmcmd` pojawiło się po raz pierwszy w Oracle Database 10g Release 2, aby ułatwić poruszanie się po dyskach ASM i ich zawartości. W Oracle Database 11g do `asmcmd` zostały dodane dalsze własności, które będą omawiane w tym punkcie. W skład owych własności wchodzi: możliwość archiwizacji i odtwarzania metadanych ASM, nowe polecenie `lddsk`, polecenie `repair` oraz — dla istniejących poleceń `asmcmd` — kilka nowych opcji typu flagi.

Arup radzi...

Wsparcie dla ekstentów o zmiennym rozmiarze jest prawdopodobnie najważniejszym dodatkiem do ASM w Oracle Database 11g. Przed 11g liczba ekstentów, zwłaszcza w ogromnych bazach danych, była po prostu zbyt duża. Skutkowało to tym, że mapy ekstentów (extent maps) były zbyt wielkie, aby można było nimi zarządzać w zbiorniku współdzielonym (shared pool); miało to wpływ na wydajność, jak również powodowało straszliwe błędy ORA-4031 Unable to allocate x bytes in shared pool¹. Dla złagodzenia negatywnych skutków w Oracle Database 10g możliwe było zdefiniowanie większych rozmiarów jednostek alokacji za pomocą parametru z podkreśleniem, tak jak pokazuje przykład:

```
_asm_ausize=16777216
```

Parametr ten, pomimo że z podkreśleniem, jest wspierany przez Oracle i udokumentowany na Meta-Link. Powyższy przykład definiuje 16-megabajtowe jednostki alokacji. W przypadku ogromnych baz danych, powyżej 10 TB, jest to zalecany rozmiar. Domyślna wartość to 1048576 (lub 1 MB), także w Oracle Database 11g. Stan typu „jeden rozmiar do wszystkiego” nie był dobrym rozwiązaniem we wszystkich przypadkach. Jeśli z góry posiadało się wiedzę o tym, do jakich rozmiarów urośnie baza danych, można było śmiało ustawić parametr zawczasu, jednak w wielu sytuacjach oszacowanie przyszłego rozmiaru bazy danych mogło być błędne lub niemożliwe. W Oracle Database 11g problem ten został rozwiązany przez oddzielenie rozmiaru ekstentu od wielkości jednostki alokacji. W 11g jeden ekstent może mieć rozpiętość więcej niż jednej jednostki alokacji. Jego rozmiar może się zwiększać wraz ze wzrostem rozmiaru bazy danych. Pozwala to na utrzymanie rozmiaru mapy obszarów w rozsądnej granicy, co z kolei poprawia wydajność.

Bądź co bądź, oznacza to także, że ekstenty z czasem zostaną na dysku w jakimś stopniu pofragmentowane. W wielu przypadkach jest to nieistotne, nawet jeśli będziemy często usuwać i tworzyć obiekty w bazie danych. Będzie to jednak poważnym problemem w przypadku częstego tworzenia i usuwania plików danych. Jeśli zaobserwujemy spadek wydajności, możemy zdefragmentować dysk, używając operacji `alter diskgroup ... rebalance`.

Archiwizacja i odtwarzanie metadanych grup dyskowych ASM

Narzędzie `asmcmd` dostarcza zestawu nowych poleceń, które pozwalają na archiwizację i odzyskiwanie metadanych związanych ze wszystkimi grupami dyskowymi ASM. Polecenie `md_backup` wykona archiwizację wszystkich lub części metadanych ASM, w zależności od użytych parametrów. Poniżej przedstawiamy przykład użycia polecenia `md_backup` w celu archiwizacji całego repozytorium metadanych ASM.

```
[oracle@localhost ~]$ asmcmd
ASMCMD> md_backup
```

Kiedy polecenie zostanie w całości wykonane, w bieżącym katalogu roboczym zostanie utworzony plik. W moim systemie Linux plik został nazwany `ambr_backup_intermediate_file`, ale w innych systemach operacyjnych jego nazwa może się różnić. Plik archiwizacji jest w znacznej mierze plikiem tekstowym, więc możemy go po prostu czytać. Bardziej zaawansowane użycie polecenia `md_backup` pozwala kontrolować miejsce, w którym plik ma zostać utworzony, wyszczególniać grupy dyskowe mające podlegać archiwizacji, a także przesłaniać różnorakie opcje.

¹ Polska wersja komunikatu błędu to: ORA-04031: nie można przydzielić X bajtów pamięci współużytkowanej — *przypr. tłum.*

Polecenie `md_restore` służy do przywracania danych z archiwizacji grup dyskowych wykonanych za pomocą polecenia `md_backup`. Przywrócenie można wykonać bez wprowadzania żadnych modyfikacji lub wcześniej przesłonić różne ustawienia w poleceniu `md_backup`. Oto przykład wykonania przywracania metadanych:

```
[oracle@localhost ~]$ asmcmd
ASMCMD> md_restore -b ambr_backup_intermediate_file
```

Oba polecenia mogą zostać wykonane poprzez przekazanie ich jako parametru podczas uruchamiania narzędzia `asmcmd`, tak jak pokazano w przykładzie poniżej:

```
[oracle@localhost ~]$ asmcmd md_backup
```

Należy mieć świadomość tego, że polecenie `md_backup` nie nadpisze poprzedniego pliku archiwizacji. Po wykonaniu archiwizacji należy samemu przenieść gdzieś taki plik, aby następna archiwizacja mogła się udać.

Po co właściwie używać poleceń `md_restore` i `md_backup`? Tak jak w bazie danych Oracle, tak i w przypadku ASM istnieje możliwość, że metadane ASM — składowane w grupach dyskowych ASM — ulegną uszkodzeniu (na przykład z powodu uszkodzeń pamięci w systemie operacyjnym). Chociaż zdarza się to (na szczęście) rzadko, w takich przypadkach nasza archiwizacja metadanych ASM może okazać się pomocna. Jeśli używamy ASM, zalecam, aby archiwizacja metadanych dołączyła do naszych rutynowych, codziennych wykonań archiwizacji.

Arup radzi...

Nie potrafię nawet wyrazić użyteczności (i trafności czasu pojawienia się) funkcji archiwizacji/przywracania metadanych. Jak wspomniał Robert, w nawiązaniu do możliwości uszkodzenia metadanych, istnieje także prawdopodobieństwo, że ktoś wykona polecenie `dd` i nadpisze zawartość nagłówka dyskowego. Mnie to spotkało, naprawdę! Kluczem do zrozumienia sytuacji jest fakt, że w nagłówku znajduje się informacja, co jest gdzie zapisane na dysku. Nadpisanie nagłówka usuwa tę informację, ale nie same dane. Jeśli potrafimy przywrócić nagłówek, możemy być też w stanie przywrócić dane za pomocą bazy danych, a zatem zawsze wykonujemy polecenie `md_backup` i przechowujemy wynikowy plik w bezpiecznej lokalizacji.

W niektórych przypadkach polecenie `md_restore` może także pomóc w przywróceniu usuniętych dysków. Kolejnym zastosowaniem jest rozwijanie automatycznego procesu kontroli zmian, za każdym razem gdy dodajemy, zmieniamy lub usuwamy dyski.

Polecenie `lsdsk`

Polecenie `lsdsk` listuje dyski, które widzi ASM. Oto przykład:

```
[oracle@localhost ~]$ asmcmd lsdsk
Path
/oracle/asmdata/asm_dgroup1_01.asm
/oracle/asmdata/asm_dgroup1_02.asm
```

Polecenie `remap`

Polecenie `remap` może zostać użyte w celu naprawy zakresu fizycznych bloków w naszym dysku ASM. Jako parametry przyjmuje ono nazwę grupy dyskowej, nazwę dysku oraz zakres bloków do remapowania. Zauważmy, że podczas gdy polecenie to może przywrócić dysk do użytku

(pod względem fizycznie i logicznie uszkodzonych bloków), nie przywróci nam ono straconych lub uszkodzonych danych. Oto przykład:

```
[oracle@localhost ~]$ asmcmd remap dgroup1 disk_001 2000-4000
```

Nowe opcje poleceń ls, lsct oraz lsdg

Polecenie `ls` ma dwa nowe parametry: `-c` oraz `-g`. Skorzystanie z parametru `-c` skutkuje użyciem widoku `V$ASM_DISKGROUP` jako źródła wydruku wyniku polecenia. Użycie parametru `-g` spowoduje, że źródłem wydruku wyniku polecenia będzie widok `GV$ASM_DISKGROUP`.

Polecenie `lsct` także posiada nowy parametr `-g`, który powoduje czytanie danych z widoku `GV$ASM_CLIENT` zamiast z widoku `V$ASM_CLIENT`.

W końcu polecenie `lsdg`, które także zostało wzbogacone o parametry `-c` i `-g`. Parametr `-c` jako źródła wydruku wyniku polecenia używa widoku `V$ASM_DISKGROUP`, natomiast parametr `-g` — widoku `GV$ASM_DISKGROUP`.

Automatic Memory Management

Firma Oracle wprowadziła wiele zmian we własnościach automatycznego zarządzania pamięcią w Oracle Database 11g. Oracle Automatic Memory Management może teraz automatycznie zarządzać zarówno obszarem SGA (system global area), jak i obszarem PGA (program global area). Kontrolę nad tą nową funkcją zapewniają nowe parametry `memory_target` i `memory_max_target`. Dotyczące jej zmiany zostały także wprowadzone w niektórych narzędziach doradczych (*advisor*). Zostały także dodane nowe widoki. W kolejnych podpunktach omówimy każde z tych zagadnień.

Przegląd Automatic Memory Management

Automatyczne zarządzanie pamięcią zostało dodane do istniejącej struktury Oracle Automated Memory Management i pozwala na użycie różnych parametrów konfiguracyjnych ustawienia pamięci. Automatic Memory Management posiada dwa nowe parametry pamięci: `memory_max_target` i `memory_target`. Rysunek 2.1 ilustruje zależności zachodzące pomiędzy różnymi parametrami pamięci.

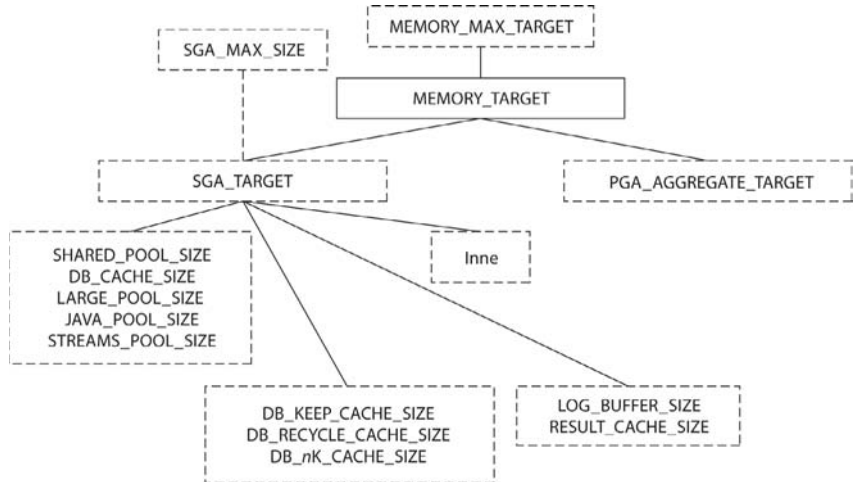
Przyjrzyjmy się bliżej dwóm wspomnianym parametrom.

Parametr `memory_target`

Parametr `memory_target` jest w pewnym sensie kombinacją wartości parametrów `sga_target` i `pga_aggregate_target`. Reprezentuje on całkowitą ilość pamięci, jaką Oracle musi zaalokować dla różnych struktur SGA i PGA. Parametr `memory_target` jest parametrem dynamicznym. Może zostać ustawiony na wartość do wartości (włącznie) parametru `memory_max_target`, który zostanie omówiony jako następny.

Rysunek 2.1.

Zależności pomiędzy parametrami pamięci



Parametr memory_max_target

Parametr `memory_max_target` pozwala na dynamiczną zmianę wartości parametru `memory_target` w granicach `memory_max_target`, tak więc w każdej chwili możemy przystosować sumaryczną ilość pamięci dostępnej dla bazy danych jako całość.

Parametr `memory_max_target` może zostać ustawiony ręcznie albo wyznaczony i domyślnie ustawiony na wartość równą lub większą wartości parametru `memory_target`. Jeśli parametr `memory_target` jest ustawiony, a parametr `memory_max_target` nie, to Oracle domyślnie przypisze temu drugiemu wartość parametru `memory_target`. Jeśli parametr `memory_target` nie jest ustawiony, to domyślną wartością parametru `memory_max_target` będzie 0. W przypadku gdy parametr `memory_target` nie jest ustawiony, ale parametr `memory_max_target` przyjmuje jakąś niezerową wartość, to możemy modyfikować ustawienia pamięci dynamicznie, poprzez zmianę `memory_target` lub innych określonych parametrów pamięci. Rysunek 2.2 ilustruje zależności pomiędzy parametrami pamięci zarządzanej automatycznie.



Uwaga

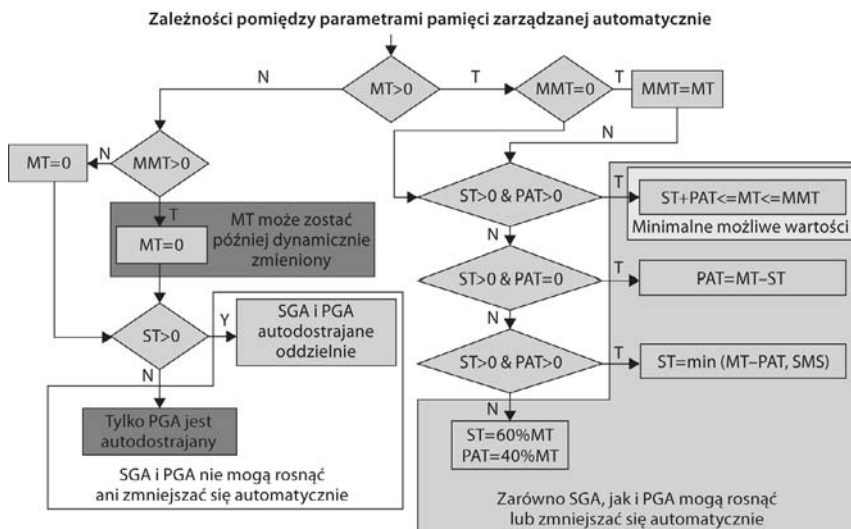
Parametrów `memory_max_target` i `memory_target` nie można używać, jeśli został ustawiony `LOCK_SGA`. Wspomnianych parametrów nie można także stosować w połączeniu z ogromnymi stronami w systemie Linux.

Parametr `memory_target` można skonfigurować podczas aktualizacji lub później. Kiedy konfigurujemy parametr `memory_target`, powinniśmy wziąć pod uwagę bieżące ustawienia parametrów `sga_target` i `pga_aggregate_target`. W większości sytuacji jako wartość `memory_target` będziemy ustawiać sumę rozmiarów zdefiniowanych przez wartości parametrów `sga_target` i `pga_aggregate_target`, zerując następnie te dwa parametry.

Możemy zdecydować się na pozostawienie parametrów `sga_target` i `pga_aggregate_target` z ich bieżącymi wartościami. Robiąc tak, wskażemy minimalne ilości pamięci, jakie powinny zostać zaalokowane na te struktury. W takim przypadku Oracle będzie mógł dla każdej ze struktur zaalokować więcej pamięci, ale nie będzie mógł zmniejszyć dostępnej dla nich ilości pamięci poniżej progu wyznaczonego przez ustawienia powiązanych parametrów. Oznacza to,

Rysunek 2.2.

Zależności pomiędzy parametrami pamięci zarządzanej automatycznie



Uwaga

Ustawianie parametru `memory_target` na niektórych platformach może przyprawić nas o łzy. Przykładowo w systemie Linux należy upewnić się, że `/dev/shm` (lub jego odpowiednik) alokuje ilość pamięci trochę większą niż ta, na którą chcemy ustawić wartość parametru `memory_target`. Niespełnienie powyższego warunku będzie skutkowało następującym błędem²:
ORA-00845: MEMORY_TARGET not supported on this system.

że jeśli przejdziemy podczas aktualizacji na nowe parametry, zostawiając wartości `sga_target` i `pga_aggregate_target` bez zmian, to powinna zostać spełniona następująca nierówność `memory_target >= (sga_target + pga_aggregate_target)`. Nieprawidłowa konfiguracja nie pozwoli na uruchomienie instancji z powodu wystąpienia błędu ORA-00838.

Kolejną rzeczą, o której należy wiedzieć, jest to, że jeśli ustawimy parametr `memory_max_target` i uruchomimy instancję, Oracle po prostu zagarnie ilość wolnej pamięci równą wartości tego parametru. Może to zależeć od systemu operacyjnego — zaobserwowaliśmy to w środowiskach systemów operacyjnych Linux i Windows. Uważajmy więc na potencjalne konsekwencje, jakie pociąga za sobą ustawienie wspomnianego parametru.



Uwaga

Należy zachować ostrożność podczas zmiany ustawień, jeśli przechodzimy na automatyczne zarządzanie pamięcią. Przechodząc na parametry `memory_max_target` i `memory_target`, upewnijmy się, że parametry `sga_target` i `sga_size` nie są ustawione. Pomimo iż jednocześnie ustawianie tych parametrów jest wspierane, należy to robić z ostrożnością, aby uniknąć nieoczekiwanych konsekwencji.

² Polska wersja komunikatu błędu to: ORA-00845: Parametr MEMORY_TARGET nie jest obsługiwany w tym systemie — *przyp. tłum.*

Nowe funkcje w Memory Advisor i nowe widoki

Oracle otrzymał nowy widok `V$MEMORY_TARGET_ADVICE`, który dostarcza wskazówek, jak ustawić parametr `memory_target`. Poniżej przedstawiamy przykładowe zapytanie do tego widoku:

```
SQL> select * from v$memory_target_advice order by memory_size_factor;
MEMORY_SIZE MEMORY_SIZE_FACTOR ESTD_DB_TIME ESTD_DB_TIME_FACTOR VERSION
-----
176          .5                126          1.1443          0
264          .75                110           1                0
352          1                  110           1                0
440          1.25              110           1                0
528          1.5                109           .99              0
616          1.75              109           .99              0
704          2                  109           .99              0
```

Przykład pokazuje, że zwiększenie ilości pamięci nie będzie miało pozytywnego wpływu na naszą bazę danych. Bieżące ustawienie `memory_size` to 352 MB (jeśli `MEMORY_SIZE_FACTOR` wynosi 1, to jest to aktualny rozmiar), co wydaje się być większą wartością, niż potrzeba. Zwiększenie jej o 50 procent, do 528 MB (`MEMORY_SIZE_FACTOR` równy 1.5), będzie skutkowało bardzo nieznaczną poprawą wydajności, jak pokazuje różnica pomiędzy wartościami kolumny `ESTD_DB_TIME` 110 i 109. Zwróćmy uwagę, że widok nie będzie zawierał żadnych danych, dopóki automatyczne zarządzanie pamięcią nie zostanie włączone (przynajmniej poprzez ustawienie parametru `memory_target`).



Uwaga

Dane w omawianych widokach zależą od migawek (*snapshot*) AWR. Jeśli wyłączyliśmy AWR lub dopiero co stworzyliśmy bazę danych, widoki mogą być puste. Wyniki zapytań do tych widoków w dużym stopniu zależą także od obciążenia i wykorzystania systemu. Wynik będzie znacząco różny dla bezczynnego systemu w porównaniu z systemem o wysokim obciążeniu i zmienności.

Kolejny widok, `V$MEMORY_DYNAMIC_COMPONENTS`, zapewnia wgląd w różne komponenty pamięci bazy danych i ich rozmiary, ustalane przez Oracle'owe udogodnienie automatycznego zarządzania pamięcią. Poniżej przedstawiamy przykładowe zapytanie do tego widoku, pokazujące obecnie skonfigurowane komponenty pamięci. Zauważmy, że oprócz zwyczajnych pul pamięci (*pools*) są tam także wpisy dotyczące SGA Target i PGA Target:

```
SQL> select component, current_size curr_size, min_size, max_size,
user_specified_size uss, granule_size gs from v$memory_dynamic_components
where current_size > 0;
COMPONENT          CURR_SIZE  MIN_SIZE  MAX_SIZE  USS        GS
-----
shared pool        67108864  62914560  67108864  0          4194304
large pool         4194304   0         4194304   0          4194304
java pool          4194304   4194304   4194304   0          4194304
SGA Target         218103808 218103808 218103808 0          4194304
DEFAULT buffer    134217728 134217728 138412032 83886080   4194304
cache
Shared IO Pool     8388608   8388608   8388608   8388608    4194304
PGA Target         150994944 150994944 150994944 0          4194304
```

W jaki sposób Oracle zarządza pamięcią?

Przeglądając widok `V$MEMORY_RESIZE_OPS`, możemy monitorować, w jaki sposób Oracle zarządza pamięcią. Widok zawiera listę ostatnich 800 żądań zmiany rozmiaru SGA obsługiwanych przez Oracle. Oto przykład:

```
SQL> select parameter, initial_size, target_size, start_time
      from v$memory_resize_ops
      where initial_size > 0 and final_size > 0
      order by parameter, start_time;
PARAMETER          INITIAL_SIZE TARGET_SIZE START_TIME
-----
db_cache_size      121634816   117440512  04/28/2007 19:33:45
db_cache_size      121634816   117440512  04/28/2007 19:33:45
db_cache_size      117440512   113246208  04/28/2007 19:33:45
db_cache_size      113246208   109051904  04/28/2007 19:33:45
db_cache_size      109051904   104857600  04/28/2007 19:34:10
db_cache_size      104857600   100663296  04/28/2007 19:34:33
db_cache_size      100663296   96468992   04/28/2007 19:34:35
pga_aggregate_target 125829120   335544320  04/28/2007 19:39:20
shared_pool_size    58720256    62914560   04/28/2007 19:33:45
shared_pool_size    54525952    58720256   04/28/2007 19:33:45
shared_pool_size    62914560    67108864   04/28/2007 19:34:10
shared_pool_size    67108864    71303168   04/28/2007 19:34:33
shared_pool_size    71303168    75497472   04/28/2007 19:34:35
```

Jak wskazuje powyższy wynik zapytania, Oracle był bardzo zajęty, dokonując zmian w pamięci podręcznej bazy danych i w zbiorniku współdzielonym (*shared pool*)!

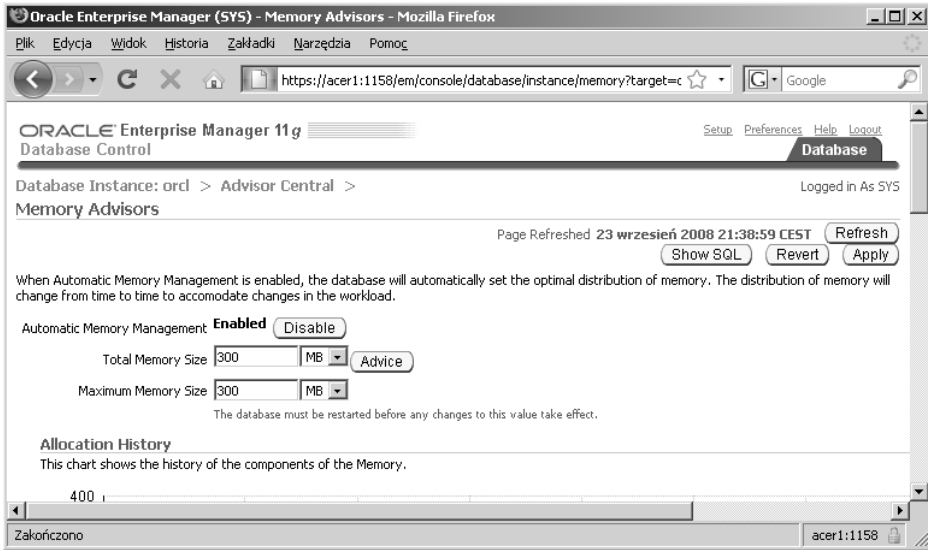
Automatic Memory Management i OEM

Oracle Enterprise Manager (OEM) całkowicie wspiera automatyczne zarządzanie pamięcią. Na stronie domowej OEM kliknijmy w umieszczoną na górze zakładkę *Server*, a następnie wybierzmy *Memory Advisors*. Z poziomu tej strony możemy włączyć lub wyłączyć automatyczne zarządzanie pamięcią, ustawić całkowite i maksymalne rozmiary parametrów pamięci i przejrzeć historię alokacji pamięci. Rysunek 2.3 przedstawia przykładową stronę narzędzia Memory Advisor.

Automatic Memory Management posiada interfejs w Memory Advisor (jest to graficzna nakładka na widok `V$MEMORY_TARGET_ADVICE`). Możemy tam sprawdzić, jak powinniśmy zaalokować pamięć dla instancji.

Przejsięcie na Automatic Memory Management

Przejsięcie na automatyczne zarządzanie pamięcią można dokonać ręcznie lub z poziomu OEM. Włączenie usługi Automatic Memory Management jest także możliwe z poziomu narzędzia Database Configuration Assistant (DBCA) podczas konfigurowania nowej bazy danych. Czytelnicy uaktualniający swoje oprogramowanie do Oracle Database 11g mogą mieć wątpliwości, czy warto przejść na Automatic Memory Management. Jak ze wszystkim co nowe, podszedłbym to tego z ostrożnością, testując wszystko dokładnie w środowisku deweloperskim. Jeśli



Rysunek 2.3. Strona narzędzia OEM Memory Advisor

istnieje nagle potrzeba przejścia, to trzeba to zrobić, ale jeśli można poczekać, to nie narażałbym produkcyjnego systemu na ryzyko. W przypadku nowych baz danych, utworzonych w 11g, prawdopodobnie zdecydowałbym się na konfigurację z Automatic Memory Management i starannie ją przetestował.

Nowe własności ADDM

Oracle Database 11g oferuje wiele nowych własności ADDM (Automatic Database Diagnostic Monitor), w skład których wchodzi:

- ♦ nowe widoki ADDM,
- ♦ ADDM uwzględniający istnienie RAC,
- ♦ nowy pakiet dbms_addm do zarządzania ADDM,
- ♦ klasyfikacja,
- ♦ dyrektywy.

Przyjrzyjmy się teraz bliżej każdej z tych nowych własności.

Nowe widoki ADDM

Poniższa tabela prezentuje nowe widoki powiązane z ADDM, które zostały dodane w Oracle Database 11g.

Nazwa widoku	Opis
DBA_ADDM_TASKS	Dostarcza historyczną reprezentację wcześniejszych zadań ADDM.
DBA_ADDM_INSTANCES	Wyświetla informacje o zadaniach ADDM na poziomie instancji.
DBA_ADDM_FINDINGS	Dostarcza dodatkowych informacji dla różnych widoków doradczych.
DBA_ADDM_FDG_BREAKDOWN	Zapewnia informację o udziale poszczególnych znalezisk (<i>finding</i>) w każdej z instancji.
DBA_ADDM_SYSTEM_DIRECTIVES	
DBA_ADDM_TASK_DIRECTIVES	

Zauważmy, że każdy z widoków DBA_ ma odpowiadający mu widok USER_, za wyjątkiem widoku DBA_ADDM_SYSTEM_DIRECTIVES.

Współistnienie RAC i ADDM

Istnieje co najmniej kilka istotnych zagadnień wydajnościowych specyficznych dla RAC. Przed Oracle Database 11g jedyną metodą analizy komponentu klastra o konfiguracji RAC było przeglądanie znacznej liczby widoków GV\$. Oracle Database 11g dostarcza nowego poziomu analizy ADDM nazwanej Database ADDM. Z Database ADDM narzędzie Oracle ADDM jest teraz świadome obecności RAC i włącza analizę klastra RAC do ogólnego procesu analizy ADDM bazy danych.

Podobnie jak w przypadku Instance ADDM (czyli ADDM dla pojedynczej instancji bazy danych), nowe procesy Database ADDM (czyli ADDM dla całej bazy danych) są uruchamiane domyślnie po wykonaniu każdej migawki AWR (*Automatic Workload Repository*). Database ADDM może zapewnić raportowanie wystąpienia wymienionych niżej stanów:

- ◆ Nadmierne używanie globalnych zasobów takich jak operacje wejścia/wyjścia i blokady globalne.
- ◆ Wysoko obciążające zapytania SQL i wystąpienia gorących bloków (*hot block*).
- ◆ Nadmierny ruch typu interconnect w globalnej pamięci podręcznej.
- ◆ Opóźnienia łączy sieciowych.
- ◆ Odchylenia czasów odpowiedzi instancji.

Włączenie Database ADDM odbywa się poprzez wywołanie procedury `set_default_task_` ↪parameter z dostarczanego przez Oracle pakietu PL/SQL `dbms_advisor`. Pozwala to ustalić wartości parametru `instances` mające wskazać instancje, które powinny podlegać analizie Database ADDM. Poniższa tabela daje wgląd w różne opcje, których można użyć w przypadku parametru `instances`.

Opcja	Uruchamiany tryb analizy ADDM
UNUSED	Wyłącza Database ADDM dla wszystkich instancji.
Lista instancji rozdzielonych znakami przecinka	Database ADDM zostanie wykonane tylko dla wymienionych instancji.
ALL	Włącza Database ADDM dla wszystkich instancji.

Oto kilka przykładów ustawiania wartości parametru `instances`:

```
-- Wyłącz Database ADDM dla wszystkich instancji
Exec dbms_advisor.set_default_task_parameter('ADDM', 'INSTANCES', 'UNUSED');
-- Skonfiguruj Database ADDM dla instancji 1. i 3.
Exec dbms_advisor.set_default_task_parameter('ADDM', 'INSTANCES', '1,3');
-- Skonfiguruj Database ADDM dla wszystkich instancji
Exec dbms_advisor.set_default_task_parameter('ADDM', 'INSTANCES', 'ALL');
```

Rezultat takiego dodatkowego poziomu analizy będzie widoczny w co najmniej kilku miejscach w OEM, na przykład na podstronie *Cluster* strony analizy wydajności. Stamtąd można przejść do bardziej szczegółowych stron. Możliwe jest także raportowanie ręczne z wykorzystaniem nowego pakietu `dbms_addm`, które zostanie omówione w kolejnym podpunkcie.

Zarządzanie ADDM poprzez `dbms_addm`

W celu wsparcia pracy administratora bazy danych w zarządzaniu Oracle ADDM, w Oracle Database 11g został wprowadzony nowy pakiet `dbms_addm`. Pakiet ten umożliwia zlecenie rozpoczęcia analizy ADDM, ukierunkowanie analizy, wydrukowanie raportu lub usunięcie wyniku poprzedniej analizy.

Niektóre najpowszechniej używane programy pakietu `dbms_addm` wraz z ich przeznaczeniem zostały wymienione w poniższej tabeli (nie uwzględnia ona dyrektyw, ponieważ zostaną pokrótce omówione później):

<code>analize_db</code>	Planuje analizę ADDM bazy danych opartą na zakresie dwóch dostarczonych migawek.
<code>analize_inst</code>	Planuje analizę ADDM instancji w oparciu o dwie dostarczone migawki.
<code>analize_partial</code>	Planuje częściową analizę ADDM w oparciu o dwie dostarczone migawki oraz listę instancji.
<code>delete</code>	Usuwa wybrane zadanie ADDM.
<code>get_report</code>	Dostarcza domyślny raport ADDM dla wybranych zadań ADDM.

Poniżej prezentujemy przykład użycia pakietu `dbms_addm` w celu wykonania analizy ADDM bazy danych i raportowania jej wyniku:

```
-- Przedstaw listę ważnych migawek z ostatnich 4 godzin
select instance_number, snap_id
from wrm$.snapshot
where end_interval_time < systimestamp - interval '4' HOUR
order by 1,2;
```



```

INSTANCE_NUMBER SNAP_ID
-----
1                24
2                23
2                25
2                26

Var tname varchar2(60);
BEGIN
  :tname:='Zadanie ADDM Database';
  dbms_addm.analyze_db(:tname, 25, 26);
END;
/
set long 1000000
Spool /tmp/dbreport.rpt
SELECT dbms_addm.get_report(:tname) FROM dual;
spool off

```

Wynikowy raport posiada w ogólności taki sam format jak standardowy raport ADDM, który można otrzymać w Oracle Database 10g (wykonując skrypt *addm rpt.sql*). Wynik analizy ADDM można usunąć przy pomocy `dbms_addm.delete`, tak jak pokazano poniżej:

```
exec dbms_addm.delete('Zadanie ADDM Database');
```

Klasyfikacja znalezisk

Framework Oracle Advisor został wprowadzony w Oracle Database 10g. ADDM jest właśnie przykładem doradcy (advisor). Framework wyróżnia symptomy, problemy, ostrzeżenia i informacyjne typy znalezisk (odpowiednie wartości kolumny `TYPE` widoku `dba_advisor_findings` to `SYMPTOM`, `PROBLEM`, `WARNING`, `INFORMATION`). Nazwy znalezisk (*finding name*) zostały dodane do frameworku Oracle Advisor dopiero w Oracle Database 11g. Dostarczają one dodatkowej informacji, która pomaga zaklasyfikować otrzymane ustalenia do jakiejś kategorii. Przykładowym znaleziskiem może być ustalenie, że wykorzystanie procesora jest zbyt wysokie, tak jak pokazuje przykład:

```

SQL> select task_name, finding_name,
type, impact_type from dba_advisor_findings
where rownum < 2;

```

TASK_NAME	FINDING_NAME	TYPE	IMPACT_TYPE
ADDM:2209966315_1_3	Wykorzystanie CPU	PROBLEM	Czas pracy bazy danych w mikrosekundach.

Nowa kolumna w Oracle Database 11g — `FINDING_NAME` — klasyfikuje znalezisko do konkretnej kategorii, w naszym przypadku do kategorii wykorzystania procesora („Wykorzystanie CPU”). Kolumna ta została dodana także do widoku `USER_ADVISOR_FINDINGS`.

Taka nowa klasyfikacja znalezisk może posłużyć wykonaniu dodatkowej analizy, tak jak w poniższym przypadku, gdzie staramy się sprawdzić, czy problem nadmiernego wykorzystania procesora nasila się o jakichś określonych porach (np., jak w naszym przykładzie, w okolicach południa):

```

SQL> select to_char(execution_end, 'hh24') hour , count(*)
from dba_advisor_findings a, dba_advisor_tasks b
where finding_name='Wykorzystanie CPU'
and a.task_id=b.task_id
group by to_char(execution_end, 'hh24')
order by 1;
HO COUNT(*)
-----
04          1
05          1
07          1
11          3
12          3
13          2
15          1
16          2
17          1

```

Wyróżniamy około 80 różnych kategorii znalezisk, które można zobaczyć w widoku DBA_ADVISOR_FINDING_NAMES.

Dyrektywy

Gdy używamy ADDM, pożądane może być ukierunkowanie analizy na ignorowanie wystąpienia niektórych stanów. Przykładowo, jeśli słabe strony schematu ROBERT są dobrze znane, nie obejmowanie go analizą ADDM będzie miało sens. W celu wykluczenia różnych analiz ADDM i zgłaszania różnych znalezisk możemy ustalić dyrektywy. Mogą one zostać przypisane do wybranych zadań ADDM lub ustawione jako dyrektywy systemowe. Dyrektywy można ustawiać z wiersza poleceń lub wewnątrz OEM. Następujące podpunkty omawiają tworzenie, usuwanie i przykłady użycia dyrektyw.

Tworzenie dyrektyw

Procedury służące do ustawiania dyrektyw znajdują się w poniższej tabeli:

Typ	Nazwa procedury	Opis
Dyrektywa: wstaw znalezisko	dbms_addm. insert_finding_directive	Ogranicza raport ADDM do specyficznych kategorii znalezisk (więcej informacji na ten temat znajdziemy we wcześniejszym podpunkcie tego rozdziału „Klasyfikacja znalezisk”).
Dyrektywa: wstaw parametr	dbms_addm. insert_parameter_directive	Tworzy dyrektywę, która zapobiega sugerowaniu przez ADDM działań mających na celu zmianę wartości konkretnych parametrów systemowych (v\$parameter).
Dyrektywa: wstaw segment	dbms_addm. insert_segment_directive	Tworzy dyrektywę, która zapobiega sugerowaniu przez ADDM działań dotyczących konkretnych właścicieli, segmentów, podsegmentów lub konkretnych numerów obiektów.

Typ	Nazwa procedury	Opis
Dyrektywa: wstaw SQL	dbms_addm. insert_sql_directive	Tworzy dyrektywę, która zapobiega sugerowaniu przez ADDM działań dotyczących zapytań SQL o konkretnych identyfikatorach. Dalsza filtracja umożliwia ograniczenie zapytania SQL do minimalnej liczby aktywnych sesji lub minimalnego czasu odpowiedzi mierzonego w mikrosekundach.

Usuwanie dyrektyw

Jak pokazuje poniższa tabela, pakiet dbms_addm dostarcza procedur służących do usuwania dyrektyw:

Typ	Nazwa procedury usuwającej
Dyrektywa: usuń znalezisko	dbms_addm.delete_finding_directive
Dyrektywa: usuń parametr	dbms_addm.delete_parameter_directive
Dyrektywa: usuń segment	dbms_addm.delete_segment_directive
Dyrektywa: usuń SQL	dbms_addm.delete_sql_directive

Ustalanie, czy dyrektywy są zdefiniowane

W wielu widokach można znaleźć nową kolumnę FILTERED, która wskazuje, czy konkretny wiersz danego widoku podległ filtracji przez dyrektywę. Do widoków zawierających kolumnę FILTERED należą:

- ◆ DBA i USER_ADVISOR_FINDINGS,
- ◆ DBA i USER_ADVISOR_RECOMMENDATIONS,
- ◆ DBA i USER_ADVISOR_ACTIONS.

Przykłady używania dyrektyw

Przykładową sytuacją, w której chcielibyśmy posłużyć się dyrektywami, może być przypadek, gdy nie chcemy, aby ADDM kiedykolwiek raportował na temat schematu ROBERT. W poniższym przykładzie wykluczamy ten schemat i uruchamiamy raportowanie:

```
var tname VARCHAR2(60);
var inst_num number;
BEGIN
-- Uruchomienie tylko dla bieżącej instancji
select instance_number into :inst_num from v$instance;
-- Nazwanie analizy
:tname := 'moja_analiza_instancji';
-- Stworzenie zadania
DBMS_ADVISOR.CREATE_TASK('ADDM', :tname);
-- MiGawka rozpoczęcia analizy
DBMS_ADVISOR.SET_TASK_PARAMETER(:tname, 'START_SNAPSHOT', 242);
-- MiGawka zakończenia analizy
DBMS_ADVISOR.SET_TASK_PARAMETER(:tname, 'END_SNAPSHOT', 243);
```

```
-- Przypisanie instancji do zadania
DBMS_ADVISOR.SET_TASK_PARAMETER(:tname, 'INSTANCE', :inst_num);
-- Ustawienie dyrektywy
DBMS_ADVISOR.INSTER_SEGMENT_DIRECTIVE(:tname, 'Segment directive ID', 'ROBERT');
-- Uruchomienie zadania
DBMS_ADVISOR.EXECUTE_TASK(:tname);
END;
/
-- Raportowanie zadania
set long 1000000
select dbms_addm.get_report('moja_analiza_instancji') from dual;
exec dbms_addm.delete_segment_directive(NULL, 'Segment directive ID');
exec dbms_addm.delete('moja_analiza_instancji');
```

Możemy także wygenerować raport ADDM za pomocą procedury `dbms_advisor.get_task_report`, nakazując zignorowanie zbioru ograniczeń, tak jak pokazano poniżej:

```
select dbms_advisor.get_task_report('moja_analiza_instancji', 'TEXT', 'ALL') from dual;
```

Nowe własności AWR

Oracle Database 11g ulepszył repozytorium Oracle Database Automated Workload Repository (AWR), które zostało wprowadzone w Oracle Database 10g. Nowe własności czynią zarządzanie bazą danych jeszcze prostszym. W skład nowych lub ulepszonych własności wliczamy:

- ♦ zmianę domyślnej retencji migawek AWR,
- ♦ nowe własności linii bazowych AWR,
- ♦ adaptacyjne progi metryk.

Domyślna retencja migawek AWR

Domyślnie Oracle Database 11g zachowuje teraz informację o migawkach AWR przez osiem dni (w przeciwieństwie do siedmiu dni, jak poprzednio). Domyślne zachowanie można zawsze zmienić. Nowa wartość będzie ustalona tylko dla nowych baz danych. Uaktualnione bazy danych utrzymają takie wartości retencji AWR, jakie zostały dla nich wcześniej określone.

Nowe własności linii bazowych AWR

Oracle Database 11g konsoliduje poprzednie własności linii bazowych przedstawionych w Oracle Database 10g. Te skonsolidowane własności określamy terminem **AWR Baseline**. Ponadto, oprócz różnych własności dostępnych w Oracle Database 10g, w Oracle Database 11g zostały dodane nowe, czyniące linie bazowe AWR jeszcze bardziej przydatnymi administratorowi bazy danych. W skład tych własności wchodzi:

- ♦ nowe typy linii bazowych AWR,
- ♦ adaptacyjne progi.

Nowe typy AWR Baseline

Oracle Database 11g wprowadza trochę zawirowań w AWR Baseline. W ich skład wchodzi:

- ◆ **Linie bazowe okien kroczących.** Przesuwne linie bazowe. Zazwyczaj są oparte na całokształcie danych statystycznych dostępnych w AWR.
- ◆ **Pojedyncze linie bazowe.** Pozwalają na zdefiniowanie linii bazowej, która ma zostać uchwycona w danym, pojedynczym okresie czasu w przyszłości.
- ◆ **Powtórzeniowe linie bazowe.** Pozwalają na zdefiniowanie linii bazowej mającej być pochwytywaną w powtarzających się okresach czasu w przyszłości.

Kolejne podpunkty omówią wszystkie nowe, wymienione tu typy linii bazowych. Rozważymy także szablony linii bazowych, widoki danych słownikowych powiązane z tymi szablonami i usuwanie szablonów.

Linie bazowe okien kroczących

Nową własnością w Oracle Database 11g są linie bazowe okien kroczących (*moving window baseline*). Linia bazowa okna kroczącego nosi nazwę SYSTEM_MOVING_WINDOW, a szczegóły jej dotyczące można znaleźć w widoku WRM\$_BASELINE. Okres rozpoczęcia i zakończenia okna linii bazowej SYSTEM_MOVING_WINDOW odnosi się do czasu pomiędzy najwcześniejszą i najpóźniejszą dostępną migawką w Oracle Database 11g. W wyniku tego okres czasu reprezentowany przez tę linię bazową jest kontrolowany wartością opcji retencji w AWR i stale się przesuwa (stąd sprytna nazwa: linia bazowa okna kroczącego).

Domyślny rozmiar okna może zostać zmieniony poprzez wywołanie procedury `dbms_workload_repository.modify_baseline_window_size`, tak jak pokazuje przykład:

```
exec dbms_workload_repository.modify_baseline_window_size ( -
window_size => 30);
```

Wartość parametru `window_size` musi wynosić nie więcej dni niż całkowita wartość liczby dni parametru retencji AWR, w przeciwnym przypadku otrzymamy błąd ORA-13541.

Przesuwnych linii bazowych można używać w OEM, aby porównać bieżące statystyki ze statystykami linii bazowej, co ma ułatwić stwierdzenie, jak radzi sobie nasza baza danych w kwestii wydajności. Przykładowo, strona wydajności w OEM pozwala na zdefiniowanie linii bazowej, z którą chcemy porównać raportowane statystyki. Można użyć systemowej przesuwnej linii bazowej, ręcznie zdefiniowanej linii bazowej lub nie używać żadnej.

Linia bazowa okna kroczącego kontynuuje odzwierciedlanie bieżącej wydajności naszego systemu. Dane tej linii bazowej są świeże, podczas gdy ręcznie zdefiniowane linie bazowe z czasem staną się nieaktualne i przestaną odzwierciedlać dokładny stan linii bazowej naszej bazy danych. Prawdopodobnie dobrym pomysłem jest, zdając sobie z tego sprawę, utrzymać kilka ręcznie zdefiniowanych linii bazowych w celach porównawczych, aby upewnić się, że wydajność naszej bazy danych ukradkiem nie spada.

Pojedyncze linie bazowe

Pojedyncze linie bazowe (*single baseline*) pod pewnymi względami przypominają linie bazowe w Oracle Database 10g; można zdefiniować pojedynczą linię bazową w oparciu o istniejący zbiór migawek AWR. Jednak pojedyncze linie bazowe pozwalają także na zdefiniowanie linii bazowej dla okresu czasu w przyszłości. Przykładowo, jeśli mamy jakieś specjalne przetworzenie, które występuje nieregularnie, ale wiemy, że będzie miało miejsce w ten weekend, możemy wykreować pojedynczą linię bazową, aby zaplanować jej wygenerowanie w przyszłości.

Pojedynczą linię bazową można skonfigurować za pomocą OEM. Na stronie domowej OEM Database należy wybrać zakładkę *Server*. Opcja zatytułowana *AWR Baselines* będzie widoczna pod *Statistics Management*. Z poziomu tej strony można utworzyć pojedynczą linię bazową. OEM zażąda od nas zdefiniowania jej czasu rozpoczęcia i czasu zakończenia lub podania zakresu migawek AWR.

Jeśli tworzymy pojedynczą linię bazową w oparciu o wcześniejsze migawki AWR, nie jest tworzony żaden jej szablon. Jednak gdy tworzymy pojedynczą linię bazową w oparciu o okres czasu w przyszłości, jest tworzony szablon w celu zaplanowania uruchomienia tej linii bazowej. Dopóki szablon linii bazowej nie zostanie wykonany, nie będzie wykazywany na liście linii bazowych AWR w OEM. Za pomocą parametru *expiration* można zdefiniować datę wygaśnięcia linii bazowej. Czas wygaśnięcia jest oparty na parametrze *end_time* stworzonej linii bazowej.

Procedury PL/SQL `dbms_workload_repository.create_baseline_template` używamy w celu ręcznego stworzenia pojedynczej linii bazowej. W poniższym przykładzie tworzymy pojedynczą linię bazową dla przyszłego czasu (30 września) i ustawiamy czas wygaśnięcia na 30 dni od wartości *end_time* tej linii bazowej.

```
Alter session set nls_date_format='yyyy-mm-dd hh24:mi:ss';
BEGIN
  DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE_TEMPLATE (
    start_time => '2008-09-30 13:00:00',
    end_time => '2008-09-30 14:00:00',
    baseline_name => 'baseline_092008',
    template_name => 'template_092008', expiration => 30);
END;
/
```

Wyniki wykonania linii bazowych można przeglądać na stronie OEM *AWR Baselines*, klikając na wybraną linię bazową. Można je także przejrzeć, posługując się różnymi widokami danych słownikowych, włączając w to widoki `DBA_HIST_BASELINE` oraz `DBA_HIST_BASELINE_DETAILS`. Można także wykorzystać ręczne raportowanie, używając skryptu `awrddrpt.sql`.

Arup radzi...

Powtórzeniowe linie bazowe są niezwykle przydatne przy rozwiązywaniu problemów wydajnościowych powiązanych z konkretnymi momentami lub przedziałami czasu. Przykładowo, wyobraźmy sobie hurtownię danych, w której proces ETL (*Extract, Transform, Load*) odbywa się w nocy, a raportowanie za dnia. Powtórzeniowa linia bazowa, której wykonanie powtarza się każdej nocy w określonych interwałach czasu, pomaga ustalić jakiś wzorec na przestrzeni dni. Badając raporty AWR każdej z linii bazowych, można stwierdzić, czy występowanie problemów wydajnościowych podlega jakiemuś wzorcowi, co z kolei jest przydatne w procesie ich rozwiązywania.

Powtórzeniowe linie bazowe

Uruchamianie powtórzeniowych linii bazowych (*repeating baseline*) może zostać zdefiniowane na podstawie regularnych, stałych interwałów czasowych. Przykładowo, jeśli istnieje proces raportowania wykonywany co sobotę od godziny 18:00 do 23:00, możemy chcieć stworzyć powtórzeniową linię bazową uruchamianą w tym okresie czasu. Może się to przydać w śledzeniu całkowitej wydajności przetwarzania w dłuższym czasie lub przy rozwiązywaniu problemów w istotnych procesach przetwarzania, które nagle doświadczyły problemów wydajnościowych.

Powtórzeniową linię bazową można skonfigurować za pomocą OEM. Na stronie domowej OEM *Database* należy wybrać zakładkę *Server*. Opcja zatytułowana *AWR Baselines* będzie widoczna pod *Statistics Management*. Z poziomu tej strony można utworzyć powtórzeniową linię bazową. OEM zażąda od nas zdefiniowania jej czasu rozpoczęcia i czasu zakończenia, jak również częstotliwości (dziennej, miesięcznej i tak dalej). Będzie także trzeba podać datę/czas jej wygaśnięcia (powtórzeniowe linie bazowe ostatecznie podlegają wygaśnięciu). W końcu OEM umożliwi zdefiniowanie, jak długo linia bazowa powinna zostać zachowana (retencja), a następnie stworzy dla nas powtórzeniową linię bazową. Po utworzeniu takiej linii bazowej Oracle utworzy reprezentujący ją szablon. Więcej na temat szablonów linii bazowych dowiemy się w następnym podpunkcie. Wykonanie każdej z linii bazowych będzie skutkowało pojawieniem się takiego wykonania na stronie OEM *AWR Baselines*.

Procedury PL/SQL `dbms_workload_repository.create_baseline_template` używamy w celu ręcznego stworzenia powtórzeniowej linii bazowej. Poniższy przykład ilustruje stworzenie takiej, która jest wykonywana w każdą niedzielę o godzinie 13:00 przez 3 godziny. Ta linia bazowa wygaśnie po 30 dniach, rozpocznie się 2008-05-31, a ostatni raz wykona się w ostatnią niedzielę przed 2008-12-31.

```
BEGIN
DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE_TEMPLATE (
  day_of_week => 'sunday', hour_in_day => 13,
  duration => 3, expiration => 30,
  start_time => '2008-05-31 17:00:00',
  end_time => '2008-12-31 20:00:00',
  baseline_name_prefix => 'baseline_Sunday_reports_',
  template_name => 'template_Sunday_reports');
END;
/
```

Szablony linii bazowych AWR

Szablony linii bazowych AWR są tworzone dla powtórzeniowych linii bazowych. Szablony te są wykorzystywane za każdym razem, kiedy uruchamiana jest zaplanowana linia bazowa. Szablony można przeglądać w OEM, gdzie znajdują się różne informacje na temat stworzonych powtórzeniowych linii bazowych.

Szablony linii bazowych AWR i słowniki danych

Informacje na temat szablonów linii bazowych są dostępne w widoku danych słownikowych `DBA_HIST_BASELINE_TEMPLATE`. Oto przykład wykorzystania wspomnianego widoku:

```
SQL> select template_name, template_type from dba_hist_baseline_template;
TEMPLATE_NAME          TEMPLATE_
-----
robert_test_002        SINGLE
template_070526        SINGLE
template_Sunday_reports REPEATING
test_repeating          REPEATING
```

Usuwanie szablonów linii bazowych

OEM dostarcza łatwego sposobu usuwania szablonów linii bazowych. Na stronie domowej OEM *Database*, w zakładce *Server*, należy wybrać *AWR Baselines*. Ze strony *AWR Baselines* wybieramy link do *AWR Baseline Templates*. W końcu na stronie *AWR Baseline Templates* widzimy status i konfigurację istniejących szablonów linii bazowych i możemy je usunąć.

Ręczne usuwanie istniejących szablonów linii bazowych odbywa się z wykorzystaniem procedury PL/SQL `dbms_workload_repository.drop_baseline_template`, jak pokazano poniżej:

```
BEGIN
  DBMS_WORKLOAD_REPOSITORY.DROP_BASELINE_TEMPLATE (
    template_name => 'template_Sunday_reports');
END;
/
```

Adaptacyjne progi metryk

Ustalanie progów alarmowych w OEM odbywa się trochę na chybił trafił. Metryki wydajności mogą być trudne do zdefiniowania i różnić się w zależności od obciążenia. W przypadku nowych baz danych ustalenie progów jest podwójnie utrudnione, gdyż nie dysponujemy żadną historyczną informacją, na której można się oprzeć. Dla systemów posiadających historię obciążenia linie bazowe mogą posłużyć takiemu zdefiniowaniu metryk, aby dokładniej odzwierciedlały one stan systemu.

W Oracle Database 11g adaptacyjne progi metryk (*adaptive metric threshold*) pozwalają wnieść się ponad to, ponieważ automatycznie ustalą progi, używając do tego linii bazowych AWR. OEM ułatwia także zaaplikowanie adaptacyjnych progów — wystarczy wykonać zaledwie kilku kliknięć. Kolejną korzyścią płynącą z używania adaptacyjnych progów metryk jest to, że wraz ze zmianą obciążenia systemu progi alarmowe ewoluują w celu odzwierciedlenia bieżącego stanu naszej bazy danych.

Na stronie OEM *Baseline Metric Thresholds* (link dostępny jest na stronie *AWR Baselines* omówionej w poprzednim punkcie książki) możemy zlecić Oracle wykonanie szybkiej konfiguracji progów. Bazując na rodzaju obciążenia najczęściej występującego w naszym systemie (system przetwarzania transakcyjnego OLTP, hurtownia danych lub system zmiennego obciążenia), OEM zdefiniuje za nas progi metryk. Raz ustalone początkowe progi można edytować według potrzeb z poziomu strony *Basic Metric Thresholds*.

Progi generowane automatycznie przystosują się do danego obciążenia systemu, tak jak wskazują linie bazowe wykorzystane do ustalenia tych progów. Zatem jeśli obciążenie raportowaniem wzrasta wieczorami, kiedy zaledwie kilku użytkowników online korzysta z bazy, wartość

metryki aktywnych sesji będzie prawdopodobnie niższa pod koniec niż w ciągu dnia. Wraz ze zmianą w czasie przesuwnej linii bazowej AWR progi alarmowe dowolnej z metryk adaptacyjnych mogą ulec zmianie. Na stronie OEM *Baseline Metric Thresholds* można sprawdzić, czy dana metryka jest metryką adaptacyjną.

AutoTask — planowanie zautomatyzowanych zadań utrzymania

System Oracle Database 10g wprowadził nowe zautomatyzowane zadania utrzymania, takie jak automatyczne zbieranie statystyk. Oracle Database 11g dodaje do infrastruktury Oracle nowy komponent zwany AutoTask. Jego zadaniem jest dostarczenie centralnego komponentu odpowiedzialnego za zarządzanie zaplanowanymi zadaniami utrzymania. W poniższym punkcie przyjrzemy się architekturze AutoTask, widokom danych słownikowych oraz zarządzaniu AutoTask ręcznie i poprzez OEM. Na koniec przyjrzemy się oknom utrzymania AutoTask.

Architektura AutoTask

AutoTask wyrasta z istniejącej architektury Oracle Database 11g. Częściami składowymi architektury AutoTask są:

- ◆ **AutoTask Background Process (ABP).** Ten proces tła jest tworzony przez MMON. Jest on odpowiedzialny za zarządzanie cechami AutoTask, koordynuje klientów (*client*) oraz planistę (*scheduler*) AutoTask, a także utrzymuje historię związaną z AutoTask, którą można obejrzeć w widoku DBA_AUTOTASK_TASK.
- ◆ **AutoTask Clients.** Są to automatyczne zadania utrzymania planowane przez AutoTask. Oracle Database 11g zawiera klienty do zbierania statystyk, klienty Segment Advisor oraz klienty Automatic SQL Tuning Advisor.
- ◆ **AutoTask Maintenance Windows.** Różne okna utrzymania istnieją dla różnych dni tygodnia. Mogą one być modyfikowane zgodnie z potrzebami.
- ◆ **Resource Manager.** Przygotowany plan zasobów ma za zadanie ograniczać ilość zasobów konsumowanych przez AutoTask. Plan ten może być modyfikowany według potrzeb.
- ◆ **OEM.** Pozwala zarządzać czasem rozpoczęcia i trwania różnych okien utrzymania AutoTask oraz dodawać i usuwać zadania utrzymania. OEM można także wykorzystać do uaktywniania i dezaktywowania zadań utrzymania.
- ◆ **Scheduler.** Planista. Operacje AutoTask polegają w znacznej mierze na planiście. Architektura AutoTask wykorzystuje okna i infrastrukturę planisty do uruchamiania zadań AutoTask.

Widoki danych słownikowych dla AutoTask

Dla wsparcia infrastruktury AutoTask zostały stworzone nowe widoki. Stare prace (*job*) w Oracle Database 11g zostały usunięte z widoków `DBA_SCHEDULER_*` i przesunięte do widoków `DBA_AUTOTASK_*` (jak np. SQL Tuning Advisor). Dodano następujące widoki związane z AutoTask:

- ♦ **DBA_AUTOTASK_CLIENT** — widok ten zawiera listę różnych klientów uruchamianych przez AutoTask. Widok `DBA_AUTOTASK_CLIENT` posiada zasób informacji, włączając w to nazwy różnych zadań AutoTask, które mają zostać uruchomione, statusy tych zadań, grupy konsumentek, do których one należą, i średni czas wykonania. Kolejne treści dostępne w widoku to: zużycie procesora przez zadanie w czasie, maksymalny czas trwania klienta, nazwa grupy okien, do której przypisany jest klient, oraz wiele innych atrybutów klienta. Oto przykład wykorzystania widoku `DBA_AUTOTASK_CLIENT`:

```
SQL> select client_name, status, consumer_group, mean_job_duration
from dba_autotask_client;
```

CLIENT_NAME MEAN_JOB_DURATION	STATUS	CONSUMER_GROUP
auto optimizer stats collection +000000000 00:04:26.000000000	ENABLED	ORA\$AUTOTASK_STATS_GROUP
auto space advisor +000000000 00:00:45.272727272	ENABLED	ORA\$AUTOTASK_SPACE_GROUP
sql tuning advisor +000000000 00:05:53.000000000	ENABLED	ORA\$AUTOTASK_SQL_GROUP



W dokumentacji znajdziemy informację, że kolumna `MEAN_JOB_DURATION` zawiera średnią, a nie środkową wartość czasu trwania zadania, jak mogłaby sugerować jej nazwa.

- ♦ **DBA_AUTOTASK_CLIENT_HISTORY** — widok, który dostarcza historycznej reprezentacji uruchomień zdarzeń klientów AutoTask w ramach okna. Pozwala on spojrzeć na historyczną informację przebiegu każdego klienta AutoTask w oparciu o dane okno, w którym AutoTask działał. Poniżej przedstawiamy przykładowe zapytanie do widoku, sprawdzające, kiedy działał klient zautomatyzowanego zbierania statystyk:

```
SQL> select client_name, window_name, jobs_created, jobs_started,
jobs_completed
from dba_autotask_client_history
where client_name like '%stats%';
```

CLIENT_NAME	WINDOW_NAME	JOBS CREATED	JOBS STARTED	JOBS COMPLETED
auto optimizer stats collection	THURSDAY_WINDOW	1	1	1
auto optimizer stats collection	SUNDAY_WINDOW	3	3	3
auto optimizer stats collection	MONDAY_WINDOW	1	1	1
auto optimizer stats collection	SATURDAY_WINDOW	2	2	2
auto optimizer stats collection	SUNDAY_WINDOW	4	4	4
auto optimizer stats collection	TUESDAY_WINDOW	1	1	1

- ◆ **DBA_AUTOTASK_CLIENT_JOB** — dostarcza informacji o aktualnie zdefiniowanych i wykonywanych pracach AutoTask. Ten widok zazwyczaj jest pusty, chyba że jakaś praca jest w danym momencie w toku.
- ◆ **DBA_AUTOTASK_JOB_HISTORY** — widok dostarczający informacji na temat historii wykonania każdego klienta AutoTask.

```
SQL> select client_name, job_status, job_start_time, job_duration
from dba_autotask_job_history
where client_name like '%stats%' order by job_start_time;
CLIENT_NAME                JOB_STATUS JOB_START_TIME
JOB_DURATION
-----
auto optimizer stats collection SUCCEEDED 18-FEB-07 02.43.45.598298 PM -07:00
+000 00:00:24
auto optimizer stats collection SUCCEEDED 18-FEB-07 06.49.51.326230 PM -07:00
+000 00:02:21
auto optimizer stats collection SUCCEEDED 18-FEB-07 10.59.53.677261 PM -07:00
+000 00:00:55
auto optimizer stats collection SUCCEEDED 23-FEB-07 12.57.25.844519 AM -07:00
+000 00:12:49
auto optimizer stats collection SUCCEEDED 24-FEB-07 06.17.02.045879 PM -07:00
+000 00:13:09
auto optimizer stats collection SUCCEEDED 05-MAR-07 10.00.06.955011 PM -07:00
+000 00:04:32
```

- ◆ **DBA_AUTOTASK_OPERATION** — widok dostarczający informacji na temat atrybutów przypisanych operacjom AutoTask. Przykładowo, praca klienta AutoTask może być oznaczona jako `SAFE TO KILL`, jeśli może zostać przerwana w środku działania bez negatywnych skutków. Przykład takiej pracy pokazany został w wydruku poniższego zapytania, gdzie kolumna `ATTRIBUTES` pracy „automatyczne zbieranie statystyk” jest oznaczona jako `SAFE TO KILL`:

```
SQL> select * from dba_autotask_operation
where client_name like '%stats%';

CLIENT_NAME                OPERATION_NAME                OPE PRIORIT
-----
ATTRIBUTES                USE_R STATUS
-----
auto optimizer stats collection auto optimizer stats job OPT INVALID
ON BY DEFAULT, VOLATILE, SAFE TO KILL FALSE ENABLED
```

- ◆ **DBA_AUTOTASK_SCHEDULE** — dostarcza czasy rozpoczęcia i trwania każdego z okien AutoTask.
- ◆ **DBA_AUTOTASK_TASK** — ten widok dostarcza danych na temat każdego indywidualnego zadania AutoTask. Oferuje zasób informacji od statystyk ostatniego wykonania danego zadania, do oszacowanych statystyk jego kolejnego wykonania.

```
SQL> select client_name, task_name, task_target_type
from dba_autotask_task;
CLIENT_NAME                TASK_NAME                TASK_TARGET_TYPE
-----
auto optimizer stats collection gather_stats_prog        System
auto space advisor         auto_space_advisor_prog  System
sql tuning advisor         AUTO_SQL_TUNING_PROG     SQL Workload
```



Uwaga

Tutaj widzimy przykładowe miejsce, gdzie AutoTask spotyka się z planistą. Zwróćmy uwagę na kolumnę `TASK_NAME` w widoku `DBA_AUTOTASK_TASK`. Wiąże ona zadanie AutoTask z programem składowanym planisty, który to program można odszukać w widoku `DBA_SCHEDULER_PROGRAMS`, tak jak pokazano poniżej:

```
select program_name
from dba_scheduler_programs
where program_name='GATHER_STATS_PROG';
PROGRAM_NAME
-----
GATHER_STATS_PROG
```

- ♦ **DBA_AUTOTASK_WINDOW_CLIENTS** — widok dostarczający informacji na temat różnych okien powiązanych z klientami AutoTask.
- ♦ **DBA_AUTOTASK_WINDOW_HISTORY** — widok ten dostarcza informacji na temat okien AutoTask.

Zarządzanie zadaniami AutoTask poprzez OEM

OEM oferuje interfejs do zarządzania zadaniami AutoTask na stronie *Scheduler Central*. Link do niej znajduje się na dole strony domowej OEM. Na stronie domowej planisty możemy zobaczyć zarówno zautomatyzowane zadania utrzymania uruchamiane przez AutoTask, jak i zwykłe prace (*job*) planisty lub prace Enterprise Managera. Rysunek 2.4 przedstawia przykład strony OEM *Scheduler Central*, wyświetlającej zautomatyzowane zadania utrzymania, których wykonanie zostało zaplanowane.

The screenshot shows the Oracle Enterprise Manager Scheduler Central interface. The browser address bar indicates the URL: `https://acer1:1158/em/console/database/instance/autoTask?pageNo`. The page title is "ORACLE Enterprise Manager 11g Database Control". The database instance is "ordl" and the user is logged in as "As SYS". The page was refreshed on 23-Sep-2008 at 21:40:07 o'clock CEST.

The "Schedulers" section includes:

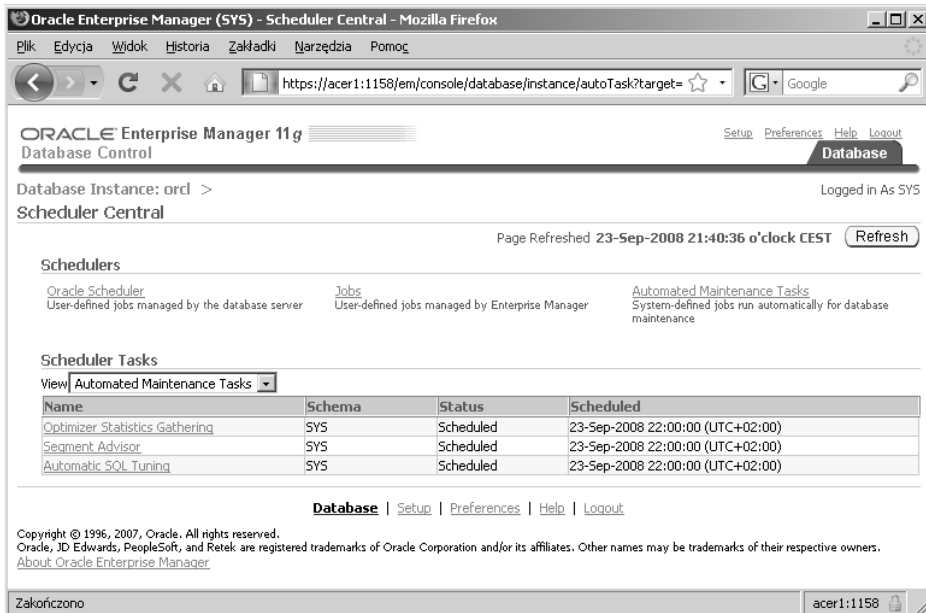
- Oracle Scheduler: User-defined jobs managed by the database server
- Jobs: User-defined jobs managed by Enterprise Manager
- Automated Maintenance Tasks: System-defined jobs run automatically for database maintenance

The "Scheduler Tasks" section shows a table with the following data:

Name	Schema	Type	Status	Scheduled
Optimizer Statistics Gathering	SYS	Automated Maintenance Tasks	Scheduled	23-Sep-2008 22:00:00 (UTC+02:00)
Segment Advisor	SYS	Automated Maintenance Tasks	Scheduled	23-Sep-2008 22:00:00 (UTC+02:00)
Automatic SQL Tuning	SYS	Automated Maintenance Tasks	Scheduled	23-Sep-2008 22:00:00 (UTC+02:00)
BSLN_MAINTAIN_STATS_JOB	SYS	Oracle Scheduler	Scheduled	28-Sep-2008 00:00:00 (UTC-05:00)
DRA_REEVALUATE_OPEN_FAILURES	SYS	Oracle Scheduler	Scheduled	MAINTENANCE_WINDOW_GROUP
HM_CREATE_OFFLINE_DICTIONARY	SYS	Oracle Scheduler	Scheduled	MAINTENANCE_WINDOW_GROUP
HM_CREATE_OFFLINE_DICTIONARY	SYS	Oracle Scheduler	Scheduled	MAINTENANCE_WINDOW_GROUP

Rysunek 2.4. Scheduler Central — strona narzędzia OEM

Na górze strony OEM znajdują się różne rodzaje dostępnych prac wymienione na liście rozwijanej. Każda z pozycji listy jest hiperłączem przenoszącym do strony pozwalającej na dalsze zarządzanie tymi rodzajami prac. Przykładowo, możemy wybrać *Automated Maintenance Tasks*, a OEM wyświetli stronę pokazaną na rysunku 2.5, która pozwala na zarządzanie zautomatyzowanymi zadaniami utrzymania.

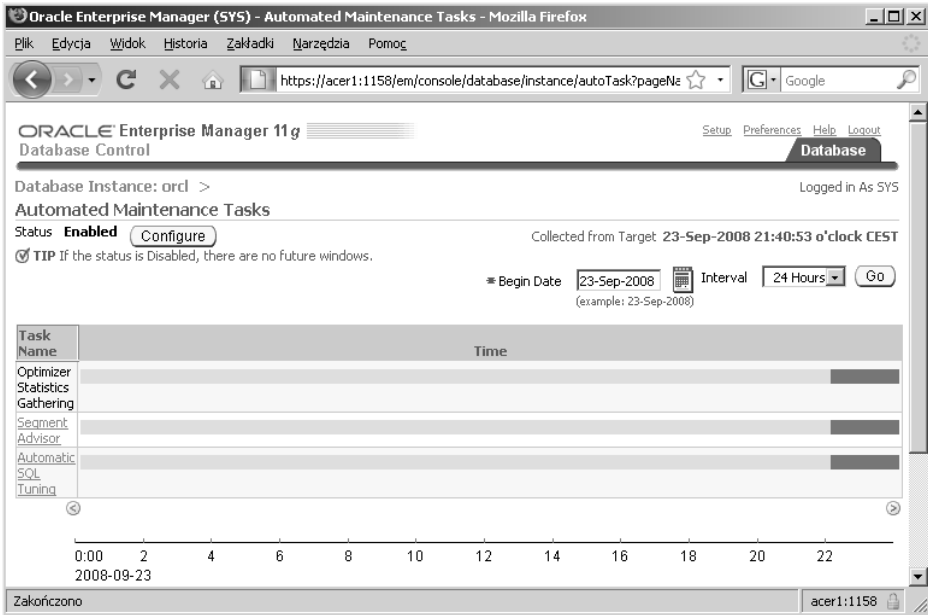


Rysunek 2.5. Strona OEM Scheduler Central wyświetlająca zadania *Automated Maintenance Tasks*

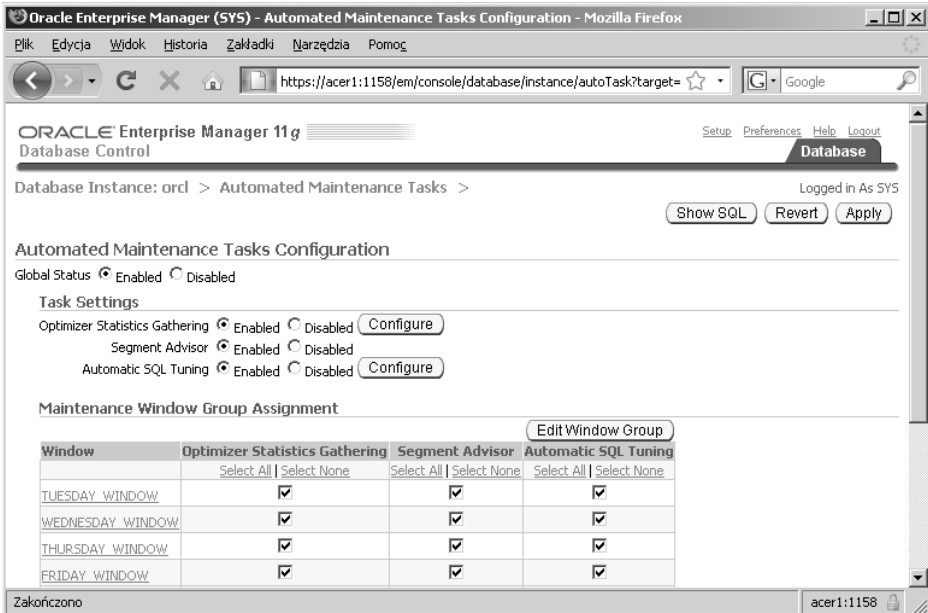
Na stronie z rysunku 2.5 można sprawdzić, czy zadania wykonują się w obrębie ich zaplanowanych okien, czy przekraczają te ramy czasowe. Zwróćmy uwagę na trzy opcje znajdujące się na górze strony pod nagłówkiem *Schedulers*. Pozwalają one na zarządzanie różnymi rodzajami zaplanowanych zadań. Przykładowo, jeśli chcemy zarządzać zautomatyzowanymi zadaniami utrzymania, powinniśmy wybrać link *Automated Maintenance Tasks*, a przejdziemy do strony z rysunku 2.6.

Ze strony z rysunku 2.6 możemy zarządzać zautomatyzowanymi zadaniami utrzymania, klikając przycisk *Configure* umieszczony na górze. Niżej na stronie znajdziemy wymienione konkretne zautomatyzowane zadania utrzymania oraz graficznie przedstawiony plan ich przeszłych i przyszłych uruchomień. Możemy kliknąć na dane zadanie, aby zobaczyć szczegółowe zalecenia dostarczone jako część frameworku doradcy. Jeśli klikniemy w *Segment Advisor*, zostaniemy przeniesieni na stronę OEM *Segment Advisor Recommendations*.

Wspomnieliśmy o dostępnej możliwości konfiguracji zautomatyzowanych zadań utrzymania poprzez kliknięcie w przycisk *Configure* widoczny na rysunku 2.6. OEM zapewnia opcje uaktywnienia lub dezaktywacji zautomatyzowanego zadania utrzymania albo na poziomie globalnym, albo na poziomach ziarnistości. Można uaktywnić lub dezaktywować określone zadania lub ich konkretne zaplanowane uruchomienia. Niektóre zadania posiadają przycisk *Configure*, który pozwala na ich dalszą konfigurację. Rysunek 2.7 przedstawia przykładową stronę OEM *Automated Maintenance Tasks Configuration*.



Rysunek 2.6. Strona OEM Automated Maintenance Tasks



Rysunek 2.7. Strona OEM Automated Maintenance Tasks Configuration

Ręczne zarządzanie zadaniami AutoTask

Oracle Database 10g do administracji nowymi zautomatyzowanymi pracami planisty, takimi jak standardowo dostępne („z pudełka”) zbieranie statystyk bazodanowych, używało pakietu `dbms_scheduler`. Wraz z pojawieniem się AutoTask będziemy używali nowego pakietu, aby zarządzać tymi pracami, a mianowicie `dbms_auto_task_admin`. OEM także wykorzystuje ten pakiet do zarządzania zautomatyzowanymi pracami.

Podprogramu `dbms_auto_task_admin.disable` będziemy używali do dezaktywowania dowolnego z zadań AutoTask. Zwróćmy uwagę, że żaden z parametrów nie posiada domyślnych wartości, dlatego w składni wywołania należy podać je wszystkie. W poniższym przykładzie wyłączymy automatyczne zbieranie statystyk:

```
BEGIN
  dbms_auto_task_admin.disable(client_name => 'auto optimizer stats collection',
    operation => NULL, window_name => NULL);
END;
/
```

Parametr `window_name` pozwala zdefiniować konkretne okno, które zamierzamy dezaktywować (w przeciwieństwie do dezaktywacji całego zadania). Standardowo mamy siedem okien, każde na jeden dzień tygodnia. Noszą one nazwy `MONDAY_WINDOW`, `TUESDAY_WINDOW` i tak dalej. Jeśli nie życzymy sobie, aby statystyki były gromadzone w niedzielę (bo na przykład w tym dniu ładujemy nowe rekordy do hurtowni danych), jesteśmy w stanie uniemożliwić wykonanie AutoTask w tym dniu następującym poleceniem:

```
BEGIN
  dbms_auto_task_admin.disable(client_name => 'auto optimizer stats collection',
    operation => NULL, window_name => 'SUNDAY_WINDOW');
END;
/
```

Natomiast aby ponownie umożliwić wykonanie danego zadania, skorzystajmy z podprogramu `dbms_auto_task_admin.enable`. Znowu musimy podać wartości dla wszystkich parametrów, jak pokazuje poniższy przykład:

```
BEGIN
  dbms_auto_task_admin.enable(client_name => 'auto optimizer stats collection',
    operation => NULL, window_name => NULL);
END;
/
```

Okna utrzymania AutoTask

Jak wspomniano wcześniej, zadania AutoTask są zaprojektowane tak, aby wykonywały się podczas trwania okien utrzymania AutoTask. Istnieje siedem domyślnych okien, po jednym na każdy dzień tygodnia. Okna dni roboczych (`MONDAY_WINDOW`, `TUESDAY_WINDOW` i tak dalej) trwają cztery godziny i rozpoczynają się o godzinie 22 lokalnego czasu. Okna dni weekendowych trwają 20 godzin, a ich czas rozpoczęcia zaplanowany jest na godzinę 6 rano lokalnego czasu.



Uwaga

Okna *_WINDOWS zastępują okna WEEKEND_WINDOW i WEEKNIGHT_WINDOW dostępne w Oracle Database 10g, jednakże oba te okna pozostają nadal zdefiniowane w Oracle Database 11g.

Okna są przypisane do planu zasobów, DEFAULT_MAINTENANCE_PLAN, który jest automatycznie uaktywniany podczas otwarcia okien utrzymania. Do planu zasobów DEFAULT_MAINTENANCE_PLAN przypisanych jest wiele grup konsumenckich oraz różne towarzyszące zadania, włączając w to:

- ♦ **ORA\$AUTOTASK_SQL_GROUP** — do tej grupy konsumenckiej są przypisane zadania automatycznego strojenia SQL, czyli zadania Automatic SQL Tuning.
- ♦ **ORA\$AUTOTASK_SPACE_GROUP** — do tej grupy przypisane są zadania Segment Advisor.
- ♦ **ORA\$AUTOTASK_STATS_GROUP** — do tej grupy przypisane są zadania automatycznego zbierania statystyk.

Każda z wymienionych grup kontroluje wykorzystanie zasobów (np. dopuszczalne obciążenie procesora) przez konkretne zadania utrzymania AutoTask.

Zmiany i nowe własności w zarządzaniu plikiem parametrów

Firma Oracle wprowadziła kilka zmian w zarządzaniu plikami parametrów serwera (pliki *SPFILE*). Włączamy w to:

- ♦ obsługę błędów odczytu/zapisu do plików *SPFILE*;
- ♦ łatwiejsze przejście na używanie plików *SPFILE*;
- ♦ zabezpieczenie przed ustawianiem przez użytkowników nieprawidłowych wartości w plikach *SPFILE*.

Przyjrzyjmy się bliżej poniższym własnościom.

Obsługa błędów odczytu/zapisu do SPFILE

W Oracle Database 11g pliki *SPFILE* są bardziej chronione. Jeśli wystąpi błąd odczytu/zapisu do pliku *SPFILE*, Oracle wygeneruje błąd w dzienniku alertów, a przyszłe wpisy do pliku parametrów zostaną zignorowane. W takiej sytuacji możemy optować za zamknięciem bazy danych i przywróceniem pliku parametrów. Możemy także utworzyć nowy plik parametrów, korzystając z nowej opcji `create spfile from memory` (zostanie ona omówiona w kolejnym punkcie). W końcu, w przypadku wystąpienia błędu związanego z plikiem *SPFILE*, możemy kontynuować działanie bazy danych.

Prostsze przejście na wykorzystywanie plików SPFILE

Niestety, zapewne są pośród nas tacy, którzy ciągle jeszcze nie przeszli na uruchamianie z wykorzystaniem pliku *SPFILE*. Tworzenie i rozpoczęcie używania tego pliku jest teraz jeszcze prostsze — możemy teraz używać nowego polecenia `create spfile from memory`. Polecenie to zbierze ustawienia z pamięci i dokona ich zrzutu do pliku *SPFILE*. Tę metodę, jeśli sobie tego życzymy, można także wykorzystać do utworzenia kopii zapasowej pliku *SPFILE* przy użyciu bieżących ustawień z pamięci. Jest to bardzo poręczny sposób wykonania kopii zapasowej pliku *SPFILE*, zanim dokonamy w nim jakichkolwiek zmian za pomocą polecenia `alter system`.

Standardowo Oracle stworzy plik *SPFILE* w lokalizacji domyślnej, zdefiniowanej w systemie operacyjnym (często jest to *ORACLE_HOME/dbs*). Możemy jednak wyspecyfikować miejsce, gdzie plik *SPFILE* ma zostać utworzony, definiując nazwę pliku i jego lokalizację, jak pokazuje następujący przykład:

```
Create spfile='/oracle/util/spfilename.sp' from memory;
```

Zabezpieczenie przed ustawianiem przez użytkowników nieprawidłowych wartości w plikach SPFILE

W Oracle Database 11g dołożono dodatkową walidację podczas dokonywania zmian w parametrach bazy danych. Obecnie w wielu przypadkach, podczas próby zmiany parametru z użyciem niepoprawnej składni, Oracle pouczy o zastosowaniu nieprawidłowej wartości. Niestety, nie jest to w Oracle Database 11g do końca spójne, ale dobre na początek. Oto przykład:

```
SQL> alter system set control_management_pack_access = 'Wrong_Value'  
scope=spfile;  
alter system set control_management_pack_access = 'Wrong_Value' scope=spfile  
*  
ERROR at line 1:  
ORA-00096: invalid value Wrong_Value for parameter  
control_management_pack_access. must be from among DIAGNOSTIC+TUNING,  
DIAGNOSTIC, NONE
```

Zmiany i nowe własności w Resource Managerze

Oracle Database 11g przedstawił szereg nowych własności powiązanych z menedżerem zasobów Resource Manager. W ich skład wliczamy:

- ◆ zdolność zmierzenia maksymalnej przepustowości wejścia/wyjścia systemu (kalibracja wejścia/wyjścia);
- ◆ domyślny plan utrzymania;
- ◆ wbudowane plany zasobów;

- ♦ historii statystyk Resource Managera składowane w AWR;
- ♦ nowe dyrektywy planu Resource Managera.

Kalibracja wejścia/wyjścia

Oracle Database Resource Manager posiada nową procedurę pozwalającą uruchomić testy kalibracji wejścia/wyjścia w naszej bazie danych oraz przejrzeć ich wyniki. Uruchomienie procedury `dbms_resource_manager.calibrate_io` wygeneruje obciążenie na wszystkich węzłach klastra. Procedura przyjmuje dwa parametry wejściowe i trzy parametry wyjściowe, jak pokazano w poniższych tabelach:

Parametr wejścia	Typ	Znaczenie
<code>num_physical_disks</code>	<code>pls_integer</code>	Jest to przybliżona liczba dysków fizycznych używanych przez bazę danych. Domyślna wartość wynosi 1.
<code>max_latency</code>	<code>pls_integer</code>	Maksymalne opóźnienie w milisekundach dozwolone dla żądań operacji wejścia/wyjścia o rozmiarze równym wartości <code>db_block_size</code> . Domyślnie 20.

Parametr wyjścia	Typ	Znaczenie
<code>max_iops</code>	<code>pls_integer</code>	Maksymalna liczba losowo rozdystrybuowanych żądań operacji wejścia/wyjścia o rozmiarze <code>db_block_size</code> , które system może udźwignąć.
<code>max_mbps</code>	<code>pls_integer</code>	Maksymalna liczba megabajtów na sekundę, jaką system może udźwignąć. Obliczenie oparte na losowo rozdystrybuowanych 1-megabajtowych odczytach.
<code>actual_latency</code>	<code>pls_integer</code>	Średnie opóźnienie żądań wejścia/wyjścia <code>db_block_size</code> w tempie <code>MAX_IOPS</code> na ms.



Uwaga

W celu użycia opisanej funkcji należy posługiwać się asynchronicznymi operacjami wejścia/wyjścia.

Oto przykład użycia procedury `dbms_resource_manager.calibrate_io`:

```

Declare
v_max_iops PLS_INTEGER:=1;
v_max_mbps PLS_INTEGER:=1;
v_actual_latency PLS_INTEGER:=1;
begin
dbms_resource_manager.calibrate_io(
max_iops=>v_max_iops,
max_mbps=>v_max_mbps,
actual_latency=>v_actual_latency);
dbms_output.put_line('Results follow: ');
dbms_output.put_line('Max IOPS: '||v_max_iops);
dbms_output.put_line('Max MBPS: '||v_max_mbps);
dbms_output.put_line('Actual Latency: '||v_actual_latency);
end;
/

```

Status kalibracji (bieżącej lub historycznej) możemy sprawdzić, odpytując widok V\$IO_CALIBRATION_STATUS, jak pokazano w następującym przykładzie:

```
select status from v$io_calibration_status;
STATUS
-----
IN PROGRESS
```

Wyniki kalibracji znajdziemy w tabeli DBA_RSRC_IO_CALIBRATE, jak pokazuje poniższy przykład:

```
select * from dba_rsrc_io_calibrate;
STATUS          LATENCY    MAX_IOPS  MAX_MBPS  NUM_DISKS
-----
CALIBRATION_TIME
-----
READY           19         77         6          1
27-MAY-07 09.50.15.421 PM
```

Domyślny plan utrzymania

Tak jak w poprzednich wersjach Oracle, zazwyczaj nie istnieje domyślnie aktywny plan utrzymania. Jest jeden wyjątek, o którym należy pamiętać, a jest on związany z planem zasobów DEFAULT_MAINTENANCE_PLAN, o którym mówiliśmy wcześniej w tym rozdziale. Kiedy okno utrzymania planisty zostaje otwarte, a plan jest z nim związany (jak podczas trwania okna utrzymania), wtedy plan powiązany z oknem planisty zostaje uaktywniony w systemie.

Jeśli nie chcemy, aby planista definiował plan zasobów, możemy wykorzystać procedurę dbms_scheduler.set_attribute w celu ustawienia atrybutu resource_plan na pusty ciąg znaków, co wyłączy tę własność. Będzie trzeba zmodyfikować wszystkie okna utrzymania w Oracle Database 11g, aby całkowicie zapobiec ustawieniu planu DEFAULT_MAINTENANCE_PLAN. Podobnie, jeśli chcemy przypisać inny plan zasobów, możemy posłużyć się tą samą metodą (czyli procedurą dbms_scheduler.set_attribute), aby dla danego harmonogramu uruchomień (schedule) ustawić atrybut nazwy planu zasobów na nową wartość.

Arup radzi...

Kalibracja wejścia/wyjścia nie jest niczym nowym w sektorze IT. Istnieje kilkanaście innych technologii, które ją umożliwiają — począwszy od rozwiązań proponowanych przez dostawców sprzętu, a na dostawcach oprogramowania systemów operacyjnych skończywszy. Nawet Oracle dostarcza narzędzie zwane Orion, częściowo realizujące ten cel. Zatem jakich dodatkowych funkcji dostarcza narzędzie kalibracji wejścia/wyjścia w 11g?

Otóż jest ich mnóstwo. Po pierwsze, wszystkie pozostałe narzędzia jedynie wykonują wiele operacji wejścia/wyjścia do dysków i budują profil oparty o ten transfer. Niektóre idą dalej, dostarczając więcej ziarnistości, inteligencji i tym podobnych, jednak w mniejszym lub większym stopniu ich działanie pozostaje takie samo. Procedura kalibracji wejścia/wyjścia pakietu dbms_resource_manager jest wyjątkowa w sensie wykonywania działań *identycznych* z rutynowymi działaniami Oracle Database, a nie generycznych żądań wejścia/wyjścia. Dzięki temu kalibracja dobrze odzwierciedla faktyczne wywołania bazy danych Oracle, co ma znaczący wpływ na wydajność. Gdyby istniał błąd w kodzie jądra Oracle, w obszarze wejścia/wyjścia, to zapewne miałby on wpływ także na kalibrację, co byłoby wielce pożądane. Oczywiście mówimy tu o pożądaniu wykrycia efektów istniejącego błędu, a nie o pożądaniu jego istnienia samego w sobie.

Jeśli generalnie jesteśmy za tym, aby plan `DEFAULT_MAINTENANCE_PLAN` był wykorzystywany przez planistę, ale są sytuacje, w których nie chcemy przesłaniać bieżącego planu zasobów, możemy skonfigurować parametr bazy danych `resource_manager_plan` z atrybutem `force` zapobiegającym przesłonięciu planu, jak pokazano w przykładzie:

```
Alter system set resource_manager_plan='force:rf_plan' scope=both;
```

Aby wybrać plan, możemy posłużyć się procedurą PL/SQL `dbms_resource_manager.switch_plan`. W celu pozbycia się atrybutu `force` wykonujemy polecenie `alter system`, podobnie jak pokazano w powyższym przykładzie, tyle że po prostu bez atrybutu `force`. Oto przykład:

```
Alter system set resource_manager_plan='rf_plan' scope=both;
```

Wbudowane plany zasobów

Oracle Database 10g dostarcza wbudowanego planu zasobów zwanego `SYSTEM_PLAN`. Sam w sobie nie był on zbyt przydatnym planem zasobów. Oracle Database 11g dostarcza nowego planu `MIXED-WORKLOAD PLAN`, zapewniającego zarządzanie zasobami w mieszanym środowisku, w którym występuje zarówno przetwarzanie OLTP, jak i DSS (lub prace wsadowe).

Statystyki Resource Managera w AWR

Jedną z ciekawszych własności AWR jest dostarczanie trwałego repozytorium historycznych statystyk. Ponieważ widoki `V$` są czyszczone, kiedy zamykamy bazę danych, AWR staje się najlepszym miejscem rzetelnej reprezentacji historycznej wydajności naszej bazy danych. Aby ułatwić śledzenie efektywności Oracle Database Resource Managera, AWR zawiera teraz widoki dostarczające historycznej informacji na temat jego wydajności. Nowe widoki to:

- ♦ **DBA_HIST_RSRC_PLAN** — zawiera historyczne dane z `V$RESOURCE_PLAN`.
- ♦ **DBA_HIST_RSRC_CONSUMER_GROUP** — zawiera historyczne dane z `V$RESOURCE_CONSUMER_GROUP`.

Każdy z tych widoków jest powiązany z konkretną migawką AWR i może zostać zmaterializowany przez OEM. Ponadto w Oracle Database 11g dodano widok `V$RSRCMGRMETRIC`, zawierający historyczne metryki Resource Managera konsumpcji zasobów przez różne grupy konsumentek.

Nowe własności dyrektyw planu Resource Managera

W Oracle Database 11g z dyrektywami planu Resource Managera zostały skojarzone nowe atrybuty, w skład których wchodzi:

- ♦ **switch_io_megabytes** — definiuje ilość megabajtów wejścia/wyjścia, które mogą zostać skonsumowane przez daną sesję, zanim zostanie podjęta akcja dyrektywy. Domyślna wartość to `NULL` (nieograniczona).
- ♦ **switch_io_reqs** — definiuje liczbę żądań wejścia/wyjścia, które mogą zostać wywołane przez daną sesję, zanim zostanie podjęta akcja dyrektywy. Domyślna wartość to `NULL` (nieograniczona).

- ◆ `switch_for_call` — zastępuje parametr `switch_time_in_call`, który w Oracle Database 11g jest niezalecany. Jeśli wartość parametru `switch_for_call` wynosi `TRUE`, a akcja została podjęta jako rezultat jednego z parametrów: `switch_time`, `switch_io_megabytes` lub `switch_io_reqs`, to na końcu głównego wywołania grupa konsumentcka zostanie przywrócona do swojej oryginalnej grupy konsumentckiej.
- ◆ `mgmt_p1` – `mgmt_p8` — parametry te zastępują `CPU_P1` – `CPU_P8`, które w Oracle Database 11g są niezalecane. Jeśli plan posiada parametr `CPU_MTH` ustawiony na wartość `EMPHASIS`, to definiuje procent zużycia procesora konieczny do zaalokowania różnych poziomów (od 1 do 8). Tylko `MGMT_P1` jest odpowiedni, jeśli parametr `CPU_MTH` jest ustawiony na wartość `RATIO` (zazwyczaj używana w przypadku prostszych planów, z tylko jednym poziomem alokacji procesora), w którym to przypadku definiuje wagę użycia procesora.

Przykład wykorzystania nowych parametrów przedstawiamy poniżej:

```
-- Stwórz plan zasobów
BEGIN
    DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
END;
/
BEGIN
    DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP (
        CONSUMER_GROUP => 'OLTP',
        COMMENT => 'OLTP');
END;
/
BEGIN
    DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP (
        CONSUMER_GROUP => 'HIGH_IO_GROUP',
        COMMENT => 'OLTP');
END;
/
BEGIN
    DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP (
        CONSUMER_GROUP => 'EXCESSIVE_IO_GROUP',
        COMMENT => 'OLTP');
END;
/
BEGIN
    DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP (
        CONSUMER_GROUP => 'REPORTING',
        COMMENT => 'RAPORTOWANIE');
END;
/
BEGIN
    DBMS_RESOURCE_MANAGER.CREATE_PLAN(
        PLAN => 'DAYTIME',
        COMMENT => 'Więcej zasobów dla aplikacji OLTP');
END;
/
BEGIN
    DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (
        PLAN => 'DAYTIME',
        GROUP_OR_SUBPLAN => 'OLTP',
        COMMENT => 'grupa OLTP',
        MGMT_P1 => 75,
        SWITCH_GROUP => 'HIGH_IO_GROUP',
```

```
SWITCH_IO_REQS => 100,
SWITCH_IO_MEGABYTES => 250,
SWITCH_FOR_CALL => TRUE);
END;
/
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (
    PLAN => 'DAYTIME',
    GROUP_OR_SUBPLAN => 'REPORTING',
    COMMENT => 'grupa raportowania',
    MGMT_P1 => 15,
    PARALLEL_DEGREE_LIMIT_P1 => 8,
    ACTIVE_SESS_POOL_P1 => 4);
  DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (
    PLAN => 'DAYTIME',
    GROUP_OR_SUBPLAN => 'OTHER_GROUPS',
    COMMENT => 'Ta jest wymagana',
    MGMT_P1 => 10);
END;
/
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE (
    PLAN => 'DAYTIME',
    GROUP_OR_SUBPLAN => 'EXCESSIVE_IO_GROUP',
    COMMENT => 'grupa wysokiego wejścia/wyjścia',
    MGMT_P1 => 15,
    PARALLEL_DEGREE_LIMIT_P1 => 2);
END;
/
```

Zależności drobnoziarniste

Przed Oracle Database 11g zmiany obiektów bazy danych mogły powodować mylne uznanie podległych obiektów za niepoprawne (*invalid*). Przykładowo, przed 11g, dodanie kolumny do tabeli mogło skutkować tym, że zależny widok został uznany za niepoprawny.

Obecnie Oracle Database 11g zapisuje metadane zależności na większym poziomie ziarnistości, więc czynności takie jak dodanie lub usunięcie z tabeli kolumny niemającej wpływu na zależny obiekt nie spowodują uznania tego podległego obiektu za niepoprawny.

Obecnie domyślna opcja DDL WAIT

W Oracle Database 10g, domyślnie, polecenia DDL nie oczekiwały na możliwość założenia blokady na obiekcie. Zamiast tego, w razie istnienia już konfliktowej blokady na obiekcie, próba wykonania polecenia kończyła się niepowodzeniem oraz generowany był błąd ORA-00054. Obecnie, w wersji 11g, w wielu przypadkach Oracle nie tylko nie zwróci błędu, ale wykona modyfikację nawet bez potrzeby oczekiwania. Przykładowo, w Oracle Database 11g można wykonać, co następuje:

```
Sesja 1:
SQL>insert into test values (1);
1 row created.
```

```
Sesja 2:
SQL>alter table test add (id2 number);
Table altered.
```

Zauważmy, że w powyższym przykładzie sesja nr 2 nie musi czekać na zatwierdzenie sesji nr 1 (np. poleceniem `commit`). Właściwie, gdybyśmy wykonali polecenie `describe` tabeli `TEST` w sesji nr 1 nawet przed wykonaniem zatwierdzenia, nie tylko widzielibyśmy nową kolumnę, ale również moglibyśmy wstawić do tabeli nowy wiersz z przypisaną wartością dla tej kolumny³.

Arup radzi...

Opcja DDL `WAIT` to bardzo przydatna własność. Ileż to razy próbowaliśmy zmodyfikować tabelę w środowisku produkcyjnym i okazywało się, że ktoś akurat jej używa? Nasze polecenie `alter table` kończyło się błędem⁴ `ORA-00054: resource busy and acquire with nowait...` Wszystko, czego było nam trzeba, to minimalne okno czasowe, w którym moglibyśmy otrzymać wyłączną blokadę na obiekcie i wykonać konieczne modyfikacje. Operacje DML mogłyby zostać wznowione chwilę później. Niestety! Nie dało się tego wykonać w Oracle Database 10g ani wcześniej.

Musiałem uciekać się do sztuczek polegających na umieszczaniu polecenia DDL w pętli i wykonywaniu go kilkanaście razy z nadzieją, że w końcu uda się założyć blokadę na obiekcie — bardzo brutalne, ale skuteczne. W Oracle Database 11g nie trzeba się wiele napracować. W sesji, która będzie wydawać polecenie DDL, należy najpierw wykonać następujące polecenie:

```
alter session set ddl_lock_timeout = 10;
```

Dzięki temu każde kolejne polecenie DDL tej sesji poczeka 10 sekund, zanim zostanie przerwane (chyba że wykona się pomyślnie przed upływem tego czasu). Jeśli wydamy polecenie DDL, które nie będzie w stanie założyć wyłącznej blokady z powodu czynności DML trwających w innych sesjach, nie zakończy się ono błędem `ORA-00054`, ale raczej zostanie zawieszona (właściwie to poczeka na możliwość założenia blokady). W momencie założenia blokady polecenie zakończy się sukcesem. Dopiero jeśli nie uda się tego zrobić w przeciągu 10 sekund, polecenie to zakończy się błędem `ORA-00054`.

Powyzsza własność jest niezwykle przydatna w wykonywaniu modyfikacji DDL w systemach o wysokiej zajętości.

Możemy także wykorzystać nowy parametr `ddl_lock_timeout` na poziomie sesji, aby wskazać, że polecenie DDL powinno poczekać na zwolnienie blokady na obiekcie. Oto przykład:

```
SQL> alter session set ddl_lock_timeout=30;
Session altered.
```

³ Ponieważ tabela `test` ma teraz dwie kolumny, próba wykonania kolejnego wstawienia w postaci `insert into test values (wartosc);` będzie powodować błąd, zatem najlepszym rozwiązaniem w jeszcze trwającej sesji nr 1 byłoby korzystać ze składni polecenia jawnie podającej listę kolumn: `insert into test(nazwa_kolumny) values (wartosc);` — *przyj. tłum.*

⁴ Polska wersja komunikatu błędu to: `ORA-00054: zasób zajęty, a zlecono uzyskanie z NOWAIT lub upłynął limit czasu` — *przyj. tłum.*

Nowa funkcjonalność dodawania kolumn

Oracle Database 11g pozwala teraz na dodanie do tabeli kolumny typu NOT NULL z domyślną wartością w jednej szybkiej i prostej operacji. Wartość domyślna jest pamiętana w bazie jako metadana i w rzeczywistości nie jest przechowywana fizycznie w tabeli. Dzięki temu proces dodawania kolumny z domyślną wartością jest znacznie szybszy.

Arup radzi...

To bardzo pożądana własność z punktu widzenia dostępności. Przed Oracle 11g, kiedy dodawaliśmy kolumnę z ograniczeniem NOT NULL i z domyślną wartością, Oracle w rzeczywistości umieszczał tę wartość w każdym wierszu tabeli z osobna. We *wszystkich* wierszach, a! Wyobraźmy sobie multimilionowowierszową tabelę, w której dane były uaktualniane kilkanaście milionów razy. Ileż informacji wycofania (*undo*) i powtórzenia (*redo*) zostało wygenerowanych! Co więcej, na tabeli na cały czas trwania operacji została założona blokada, aby zapobiec operacjom DDL. Wszystko to powodowało niezłe zamieszanie wśród użytkowników.

W Oracle 11g polecenie `alter table pracownicy add(stopien varchar2(1) default 'X' not null)` w rzeczywistości nie powoduje jednoczesnego uaktualnienia wartości w tabeli i modyfikacji jej struktury. Można się o tym przekonać, ustawiając zdarzenie (*event*) 10046 przed uruchomieniem polecenia DDL i sprawdzając plik śladu po wykonaniu:

```
SQL> alter session set events '10046 trace name context forever, level 16';
Session altered.
SQL> alter table pracownicy add(stopien varchar2(1) default 'X' not null);
Table altered.
SQL> alter session set events '10046 trace name context off';
Session altered.
```

Teraz należy sprawdzić plik śladu; nie znajdziemy tam odniesienia do wyrażenia `update pracownicy...` Takie zachowanie, czyli nie ustawianie fizycznie wartości kolumny, skutkuje znaczącym zmniejszeniem rozmiarów informacji wycofania i powtórzenia oraz kończy się szybciej.

Kolejne przydatne zastosowanie tej własności objawia się podczas dodawania do tabeli kolumny NOT NULL bez wartości domyślnej.

Słowo na koniec

W niniejszym rozdziale poruszyliśmy wiele zagadnień. Jak widać, Oracle Database 11g jest pełen porządnego „towaru”. Myślę, że niektórymi z najbardziej przydatnych własności są te najprostsze, jak opcja DDL WAIT (widać, że brakuje jej skomplikowania, jakim cechuje się część innych własności!) lub nowa możliwość dodawania kolumn. Zatwardziali użytkownicy ASM również znajdą nowe funkcje pomocnymi. Arup także wykonał wystrzałową robotę w tym rozdziale, gromkie brawa! Tak czy owak, taki nawał informacji, a to jeszcze nie koniec. Rozdział 3. już czeka w bloku startowym, a zapewnia jeszcze lepszą zabawę — jesteście w stanie w to uwierzyć? Trzymajcie się mocno, to będzie ostra jazda!