

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

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

Oracle Database 10g. Administracja bazy danych w Linuksie

Autor: Edward Whalen

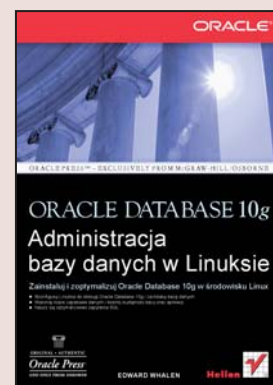
Tłumaczenie: Maciej Jezierski, Marek Pańczyński

ISBN: 978-83-246-1197-3

Tytuł oryginału: [Oracle Database 10g](#)

[Linux Administration](#)

Format: B5, stron: 464



Zainstaluj i zoptymalizuj Oracle Database 10g w środowisku Linux

- Skonfiguruj Linuksa do obsługi Oracle Database 10g i zainstaluj bazę danych
- Wykonaj kopie zapasowe danych i dostrój wydajność bazy oraz aplikacji
- Naucz się optymalizować zapytania SQL

Nazwa Oracle to już niemal synonim bezpiecznej, wydajnej i stabilnej bazy danych. Baza Oracle wykorzystywana jest wszędzie tam, gdzie wymagany jest wysoki poziom bezpieczeństwa danych, stabilność, szybkość wykonywania zapytań i wysoka wydajność mechanizmów zarządzających. Udostępnienie danych użytkownikom sieci wymaga zainstalowania bazy na serwerze obsługiwanym przez wydajny system operacyjny, za jaki od dawna uważany jest Linux. Ten dostępny nieodpłatnie system doskonale nadaje się do roli „bazy” dla Oracle’a, łatwo poddaje się „tuningowi” i daje możliwość korzystania z doskonałej pomocy technicznej w postaci setek tysięcy użytkowników na całym świecie.

Książka „Oracle Database 10g. Administracja bazy danych w Linuksie” stanowi kompleksowe omówienie zagadnień związanych z instalacją bazy danych Oracle 10g w środowisku Linux, strojeniem jej wydajności, codziennymi zadaniami administracyjnymi i korzystaniem z niej w celach produkcyjnych. Czytając ją, poznasz architekturę Linuksa i Oracle’a, dowiesz się, jak zainstalować i skonfigurować bazę danych oraz związane z nią usługi, nauczysz się optymalnie integrować Linuksa z Oraclem. Nabyta wiedza pozwoli Ci lepiej zabezpieczać dane, planować i realizować proces ich replikacji, administrować kontami użytkowników, tworzyć klastry aplikacji, optymalizować zapytania SQL i budować indeksy baz danych. Przeczytasz tu także o konfigurowaniu serwerów i korzystaniu z Oracle Application Server 10g.

- Architektura bazy danych Oracle
- Tworzenie baz i tabel
- Instalacja Oracle 10g w Linuksie
- Konfigurowanie systemu i bazy danych
- Wdrożenie klastra aplikacji w oparciu o mechanizm RAC
- Optymalizowanie bazy danych
- Tworzenie kopii zapasowych i odzyskiwanie danych
- Mechanizmy replikacji danych
- Optymalizacja zapytań SQL
- Konfiguracja warstwy sprzętowej

Zaprojektuj i wdrażaj wysoko wydajne rozwiązania bazodanowe



Spis treści

O autorze	11
O współautorach	13
Wstęp	15
Część I Oracle na Linuksie	27
Rozdział 1. Architektura Linuksa	29
Przegląd systemu operacyjnego	30
Jądro monolityczne	30
Mikrojądro	32
Przegląd Linuksa	32
Struktura katalogów	33
Narzędzia i katalogi	33
Środowisko programistyczne	37
Interfejs użytkownika w Linuksie	41
X Window	41
Środowiska graficzne	41
Narzędzia serwera Web w Linuksie	43
Serwer sieci Web Apache	43
Programowanie CGI w Linuksie	43
Proces uruchamiania Linuksa	43
Kod źródłowy Linuksa	44
Ponowne budowanie jądra Linuksa	44
Procesy i wątki	46
Czym jest proces?	46
Korzystanie z procesów przez Oracle'a	47
Czym jest wątek?	47
Urządzenia	47
Urządzenia znakowe	47
Urządzenia blokowe	48
System zarządzania przestrzenią dyskową LVM	48
System zarządzania pamięcią masową ASM	48
Systemy plików	49
ext3	49
OCFS	49
Surowe urządzenia	50
Podsumowanie	50

Rozdział 2. Architektura Oracle'a	51
Server Oracle Database	51
Baza danych Oracle Database	52
Fizyczne struktury danych	52
Logiczne struktury danych	56
Instancja Oracle'a	74
Definicja instancji Oracle'a	74
Składniki instancji	74
Procesy użytkownika	78
Proces serwera	78
Pamięć globalnego obszaru programu (PGA)	79
Serwer wielowątkowy (MTS) a dedykowane procesy serwera	79
Nowe funkcjonalności w Oracle 10g	80
Funkcjonalności związane ze skalowalnością i wydajnością	80
Klastrowanie	80
Dostępność	81
Zarządzanie	81
Funkcjonalności business intelligence	81
Tworzenie aplikacji	82
Obsługa 64-bitowego Linuksa	82
Przetwarzanie siatkowe	82
Podsumowanie	83
Rozdział 3. Zarządzanie transakcjami oraz obiekty schematu Oracle'a	85
Funkcjonalność Oracle'a	85
Odczyt z bazy danych	85
Współbieżność	86
Zapis do bazy danych	86
Blokowanie	87
Spójność odczytu	87
Integralność danych	87
Dziennik powtórzeń	88
Cofanie	95
Wprowadzenie do zarządzania transakcjami	97
Transakcje	98
Wprowadzenie do obiektów schematu Oracle'a	102
Tabele	102
Indeksy	103
Widoki	104
Sekwencje	104
Klastry	105
Synonimy	105
Podsumowanie	105
Część II Instalacja Oracle 10g w Linuksie	107
Rozdział 4. Określanie wymagań Oracle 10g w systemach Linux	109
Wprowadzenie do określania wymagań	110
Określanie wymagań dla szczytowego i równego obciążenia	110
Umowy o poziomie usług	111
Zasoby systemowe, możliwości i kolejkowanie	113
Wydajność procesora a kolejkowanie	113
Wydajność wejścia-wyjścia a kolejkowanie	114
Wydajność sieci	115
Pamięć	115

Metryki	116
Dostosowanie Oracle'a	117
Parametry ograniczające	117
Parametry zasobów	119
Określanie wymagań sprzętowych	121
Określanie wymagań procesora	121
Określanie wymagań pamięci	122
Określanie wymagań podsystemu wejścia-wyjścia	122
Podsystemy dyskowe RAID	130
Sprzętowy a programowy RAID	130
Paskowanie	131
Przegląd parzystości	135
Przewidywanie obciążenia	139
Podsumowanie	139
Rozdział 5. Konfigurowanie systemu Linux dla Oracle'a	141
Czynności niezbędne przed instalacją Linuksa	141
Minimalne wymagania	142
Konfiguracja systemów plików	143
Wybór komponentów Linuksa	143
Czynności niezbędne po instalacji Linuksa	146
Konfiguracja parametrów jądra	146
Dodawanie użytkowników i grup	148
Konfiguracja ograniczeń powłoki dla użytkownika Oracle'a	149
Tworzenie struktury katalogów	149
Konfigurowanie środowiska Oracle'a	150
Lista kontrolna instalacji	151
Lista kontrolna instalacji dla Oracle 10g R1	151
Lista kontrolna instalacji dla Oracle 10g R2	153
Podsumowanie	154
Rozdział 6. Instalacja Oracle 10g na Linuksie	155
Czynności wykonywane przed instalacją	155
Parametry konfiguracyjne jądra	155
Wymagania pamięci masowej	156
Proces instalacji	166
Przygotowywanie plików instalacyjnych	167
Uruchamianie programu instalacyjnego	167
Czynności wykonywane po instalacji	170
Pobieranie oraz instalacja poprawek	170
Czynności konfiguracyjne	170
Testowanie i sprawdzanie poprawności	172
Podsumowanie	173
Część III Oracle RAC w Linuksie	175
Rozdział 7. Koncepcje klastra Oracle Real Application Cluster	177
Czym jest klastrowanie?	178
Terminologia klastrów	178
Definicja klastra	180
RAC (Real Application Clusters)	182
Przegląd RAC	182
Komponenty Oracle RAC 10g	184
Komponenty bazy danych RAC	188

Wykorzystanie współużytkowanej pamięci masowej w Oracle RAC	191
Wykorzystanie NAS jako współużytkowanej pamięci masowej	193
Podsumowanie	194
Rozdział 8. Uruchomienie klastra aplikacji Oracle w systemie Linux	195
Planowanie wdrożenia systemu RAC	195
Komponenty RAC	196
Określanie skali systemu RAC	198
Konfiguracja systemu Linux	200
Sprawdzenie sprzętu	200
Sprawdzenie wymagań sieciowych	201
Konfiguracja jądra systemu Linux	202
Konfigurowanie systemu Linux	205
Konfiguracja modułów (programów) systemu Linux	207
Katalogi Oracle'a	207
Wyznaczanie zmiennych środowiskowych	208
Parametry pracy sieciowej	208
Konfiguracja klastra RAC	211
Konfiguracja usług CRS	212
Konfiguracja połączenia wewnętrznego	214
Konfiguracja współdzielonych dysków	214
Testowanie klastra RAC i zarządzanie nim	217
Dokumenty pomocy dla Oracle 10g RAC	221
Podsumowanie	222
Rozdział 9. Administrowanie serwerem Oracle i klastrem RAC w systemie Linux	223
Konfigurowanie jądra systemu Linux	223
Konfiguracja pamięci współdzielonej	223
Konfiguracja semaforów i innych parametrów jądra	224
Monitorowanie i modyfikowanie parametrów konfiguracyjnych systemu Linux	224
Konfiguracja sieci	225
Monitorowanie pracy systemu Linux	226
Polecenie sar	226
Polecenie vmstat	227
Polecenie top	227
Polecenie iostat	228
Aktualizacja systemu operacyjnego Linux	229
Uruchamianie instancji	229
Uruchomienie bazy danych	230
Zamontowanie bazy danych	230
Otwarcie bazy danych	231
Uruchomienie bazy danych za pomocą aplikacji Enterprise Manager Database Control	232
Baza danych Oracle Database 10g i demon CSSD	233
Uwierzytelnianie użytkowników	233
Uwierzytelnianie w bazie danych	234
Uwierzytelnianie w systemie operacyjnym	234
Uwierzytelnianie sieciowe	234
Uwierzytelnianie zdalne	235
Uwierzytelnianie na poziomie warstwy pośredniej	235
Uwierzytelnianie SSL	235
Autoryzacja użytkowników	236
Profile użytkowników	236
Uprawnienia	237
Dodawanie kont użytkowników	238
Podsumowanie	239

Rozdział 10. Optymalizacja platformy Oracle i klastra RAC w systemie Linux	241
Metodologia zwiększania wydajności	242
Krok 1: Wstępna analiza	242
Krok 2: Monitorowanie systemu	243
Krok 3: Analiza wyników	243
Krok 4: Opracowanie hipotezy	244
Krok 5: Wyznaczenie propozycji rozwiązania	245
Krok 6: Implementacja rozwiązania	245
Krok 7: Testowanie i monitorowanie	246
Dalsza analiza i testowanie	246
Dostrajanie platformy Oracle w systemie Linux	247
Dostrajanie aplikacji	248
Dostrajanie instancji	248
Monitorowanie pracy instancji Oracle'a	251
Zdarzenia oczekiwania	258
Optymalizacja pracy urządzeń	258
Monitorowanie systemu Linux	258
Monitorowanie systemu Linux za pomocą polecenia PS	260
Monitorowanie systemu Linux za pomocą polecenia vmstat	261
Monitorowanie systemu Linux za pomocą polecenia iostat	261
Monitorowanie systemu Linux za pomocą polecenia sar	262
Monitorowanie systemu Linux za pomocą polecenia top	263
Wykorzystanie modeli wielkich pamięci	264
Konfiguracja systemu Linux do obsługi SGA o rozmiarze większym niż 1,7 GB (i mniejszym niż 2,7 GB)	264
Konfiguracja serwera Oracle do obsługi SGA o rozmiarze większym niż 2,7 GB	265
Testy obciążeniowe sprawdzające wydajność systemu	266
Jaki jest cel testowania systemu?	267
Testy obciążeniowe zadań wsadowych	268
Testy obciążeniowe procedur wprowadzania danych	269
Testy obciążeniowe aplikacji OLTP	269
Metryki wydajności	271
Wskazówki dotyczące testowania obciążenia	273
Podsumowanie	273
 Część IV Administrowanie bazą danych Oracle Database 10g w systemie Linux oraz optymalizacja	 275
Rozdział 11. Kopie zapasowe i odbudowa bazy danych	277
Podstawy sporządzania kopii zapasowych	277
Techniki sporządzania kopii zapasowej i odbudowy bazy danych	278
Odbudowa instancji	279
Odbudowa danych — odbudowa nośnika	279
Odbudowa bieżącego i zarchiwizowanego dziennika powtórzeń	280
Jak działają transakcje?	282
Funkcjonowanie systemu RDBMS	283
Wprowadzenie do kopii zapasowych	286
Kopia typu ARCHIVELOG	286
Kopia typu CONTROLFILE	288
Pełna kopia	288
Kopie zapasowe przestrzeni tabel i pliku danych	289
Przyrostowe kopie zapasowe	290
Jak wykonać kopię zapasową bazy danych?	291
Menedżer przywracania danych RMAN	292
Automatyczne archiwizowanie i odtwarzanie dyskowe (obszar natychmiastowego odtwarzania)	296

Narzędzia systemu operacyjnego — gorąca kopia zapasowa	297
Narzędzia Export i Data Pump	299
Komercyjne aplikacje archiwizacji	302
Strategie sporządzania kopii zapasowych	302
Co i kiedy archiwizować?	303
Przykładowe strategie archiwizacji	306
Podsumowanie	307
Rozdział 12. Odtwarzanie danych	309
Odbudowa bazy danych	309
Odbudowa a odtwarzanie	310
Jak odbudować bazę danych?	310
Odbudowa instancji	311
Odbudowa nośnika	311
Odbudowa plików dziennika	313
Odbudowa bazy danych z określonego punktu czasowego	313
Technologia Flashback platformy Oracle 10g	315
Opracowanie strategii odbudowy	321
Planowanie odbudowy instancji	321
Planowanie odbudowy nośnika	322
Planowanie odbudowy po błędzie operatora	323
Podsumowanie	323
Rozdział 13. Oracle Data Guard w systemie Linux	325
Sposób wykorzystania mechanizmu Data Guard	325
Usługi Data Guard	327
Fizyczna zapasowa baza danych	330
Logiczna zapasowa baza danych	331
Konfiguracja systemu Oracle Data Guard i zarządzanie nim	333
Konfiguracja fizycznej zapasowej bazy danych	333
Konfiguracja logicznej zapasowej bazy danych	338
Podsumowanie	340
Rozdział 14. Zaawansowana replikacja danych w systemie Linux	341
Wprowadzenie do zagadnień replikacji Oracle	341
Replikacja z wieloma serwerami nadrzędnymi	343
Replikacja materializowanych widoków	344
Replikacja hybrydowa	344
Replikacja Oracle w systemie Linux	345
Wydajność układów wejścia-wyjścia	345
Projekt bazy danych	345
Dostateczna pojemność sieci	346
Podsumowanie	347
Rozdział 15. Konfiguracja sprzętowa systemu Linux i serwera Oracle	349
Podstawy optymalizacji sprzętowej	349
Wybór odpowiedniego sprzętu	350
Optymalizacja sprzętowa w systemie Linux	357
Optymalizacja pracy sieci w systemie Linux	362
Wybór odpowiedniej sieci	363
Zmiana ustawień systemu Linux w celu zwiększenia wydajności sieci	363
Podsumowanie	364

Rozdział 16. Optymalizacja instrukcji SQL	365
Analiza planu wykonywania instrukcji	365
Tabela planu	365
Przeglądanie planu wykonywania instrukcji	369
Analiza planu wykonywania instrukcji	371
Proces decyzyjny optymalizatora	373
Praktyczny przykład	375
Wpływanie na decyzje optymalizatora	376
Techniki optymalizacji SQL-a	381
Podsumowanie	392
Rozdział 17. Indeksy i partycjonowanie	393
Wprowadzenie do stosowania opcji indeksów bazy danych Oracle 10g	393
Opcje indeksu bazy danych Oracle 10g	395
Współczynnik zgrupowania indeksu	403
Partycjonowanie bazy danych Oracle 10g	404
Opcje partycjonowania indeksu	408
Opcje partycjonowania tabel IOT	410
Zwiększanie wydajności przez zastosowanie partycjonowanych indeksów	410
Podsumowanie	411
Część V Produkty firmy Oracle dla systemu Linux	413
Rozdział 18. Oracle Application Server 10g dla systemu Linux	415
Z jakich komponentów składa się serwer Oracle Application Server 10g?	416
Java 2 Enterprise Edition	416
Zarządzanie i bezpieczeństwo	417
Portale	417
Wireless	418
Inteligencja biznesowa	418
Integracja rozwiązań z grupy e-biznes	418
Wykorzystanie platformy Oracle Application Server 10g	418
Instalacja i konfiguracja Oracle Application Server 10g w systemie Linux	424
Instalacja	424
Konfiguracja	424
Podsumowanie	428
Rozdział 19. Oracle E-Business Suite w systemie Linux	429
Ogólne informacje na temat aplikacji Oracle E-Business Suite dla systemu Linux	429
Czym jest pakiet Oracle E-Business Suite?	430
Architektura platformy E-Business Suite	431
Które moduły są dostępne dla systemu Linux?	433
Wykorzystanie pakietu Oracle E-Business Suite w systemie Linux	433
Oprogramowanie E-Business Suite w środowisku RAC	434
Instalacja i konfiguracja oprogramowania E-Business Suite w systemie Linux	434
Instalacja	434
Przeniesienie platformy E-Business Suite do systemu Linux	437
Przeniesienie warstwy aplikacji	437
Przeniesienie warstwy baz danych	438
Przekształcenie warstwy baz danych w klaster RAC	439
Testowanie	440
Podsumowanie	440
Skorowidz	441

Rozdział 8.

Uruchomienie klastra aplikacji Oracle w systemie Linux

W poprzednim rozdziale zostały przedstawione ogólne założenia funkcjonowania klastra RAC systemu Oracle. Zapoznanie się z zasadami działania tego rozwiązania jest wstępem do rzeczywistej implementacji klastra RAC. W tym rozdziale zostaną opisane zagadnienia związane z uruchomieniem usług Cluster Ready Services, instalowaniem serwera baz danych Oracle w środowisku RAC oraz utworzeniem klastra.

Wszystkie opisywane procedury odnoszą się do oprogramowania Oracle 10g Real Application Cluster (RAC) dla systemu Linux. Tematyka rozdziału obejmuje zasady projektowania systemu RAC, konfigurowania systemu Linux dla potrzeb środowiska RAC oraz testowania systemu RAC. Zostały tu również omówione różnice w zapisie danych i informacji klastra w systemie plików klastra Oracle (ang. *Oracle Cluster File System — OCFS*) na nieformatowanych urządzeniach (ang. *raw devices*) oraz w systemach ASM. W końcowej części rozdziału zostały z kolei przedstawione podpowiedzi, które umożliwiają wybranie najlepszej platformy systemowej dla danej implementacji.

Planowanie wdrożenia systemu RAC

Pierwszym etapem wdrażania systemu DBMS jest utworzenie planu, który umożliwia wybór najważniejszych opcji procedury. Dlatego też w pierwszej kolejności zostanie przedstawiony przegląd komponentów RAC — na wypadek gdyby ktoś pominął poprzedni rozdział.

Komponenty RAC

Klaster RAC składa się z kilku istotnych komponentów. Węzły klastra wyznacza od 2 do 32 serwerów, które korzystają ze wspólnego systemu składowania danych. Poszczególne rodzaje systemów składowania danych zostały szczegółowo opisane w poprzednim rozdziale.

Jednym z ważniejszych komponentów klastra RAC są usługi Cluster Ready Services (CRS). Ich uruchomienie jest konieczne przed zainstalowaniem serwera Oracle i utworzeniem bazy danych. Usługi te składają się z plików binarnych zapisanych w każdym węźle klastra oraz plików konfiguracyjnych umieszczonych na wybranym dysku systemu składowania danych. Oprogramowanie CRS musi być instalowane we własnym katalogu *ORACLE_HOME*.

Po zainstalowaniu usługi CRS są uruchamiane automatycznie. Choć istnieje również możliwość ręcznego ich uruchomienia. Usługi CRS wykorzystują w pracy specjalny plik zarządzania węzłami klastra (tzw. *voting disk*) i rejestr klastra Oracle (ang. *Oracle Cluster Registry* — *OCR*). W każdym węźle działa tylko jeden zbiór usług CRS, niezależnie od liczby zainstalowanych instancji Oracle'a. Do usług CRS zalicza się trzy demony: *CRSD*, *OCSSD* i *EVMD*. Proces *CRSD* zapewnia realizację operacji zarządzania węzłem (na przykład obsługę rejestru *OCR*) oraz odbudowy węzła (w celu zapewnienia nieprzerwanej pracy systemu). Odpowiada również za zarządzanie zasobami aplikacji. Pracuje z prawami użytkownika *root* i automatycznie wznowia działanie po wystąpieniu awarii. Proces *OCSSD* wykonuje wszystkie operacje związane z przynależnością węzła do klastra. Jest uruchamiany z prawami użytkownika *oracle*. Błąd w działaniu procesu wiąże się z ponownym uruchomieniem klastra. Demon *EVMD* odpowiada za zarządzanie zdarzeniami. Do obsługi wywołań wykorzystuje proces *racgenvt*. Czwarty demon — *oprocld* — pracuje tylko na platformach, na których nie działa oprogramowanie klastra dostarczane przez producenta. Proces *OPROCRD* odpowiada za monitorowanie klastra.

Kolejnym bardzo ważnym elementem klastra RAC jest wirtualny adres IP. Każdy serwer operuje trzema adresami IP. Jeden z nich jest przeznaczony do publicznego (ogólnego) dostępu do bazy danych. Drugi gwarantuje prywatne połączenie wewnętrzne o dużej szybkości transmisji danych. Trzeci z adresów jest adresem wirtualnym (ang. *Virtual IP* — *VIP*). Adres *VIP* umożliwia bezkonfliktowe przeniesienie ruchu do innego węzła w przypadku awarii. W jednym systemie można zdefiniować wiele wirtualnych adresów IP, przy czym wirtualny adres IP i publiczny adres IP muszą należeć do tej samej podsieci i muszą być wymienione w pliku */etc/hosts* oraz w serwerze nazw domenowych (DNS).

Planując architekturę systemu RAC, trzeba znać założenia własnej aplikacji. Kolejne zadanie sprowadza się więc głównie do poznania własnego serwisu. Brak gruntownej wiedzy na temat własnej aplikacji może być przyczyną podjęcia niewłaściwych decyzji podczas opracowywania architektury RAC. Oto kilka pytań, na które należałoby sobie odpowiedzieć:

- ♦ Czy użytkownicy będą korzystali z bazy danych za pomocą połączeń bezpośrednich, czy za pośrednictwem warstwy pośredniej (serwera aplikacji)?
- ♦ Ile aktywnych połączeń z bazą danych trzeba będzie zagwarantować?
- ♦ Czy działanie aplikacji wiąże się w większości z operacjami odczytu, czy też liczba operacji zapisu przeważa nad liczbą operacji odczytu?

- ♦ Czy w firmie są zatrudnieni specjaliści ds. zarządzania urządzeniami SAN i administratorzy baz danych? Czy konieczne będzie zatrudnienie pracowników odpowiedzialnych za zarządzanie klastrem RAC?
- ♦ Czy dany system będzie istotnie rozbudowywany w ciągu kilku przyszłych miesięcy?
- ♦ Jaki będzie rozmiar baz danych?
- ♦ Czy będą wykorzystywane archiwa dzienników? Jeśli tak, gdzie będą przechowywane pliki dzienników?
- ♦ Czy wydajność i niezawodność są istotnymi aspektami funkcjonowania aplikacji?
- ♦ W jaki sposób będą sporządzane kopie zapasowe systemu?
- ♦ Czy wymagania odnośnie niezawodności systemu narzucają konieczność wdrożenia rozwiązania Data Guard w klastrze RAC?

Planując implementację klastra RAC, należy pamiętać o wymienionych kwestiach. Przeanalizujmy je więc bardziej szczegółowo. Czy będzie wykorzystywany serwer aplikacji (serwer warstwy pośredniej)? Brak serwera pośredniego (serwera aplikacji) może spowodować zwiększenie liczby połączeń bazodanowych. Zastosowanie takich rozwiązań jak pula połączeń (ang. *connection pool*) pozwala serwerowi aplikacji na przejęcie kontroli nad połączeniami bazodanowymi i na współdzielenie ich przez wszystkich użytkowników, co skutkuje zmniejszeniem liczby potrzebnych połączeń bazodanowych. Jeśli aplikacja sieciowa jest zaprojektowana w taki sposób, że wszyscy użytkownicy ustanawiają bezpośrednie połączenia z bazą danych, konieczne może się okazać wprowadzenie zmian w parametrach systemu Linux.

Ile połączeń z bazą danych będą zestawiali użytkownicy aplikacji? W przypadku dużej liczby połączeń często zachodzi konieczność odmiennego zaprojektowania poszczególnych z nich.

Czy działanie aplikacji wiąże się w większości z operacjami odczytu, czy też liczba operacji zapisu przeważa nad liczbą operacji odczytu? Intensywność zapisu informacji w bazie danych determinuje rodzaj macierzy RAID, jaki trzeba zastosować we współdzielonym systemie składowania danych. Duża częstotliwość zapisu może na przykład być przyczyną wyeliminowania macierzy RAID 5.

Jakie jest doświadczenie personelu? Ustalenie odpowiedzi na to pytanie jest bardzo ważne. Na przykład zastosowanie systemu SAN nie ma sensu, jeśli w firmie nie ma osób znających tę technologię. Oczywiście można zlecić zarządzanie firmom zewnętrznym lub zatrudnić nowych pracowników, ale to zwiększa koszty przedsięwzięcia.

Czy system zostanie istotnie rozbudowany w ciągu kilku najbliższych miesięcy lub roku? Odpowiedź na to pytanie bezpośrednio wpływa na skalę implementacji klastra RAC. Nietrudno sobie wyobrazić sytuację, w której bieżąca działalność przedsiębiorstwa jest obsługiwana przez dyski jednej półki macierzy SAN i w której rozmiar bazy danych zwiększa się stukrotnie w ciągu jednej nocy! Jeśli planowane jest rozszerzenie klastra RAC z dwóch węzłów do trzech, trzeba się upewnić, że urządzenia takie, jak przełączniki ethernetowe i przełączniki Fiber Chanel będą w stanie obsłużyć trzeci węzeł.

Zagadnienie to wiąże się z kolejnym problemem. Jaki będzie rozmiar bazy danych? Rozmiar bazy danych jest najważniejszym parametrem podczas projektowania klastra RAC. Określając rozmiar pamięci masowej, należy wziąć pod uwagę cały system. W całkowitym rozmiarze pamięci masowej trzeba uwzględnić rozmiary tabel, indeksy, dzienniki powtórzeń (ze wszystkich węzłów), archiwa dzienników i pliki kworum. Dodatkowo zgodnie z zaleceniami każdy węzeł powinien dysponować przynajmniej jedną partycją przeznaczoną dla dzienników powtórzeń. Partycje takie powinny być wyznaczane na niezależnych dyskach LUN. Na rozmiar współdzielonej pamięci masowej ma wpływ także rodzaj wybranej macierzy RAID. Więcej informacji na ten temat znajduje się w kolejnym punkcie.

Czy wydajność lub niezawodność są kluczowymi aspektami projektowania aplikacji? Odpowiedź na to pytanie determinuje rodzaj mechanizmu reagowania na awarie, który musi zostać przygotowany. Powinna być również uwzględniona podczas podejmowania decyzji o rodzaju macierzy RAID. Bywa też istotna w czasie ustalania, czy należy zgrupować karty sieciowe, czy nie.

Następny krok procedury planowania wymaga ustalenia, czy współdzielone pliki będą przechowywane na nieformatowanych dyskach, w systemie OCFS, czy w systemie ASM. Nieformatowane dyski zapewniają największą wydajność, ale znacznie ograniczają możliwość zarządzania plikami. Dostęp do nieformatowanych dysków ma tylko oprogramowanie Oracle. Ich zawartość nie może być zmieniana za pomocą narzędzi systemu operacyjnego Linux. Ewentualna próba edycji danych może się zakończyć uszkodzeniem pliku. Automatyczne zarządzanie pamięcią masową (ang. *Automatic Storage Management — ASM*) jest nowym mechanizmem przechowywania danych Oracle'a. Jak wszystkie nowe produkty musi zapracować na zaufanie użytkowników. Dla większości instalacji najlepszym rozwiązaniem wydaje się więc system plików OCFS. Tylko konieczność uzyskania wyjątkowo dużej wydajności jest usprawiedliwieniem, aby zastąpić go nieformatowanymi dyskami.

Określanie skali systemu RAC

Wielu projektantów systemów traktuje proces szacowania skali przedsięwzięcia jak wiedzę tajemną i poprzestaje na zgromadzeniu jak największej ilości danych statystycznych, porównawczych i specyfikacji technicznych. Określenie skali systemu RAC jest jednak czymś więcej niż zadaniem zgromadzenia wszystkich wymienionych informacji.

Poniżej zostało wymienionych kilka zadań, które należy wykonać, chcąc poprawnie oszacować skalę systemu RAC lub dowolnego innego systemu.

- ♦ **Zdefiniowanie wymagań klienta.** Jest to jeden z najważniejszych kroków w całej procedurze. Odpowiada on za przygotowanie odpowiedniego punktu wyjścia dla dalszych operacji. Większość klientów nie dostarcza założeń aplikacyjnych „na tacy”. Dlatego zebranie odpowiednich informacji wejściowych wymaga dużej aktywności ze strony projektanta. Oczywiście konieczne jest tu przełożenie pewnych określeń biznesowych na terminologię informatyczną.
- ♦ **Analiza danych systemowych.** Zawsze należy postarać się o jak największą ilość informacji technicznych na temat pracy porównywalnego lub bieżącego systemu produkcyjnego. Dane te są przydatne nawet wówczas, gdy platforma sprzętowa systemu docelowego jest zupełnie inna lub gdy można się spodziewać, że nowy

system będzie się charakteryzował znacznie większą wydajnością. Należy zatem zebrać dane statystyczne na temat wykorzystania procesora i pamięci oraz o liczbie wykonywanych w ciągu sekundy operacji wejścia-wyjścia.

- ♦ **Opracowanie konfiguracji systemu.** Kolejny etap polega na zaprojektowaniu systemu w taki sposób, aby spełniał wymagania klienta. W projekcie muszą być jednak uwzględnione zgromadzone wcześniej informacje techniczne. Celem zadania jest oszacowanie stopnia wykorzystania procesora i pamięci oraz obciążenia podsystemu wejścia-wyjścia. Warto również określić potrzebne zasoby pamięci masowej oraz jej wydajność, biorąc pod uwagę obciążenie podsystemu wejścia-wyjścia oraz szacowany rozmiar danych klienta. W tym kroku uwzględniane są wszystkie wskazówki, „porady praktyczne” i testy porównawcze. Jeśli to możliwe, należy porównać przygotowane założenia z parametrami innego działającego systemu. W projekcie trzeba uwzględnić wymagania klienta. Upewnienie się we wstępnej fazie projektu, że opracowywany system spełnia wszystkie założenia, jest znacznie mniej kosztowne niż odkrycie, że jest inaczej, po zakupie potrzebnego wyposażenia.
- ♦ **Implementacja.** Wdrożenie projektu powinno się odbywać stopniowo. Dzięki temu podczas implementowania rozwiązania można zająć się optymalizacją jego pracy.

Podstawowe założenia odnośnie rozmiaru klastra RAC

Projektując klastr RAC dla platformy Oracle 10g, trzeba wziąć pod uwagę kilka parametrów skali przedsięwzięcia. Część z nich jest związana z wymaganiami narzucanymi przez oprogramowanie Oracle, a część jest wyznaczana w procesie prób i błędów oraz na podstawie doświadczenia projektanta.

Zaleca się spełnienie przynajmniej minimalnych wymagań dla instalowanej wersji oprogramowania RAC Oracle. Są one zmieniane wraz z udostępnianiem kolejnych wersji platformy, więc aby się z nimi zapoznać, trzeba pobrać stosowną dokumentację z serwisu Oracle’a.

Jednym z podstawowych założeń projektowych jest minimalna liczba partycji współdzielonej pamięci masowej. W systemie Linux dla klastra RAC Oracle 10g są wymagane co najmniej dwie partycje. Pierwsza z nich nie powinna być mniejsza niż 1 GB. Są na niej składowane pliki kworum usług CRS oraz dysk głosowania (ang. *voting disk*) i rejestr OCR. Zawartość tej partycji powinna być odseparowana od plików danych. Dzięki temu utrata informacji o pracy klastra nie spowoduje utraty samych danych. Dla rejestru OCR należy zarezerwować przestrzeń co najmniej 100 MB, a dla dysku głosowania — co najmniej 20 MB. Dodatkowo zaleca się przechowywanie rejestru i dysku głosowania w macierzy RAID, aby partycja ta była zawsze dostępna.

Druga partycja powinna zostać przeznaczona na pliki bazy danych Oracle, czyli na dzienniki powtórzeń, zarchiwizowane dzienniki, pliki przestrzeni tabel i pliki indeksów. Wyznaczenie właściwego rozmiaru tej partycji jest jednym z najważniejszych ustaleń projektowych.

Zgodnie z zaleceniami wymienianymi w artykułach poświęconych projektowaniu klastra RAC Oracle należy utworzyć następujące partycje lub dyski RAID:

- ♦ **Jedną partycję w każdym węźle dla dzienników powtórzeń.** W zasadzie najlepszym rozwiązaniem byłoby przeznaczenie na pliki dzienników powtórzeń dwóch dysków. Można by sądzić, że to rozrzutność, ale w systemach o naprawdę dużej wydajności

dwie partycje są niezbędne. Podczas przełączania dzienników dotychczasowy dziennik podlega archiwizacji, a nowy jest wykorzystywany do rejestracji bieżących informacji. Zastosowanie tylko jednego dysku RAID dla obydwu dzienników powoduje, że operacje wejścia-wyjścia nie są wykonywane sekwencyjnie. Z tego względu, mimo że udostępnienie tylko jednej partycji w węzle jest rozwiązaniem akceptowalnym, zaleca się stosowanie dwóch partycji.

- ♦ **Jedną partycję dla zarchiwizowanych dzienników.** Proces zapisu zarchiwizowanych dzienników ma charakter synchroniczny. Umieszczenie ich na oddzielnej partycji wyklucza więc możliwość obniżenia wydajności bazy danych.
- ♦ **Przynajmniej jedną partycję RAID 10 dla plików danych.** Firma Oracle zaleca stosowanie zasady SAME (ang. *Strip and Mirror Everything*) stanowiącej, że wszystkie dane powinny być przeplatane (RAID 0) i duplikowane (RAID 1). Im więcej dysków zostanie przeznaczonych na macierz RAID 10, tym większa jest szansa wystąpienia odwołania do niezajętego dysku podczas wykonywania kolejnych zapytań.

Konfiguracja systemu Linux

Po zapoznaniu się z podstawowymi informacjami na temat klastra RAC i zaplanowaniu jego wdrożenia można przystąpić do parametryzowania systemu Linux. Pierwszym zadaniem, które trzeba wykonać, jest sprawdzenie, czy wykorzystywany sprzęt spełnia wymagania klastra RAC Oracle 10g. Kolejna ważna czynność polega na odpowiednim dobraniu ustawień jądra systemu. Ostatnim zadaniem jest natomiast sama konfiguracja systemu Linux.

Doświadczenie administratorów Oracle RAC uczy, że najważniejszym etapem całej procedury jest właściwy dobór parametrów pracy sieci. Pomyłka może bowiem oznaczać konieczność ponownego zainstalowania oprogramowania Oracle, co z kolei wiąże się z kilkoma dodatkowymi dniami pracy. Warto zatem uważnie zapoznać się z informacjami prezentowanymi w kolejnych punktach podrozdziału, ponieważ dzięki nim można zaoszczędzić wiele godzin pracy.

Sprawdzenie sprzętu

Wykorzystywany system musi spełniać wymienione poniżej minimalne wymagania sprzętowe:

- ♦ 512 MB pamięci RAM,
- ♦ 1 GB przestrzeni wymiany (lub dwukrotny rozmiar pamięci RAM); w systemach wyposażonych w pamięć RAM o rozmiarze większym niż 2 GB przestrzeń wymiany może być taka sama jak pamięć RAM lub dwukrotnie większa,
- ♦ 400 MB przestrzeni dyskowej w katalogu */tmp*,
- ♦ do 4 GB przestrzeni dyskowej na oprogramowanie Oracle (w zależności od rodzaju instalacji),
- ♦ 5 GB współdzielonej pamięci masowej przeznaczonej na pliki dzienników powtórzeń i pliki kworum.

Aby sprawdzić, jaki jest rozmiar fizycznej pamięci RAM, którą dysponuje serwer, wystarczy wykonać następujące polecenie:

```
# grep MemTotal /proc/meminfo
```

Informacji na temat rozmiaru przestrzeni wymiany dostarcza instrukcja:

```
# grep SwapTotal /proc/meminfo
```

Chcąc ustalić, jaki jest rozmiar dostępnej przestrzeni dyskowej w katalogu */tmp*, należy wprowadzić polecenie:

```
# df -k /tmp
```

Jeżeli w katalogu */tmp* dostępna przestrzeń dyskowa nie przekracza 400 MB, konieczne jest wykonanie jednego z poniższych zadań.

- ♦ Usuń niepotrzebne pliki z katalogu */tmp*, tak aby zwolnić potrzebną przestrzeń dyskową.
- ♦ Ustaw wartości zmiennych środowiskowych *TEMP* i *TMPDIR* podczas konfigurowania środowiska pracy dla konta *oracle*.
- ♦ Rozszerz system plików obejmujący katalog */tmp*. Jeśli to konieczne, można się skontaktować w tej sprawie z administratorem systemu.

Aby określić ilość wolnego miejsca na dysku przeznaczonego na pliki binarne oprogramowania Oracle, trzeba wykonać instrukcję:

```
# df -m
```

Przedstawione poniżej polecenie pozwala na sprawdzenie, czy architektura danego systemu umożliwia uruchomienie oprogramowania Oracle.

```
# grep "model name" /proc/cpuinfo
```

Listing wyników przedstawionej instrukcji zawiera dane o rodzaju procesora, które należy porównać z wymaganiami instalacyjnymi oprogramowania Oracle.

Sprawdzenie wymagań sieciowych

Przed rozpoczęciem instalacji oprogramowania RAC Oracle należy się upewnić, że urządzenia sieciowe i przypisane im adresy IP spełniają założenia instalacyjne.



Uwaga

Aktualne informacje na temat protokołów sieciowych i urządzeń obsługiwanych przez oprogramowanie Oracle Real Application Cluster (RAC) są publikowane na stronach serwisu Certification w portalu Oracle MetaLink.

Wymagania odnośnie sprzętu sieciowego

Każdy węzeł klastra RAC musi spełniać następujące założenia:

- ♦ Każdy węzeł musi być wyposażony w co najmniej dwie karty sieciowe — jedną pełniącą rolę publicznego interfejsu sieciowego i jedną pracującą jako interfejs prywatny (w połączeniach wewnętrznych).
- ♦ Nazwy interfejsów przypisane kartom sieciowym muszą być jednakowe we wszystkich węzłach klastra.
- ♦ W celu zwiększenia niezawodności interfejsów sieciowych można w każdym węźle zgrupować interfejsy publiczne i prywatne.
- ♦ Każda karta interfejsu publicznego musi obsługiwać protokół TCP/IP.
- ♦ Urządzenia sieci prywatnej (wykorzystywane do połączeń wewnętrznych) muszą obsługiwać protokół UDP. Karty sieciowe i przełączniki muszą pozwalać na stosowanie protokołu TCP/IP z dużą przepustowością (zaleca się połączenia o gigabitowej przepływności lub jeszcze wydajniejsze).



Uwaga

Protokół UDP jest domyślnym protokołem połączeń wewnętrznych klastra RAC. Protokół TCP jest z kolei protokołem wymiany danych między usługami CRS.

Konfiguracja jądra systemu Linux

Pierwszym etapem konfiguracji jest ustalenie, która wersja jądra w ogóle zostanie wykorzystana. Lista certyfikowanych wersji jądra jest publikowana w serwisie Oracle MetaLink.

Po wyborze i zainstalowaniu odpowiedniego jądra trzeba przystosować je do potrzeb platformy Oracle 10g RAC. W tym celu należy przede wszystkim zagwarantować zgodność systemu z wymaganiami oprogramowania Oracle 10g RAC.

Sprawdzenie zgodności systemu z wymaganiami oprogramowania

Wersja systemu operacyjnego musi być zgodna z jedną z wymienionych opcji.

Oracle 10g R1

- ♦ Red Hat Enterprise Linux AS/ES 2.1 lub 3.0 (x86)
- ♦ United Linux 1.0, service pack 3 lub późniejszy (x86)

Oracle 10g R2

- ♦ Red Hat Linux AS/ES 3.0 lub 4.0
- ♦ SUSE Linux Enterprise Server 9.0

Wymagania odnośnie pakietów oprogramowania systemu operacyjnego dla klastra RAC są takie same, jak dla niezależnego serwera. Informacje na ten temat zostały zamieszczone w rozdziale 5.

System plików klastra Oracle (OCFS) wersja 1.0.9-12 lub późniejsza

- ♦ *ocfs-support*
- ♦ *ocfs-tools*
- ♦ *ocfs-kernel_wersja*.



Uwaga

Oprogramowanie OCFS jest potrzebne tylko wówczas, gdy pliki bazy danych mają być zapisywane w klastrowym systemie plików. Jeżeli będą one przechowywane na nieformatowanych urządzeniach lub w systemie ASM, instalacja pakietów OCFS jest zbędna.

Aby się upewnić, że dany system spełnia wymienione wymagania, wystarczy wykonać kilka przedstawionych poniżej czynności.

1. Informacje o zainstalowanej dystrybucji i wersji systemu Red Hat Linux są wyświetlane po wprowadzeniu polecenia:

```
cat /etc/issue
```

lub

```
uname -r
```

2. Wykonanie przedstawionych poleceń w systemie SLES 8 dostarcza również informacji o tym, czy został zainstalowany pakiet Service Pack 3. Wersja jądra nie może być wcześniejsza niż *2.4.21-138-default*. W przypadku wcześniejszych wersji jądra należy skontaktować się dystrybutorem systemu SLES i uzyskać informacje na temat sposobu zainstalowania pakietu Service Pack 3.
3. Aby sprawdzić, czy są zainstalowane wymagane pakiety oprogramowania, wystarczy wykonać instrukcję przedstawioną poniżej.

```
rpm -q nazwa_pakietu
```

Jeżeli dany pakiet nie jest dostępny w systemie, należy go zainstalować z płyty dystrybucyjnej Linux.

4. Aby sprawdzić, czy pakiety systemu OCFS zostały zainstalowane, można wprowadzić następujące polecenie:

```
rpm -qa | grep ocfs
```

Jeżeli baza danych powinna zostać zainstalowana na systemie plików OCFS, a w systemie operacyjnym nie ma pakietów OCFS, można je pobrać ze strony <http://oss.oracle.com/projects/ocfs/>.

Sprawdzenie wymaganych nakładek

Część z instalowanych produktów wymaga zainstalowania odpowiednich nakładek na system. Przed zainstalowaniem oprogramowania trzeba więc zapoznać się z aktualną listą wymaganych nakładek publikowaną w dokumentacji Oracle'a.



Uwaga

Uniwersalny instalator Oracle'a (Oracle Universal Installer) sprawdza, czy system spełnia wymagania oprogramowania. Aby mieć pewność, że weryfikacja zostanie zakończona pomyślnie, należy zagwarantować zgodność systemu z wymaganiami przed uruchomieniem instalatora lub uruchomić instalator w trybie preinstalacji — z użyciem polecenia `runInstaller -executeSysPrereqs` — i wówczas zainstalować brakujące nakładki i pakiety systemu operacyjnego.

Ustawienie parametrów jądra i powłoki

Standardowa instalacja Oracle'a w systemie Linux gwarantuje poprawną pracę oprogramowania. Jednak przed uruchomieniem klastra RAC warto się upewnić, że pewnym parametrom jądra zostały przypisane co najmniej minimalne wymagane wartości.



Uwaga

Wymienione w kolejnych punktach parametry jądra i ograniczenia powłoki są jedynie ustawieniami zalecanymi. W przypadku użytkowych systemów baz danych należy je zmodyfikować w taki sposób, aby system pracował z jak największą wydajnością. Wspomniane parametry jądra i ograniczenia powłoki muszą zostać zmienione we wszystkich węzłach klastra.

Zmiana parametrów jądra

Zmiana parametrów jądra jest jedną z najważniejszych czynności w procesie wdrażania klastra RAC.

Za pomocą dowolnego edytora tekstu należy utworzyć lub zmodyfikować plik `/etc/sysctl.conf`, dodając do niego lub edytując następujące wiersze:

```
kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmni = 4096
kernel.sem = 4096
fs.file-max = 131072
net.ipv4.ip_local_port_range = 1024 65000
```

Zapisanie wymienionych dyrektyw w pliku `/etc/sysctl.conf` zapewnia ich uwzględnianie podczas każdego uruchomienia systemu.

Aby uaktywnić wartości nowo wprowadzonych ustawień jądra, można zastosować polecenie:

```
sysctl -p
```

Listing wynikowy instrukcji należy porównać z wartościami określonymi w wymaganiach systemowych. Jeżeli ustawione wartości są niewłaściwe, należy zmodyfikować dyrektywy pliku `/etc/sysctl.conf` i ponownie wykonać powyższą instrukcję. W systemach SLES, aby plik `/etc/sysctl.conf` był uwzględniany podczas każdego uruchomienia, konieczne jest wykonanie polecenia:

```
/sbin/chkconfig boot.sysctl on
```

Procedurę tę należy powtórzyć we wszystkich węzłach klastra.

Wyznaczenie ograniczeń powłoki dla konta oracle

Zwiększenie ograniczeń powłoki systemowej dla konta `oracle` pozwala na zwiększenie wydajności oprogramowania. Zagadnienie to zostało opisane w rozdziale 5. — procedura konfiguracji klastra RAC nie różni się pod tym względem od obowiązującej dla niezależnych systemów Oracle'a. Z tego też względu nie zostanie tutaj powielona.

Konfigurowanie systemu Linux

Procedura tworzenia kont dla oprogramowania Oracle jest zbliżona do obowiązującej w niezależnych systemach Oracle'a. Bardzo ważne jest jednak, aby konta były tworzone w taki sam sposób we wszystkich węzłach klastra RAC. We wszystkich węzłach identyfikatory użytkowników i grup związanych z kontami muszą być bowiem jednakowe. Konieczne jest również poprawne sparametryzowanie mechanizmów `ssh` lub `rsh`.

Utworzenie konta

Przygotowując system do instalacji klastra RAC, trzeba utworzyć kilka kont użytkowników i grup systemu Linux. Liczba tych kont zależy od tego, czy w danym systemie było już instalowane oprogramowanie Oracle oraz od rodzaju instalowanych produktów. Wymienione poniżej konta użytkowników i grup są wymagane w przypadku instalowania bazy danych Oracle.

Grupa administratorów bazy danych (dba)

To konto grupowe trzeba utworzyć podczas pierwszej instalacji oprogramowania bazy danych Oracle w danym systemie. Obejmuje ono użytkowników systemu Linux, którzy mają prawa administrowania bazą danych (uprawnienia `SYSDBA`). Domyślną nazwą grupy jest `dba`. Aby wyznaczyć inną nazwę niż standardowa, trzeba wybrać instalację typu *Custom* lub uruchomić program instalatora z prawami użytkownika, który nie należy do tej grupy. Instalator zgłosi wówczas konieczność określenia nazwy dla grupy administratorów.

Grupa operatorów bazy danych (oper)

To konto grupowe ma charakter opcjonalny. Jego powołanie jest uzasadnione wtedy, gdy trzeba pewnej grupie użytkowników systemu Linux nadać ograniczone prawa administrowania bazą danych (prawa `SYSOPER`). Domyślnie członkowie grupy `dba` mają jednocześnie uprawnienia grupy `oper`. Jeśli konieczne jest utworzenie grupy operatorów innej niż `dba`, trzeba wybrać instalację typu *Custom* lub uruchomić program instalatora z prawami użytkownika, który nie jest członkiem grupy `dba`. Instalator umożliwi wówczas wprowadzenie nazwy grupy. Zazwyczaj wybieraną nazwą dla tej grupy jest `oper`.

Konto użytkownika i grupy opisywane jako kolejne są wymagane dla wszystkich rodzajów instalacji.

Grupa inwentarza Oracle (oinstall)

Grupę tę trzeba utworzyć podczas pierwszej instalacji oprogramowania Oracle w systemie. Standardową nazwą dla tej grupy jest `oinstall`. Użytkownicy grupy `oinstall` są właścicielami inwentarza Oracle (ang. *Oracle Inventory*), czyli katalogu wszystkich komponentów oprogramowania Oracle zainstalowanych w systemie.



Uwaga

Jeżeli oprogramowanie Oracle jest już zainstalowane, istniejąca grupa inwentarza musi być podstawową grupą użytkownika, który instaluje nowe oprogramowanie. W kolejnych punktach został opisany sposób ustalenia nazwy istniejącej grupy inwentarza.

Właściciel oprogramowania Oracle (oracle)

Konto użytkownika musi zostać utworzone podczas pierwszej instalacji oprogramowania Oracle w systemie. Użytkownik ten staje się właścicielem wszystkich plików zainstalowanej platformy. Zazwyczaj wybieraną nazwą dla konta jest `oracle`. Podstawową grupą użytkownika `oracle` musi być natomiast grupa inwentarza. Grupami dodatkowymi muszą być grupy administratorów bazy danych i operatorów bazy danych.

Grupa inwentarza musi być wykorzystywana we wszystkich operacjach instalacji jakiegokolwiek oprogramowania Oracle. Wszystkie instalacje uzupełniające (następujące po pierwszej instalacji platformy) również muszą być wykonane przez użytkownika należącego do tej grupy. Można jednak utworzyć inne konto właściciela platformy Oracle, grup administratorów, grup operatorów (inne niż `oracle`, `dba` i `oper`) dla różnych instalacji. Wyznaczenie różnych grup dla różnych instalacji daje gwarancję, że administratorzy baz danych uzyskują dostęp tylko do właściwych im baz danych, a nie do wszystkich baz występujących w systemie.

W czasie pierwszej instalacji oprogramowania Oracle w systemie instalator tworzy plik o nazwie `oraInst.loc`. Plik ten zawiera informacje o nazwie grupy inwentarza oraz o ścieżce dostępu do katalogu inwentarza. Aby sprawdzić, czy konto właściciela oprogramowania Oracle (`oracle`) istnieje, wystarczy wprowadzić instrukcję:

```
id oracle
```

Jeżeli konto istnieje, wynik wykonania polecenia jest zbliżony do przedstawionego poniżej.

```
uid=440(oracle) gid=200(oinstall) grupy=200(oinstall),201(dba),202(oper)
```

Jeśli konto istnieje, trzeba zdecydować, czy do instalacji zostanie użyte właśnie ono, czy powinno zostać utworzone nowe. W przypadku podjęcia decyzji o wykorzystaniu istniejącego konta należy sprawdzić, czy podstawową grupą użytkownika jest grupa inwentarza oraz czy dany użytkownik należy do grup opcjonalnych właściwych dla administratorów i operatorów baz danych.

Konto właściciela oprogramowania Oracle, grupy inwentarza, administratorów i operatorów muszą istnieć we wszystkich węzłach klastra i muszą mieć identyczne parametry. Aby utworzyć jednakowe konta użytkowników i grup, trzeba sprawdzić identyfikatory konta i grup w systemie, w którym zostały utworzone, a następnie utworzyć konta o takich samych identyfikatorach w pozostałych węzłach klastra.

Ustalenie identyfikatorów kont użytkowników i grup

Przedstawione poniżej polecenie pozwala na sprawdzenie identyfikatora właściciela oprogramowania Oracle (wartość `uid`) oraz identyfikatorów grup (wartości `gid`) inwentarza, administratorów i operatorów.

```
id oracle
```

Wynik wykonania polecenia jest zbliżony do następującego:

```
uid=440(oracle) gid=200(oinstall) grupy=200(oinstall),201(dba),202(oper)
```

Na podstawie listingu wynikowego można bez trudu ustalić, jaki jest identyfikator właściciela oprogramowania oraz jakie są identyfikatory grup, do których ten użytkownik należy.

Konfiguracja modułów (programów) systemu Linux

Przed zainstalowaniem oprogramowania klastra RAC konieczne jest właściwe skonfigurowanie usługi SSH we wszystkich węzłach klastra. Instalator Oracle wykorzystuje bowiem polecenia `ssh` i `scp` do zdalnego wykonywania instrukcji oraz kopiowania plików między węzłami klastra. Usługa SSH musi być więc odpowiednio skonfigurowana, aby na ekranie nie pojawiały się monity o wprowadzenie hasła.

Katalogi Oracle'a

Oprogramowanie Oracle wymaga utworzenia lub wskazania wielu katalogów. Są wśród nich:

- ♦ katalog inwentarza Oracle,
- ♦ bazowy katalog oprogramowania Oracle,
- ♦ katalog domowy usług CRS,
- ♦ katalog domowy użytkownika `oracle`.



Uwaga

Usługi CRS muszą dysponować innym katalogiem domowym niż `$ORACLE_HOME`.

W kolejnych punktach podrozdziału zostały opisane wymagane parametry tych katalogów.

Katalog inwentarza Oracle

W katalogu inwentarza Oracle znajduje się rejestr wszystkich elementów zainstalowanego oprogramowania Oracle. Ten obowiązkowy katalog jest współdzielony przez wszystkie instalacje oprogramowania w danym systemie. W czasie pierwszej instalacji program instalatora wymaga wskazania ścieżki do tego katalogu. Firma Oracle zaleca, aby ścieżka była określona jako:

```
$ORACLE_BASE/oraInventory
```

Katalogu tego nie trzeba przygotowywać osobiście. Instalator na podstawie podanej wartości ścieżki tworzy katalog, definiuje właściciela, grupę oraz prawa dostępu. Zawartość katalogu należy regularnie archiwizować. Nie wolno też go usuwać, chyba że usuwane jest całe oprogramowanie Oracle. Różne produkty firmy Oracle mogą dysponować różnymi katalogami *oraInventory*. Poszczególne wersje Uniwersalnego Instalatora Oracle (OUI) znane są z tego, że nie współdziałają ze sobą najlepiej. Dlatego różnym produktom Oracle'a (na przykład pakietom Database, Application Server czy Enterprise Manager Grid Control) mogą odpowiadać oddzielne pliki *oraInst.loc*. Trzeba więc pamiętać o wskazaniu w pliku *oraInst.loc* katalogu każdej nowej instalacji.

Katalog bazowy oprogramowania Oracle

Katalog bazowy oprogramowania Oracle jest katalogiem najwyższego poziomu dla instalowanych plików oprogramowania. Zazwyczaj jest to katalog */opt/oracle*. Jego właścicielem powinien być użytkownik, którego konto zostało wskazane jako konto właściciela całego oprogramowania Oracle.

Niezależnie od tego, czy zostanie utworzony nowy katalog bazowy, czy zostanie wykorzystany istniejący, pełną ścieżkę do tego katalogu trzeba zdefiniować w zmiennej środowiskowej `ORACLE_BASE`.



Uwaga

Katalog bazowy Oracle'a powinien być umieszczony w lokalnym systemie plików. Nie należy tworzyć go w systemie OCFS w wersji 1. Kolejne wersje systemu plików OCFS mogą poprawnie obsługiwać katalog bazowy, jednak w wersji 1 systemu OCFS nie jest to możliwe.

Wyznaczanie zmiennych środowiskowych

Po utworzeniu opisanych katalogów administrator musi zdefiniować następujące zmienne środowiskowe:

- ♦ `ORACLE_HOME`,
- ♦ `ORACLE_BASE`,
- ♦ `CRS_HOME`,
- ♦ `PATH`.

W przypadku pierwszej instalacji oprogramowania Oracle w węźle konieczne jest również uruchomienie skryptu *oraInstRoot.sh*, umieszczonego w katalogu */opt/oracle/oraInventory* (konieczność wykonania skryptu jest sygnalizowana przez instalator OUI).

Parametry pracy sieciowej

Kolejny etap wdrażania klastra RAC w systemie Linux polega na zdefiniowaniu parametrów pracy sieciowej. Związane z nim czynności muszą zostać wykonane przed próbą zainstalowania bazy danych Oracle 10g lub usług CRS. Niektóre adresy IP i nazwy domenowe są zapisywane

na stałe w plikach oprogramowania. Dlatego jakakolwiek pomyłka w ustawieniach sieciowych może oznaczać konieczność częściowego przeinstalowania modułów RAC. Pierwszym elementem, na który należy zwrócić uwagę, są adresy IP i nazwy domenowe.

Adresy IP i nazwy domenowe

Przed rozpoczęciem instalacji jakiegokolwiek części oprogramowania Oracle konieczne jest ustalenie lub uzyskanie następujących adresów IP dla każdego węzła:

- ♦ Jeden adres IP dla każdego interfejsu sieci publicznej oraz odpowiadająca mu zarejestrowana (w serwerze DNS) nazwa domenowa. Jeżeli każdy z węzłów dysponuje dwoma interfejsami sieci publicznej, trzeba zapewnić dla nich dwa adresy IP oraz dwie zarejestrowane nazwy domenowe.
- ♦ Jeden nieużywany wirtualny adres IP (VIP) i związana z nim wirtualna nazwa domenowa zarejestrowana w serwerze DNS. Kreator konfiguracji VIP (ang. *VIP Configuration Assistant — VIPCA*) przypisuje wirtualny adres IP podstawowemu interfejsowi sieci publicznej. Wirtualny adres IP musi należeć do tej samej podsieci, w której pracuje związany z nim interfejs publiczny. Po zakończeniu instalacji można zapisać wirtualne nazwy domenowe lub adresy IP w stacjach klienckich. W przypadku awarii węzła ruch zostanie przeniesiony do innego węzła na podstawie tego wirtualnego adresu IP.
- ♦ Prywatny adres IP oraz opcjonalnie nazwę domenową dla każdego interfejsu sieci prywatnej. Firma Oracle zaleca stosowanie w tym celu adresów z puli zarezerwowanej dla sieci LAN, na przykład 10.*.* lub 192.168.*.*. Do powiązania prywatnych adresów IP z odpowiadającymi im nazwami domenowymi można wykorzystać plik */etc/hosts* (w każdym węźle klastra).

Jeśli na przykład każdy węzeł dysponuje dwoma interfejsami publicznymi i dwoma prywatnymi, w jednym z węzłów (*rac1*) mogłyby zostać wykorzystane wymienione poniżej adresy IP i nazwy domenowe. Podobne adresy IP i nazwy domenowe powinny być wówczas zdefiniowane w pozostałych węzłach.

Nazwa domenowa	Rodzaj	Adres IP	Zarejestrowany w
<i>db01.domena.pl</i>	Publiczna	172.20.43.100	DNS
<i>db02.domena.pl</i>	Publiczna	172.20.43.102	DNS
<i>db01-vip.domena.pl</i>	Wirtualna	172.20.43.104	DNS
<i>db02-vip.domena.pl</i>	Wirtualna	172.20.43.106	DNS
<i>wezelrac1</i>	Prywatna	192.168.0.1	<i>/etc/hosts</i>
<i>wezelrac2</i>	Prywatna	192.168.0.2	<i>/etc/hosts</i>



Często wszystkie adresy IP i odpowiadające im nazwy domenowe są definiowane w pliku */etc/hosts*. W pliku tym powinna występować również definicja interfejsu pętli zwrotnej.

Weryfikacja ustawień pracy sieciowej

Aby upewnić się, że każdy z węzłów spełnia wymagania odnośnie parametrów pracy sieciowej, należy wykonać opisaną poniżej procedurę.

1. Jeśli jest to konieczne, zainstaluj interfejsy sieci publicznej i prywatnej, a następnie przypisz im odpowiednio publiczne i prywatne adresy IP.
2. Zarejestruj nazwy i adresy IP interfejsów sieci publicznej w serwerze DNS.
3. Zarejestruj w serwerze DNS jeden adres wirtualny i nazwę wirtualną dla każdego węzła.
4. Dla każdego interfejsu każdego węzła utwórz wpis w pliku `/etc/hosts`, uwzględniając w nim prywatny adres IP i prywatną nazwę:

```
192.168.0.1 wezelrac1
```

5. Zweryfikuj nazwy interfejsów i odpowiadające im adresy IP, wykorzystując do tego celu polecenie:

```
/sbin/ifconfig -a
```

Listing wyników ostatniej z instrukcji umożliwia sprawdzenie, czy adresy IP poszczególnych kart sieciowych odpowiadają tym adresom, które miały zostać wyznaczone jako publiczne i prywatne.

Kojarzenie (grupowanie) interfejsów publicznych i interfejsów połączeń wewnętrznych

Oprogramowanie Oracle 10g jest kupowane przede wszystkim ze względu na jego niezawodność. Warto więc poświęcić chwilę czasu na zapewnienie rzeczywistej niezawodności całego systemu. *Kojarzenie* (ang. *bonding*) jest terminem opisującym procedurę grupowania dwóch lub większej liczby interfejsów sieciowych. Dzięki niemu ruch IP może być przenoszony przez jeden z interfejsów lub przez obydwa. Jeżeli jeden z interfejsów ulegnie awarii, drugi przejmuje jego funkcję. Skojarzone interfejsy są osiągalne z zewnątrz pod jednym adresem IP.

Operacja kojarzenia interfejsów musi zostać wykonana poprawnie. Może być jednak przeprowadzona po skonfigurowaniu klastra RAC. Grupowanie interfejsów przed instalacją oprogramowania RAC nie jest konieczne.

Ustawienie wirtualnych adresów IP

Zgodnie z wcześniejszymi informacjami jednym z wymogów klastra RAC jest wyznaczenie wirtualnego adresu IP i zarejestrowanie dla niego nazwy domenowej w serwerze DNS. Wirtualny adres IP jest wykorzystywany przez oprogramowanie Oracle jedynie w czasie awaryjnego przełączania systemów. Adresy VIP są definiowane po zainstalowaniu usług CRS.

Adres VIP jest wykorzystywany przez oprogramowanie Oracle jako klucz do węzłów klastra. Wśród parametrów pracy sieciowej — zdefiniowanych w pliku `tnsnames.ora` — wymieniona jest nazwa lub adres IP serwera Oracle wyposażonego w proces nasłuchu sieciowego.

Jeśli adresy VIP są wykorzystywane w działaniu systemu, w pliku `tnsnames.ora` w miejscu rzeczywistej nazwy domenowej lub adresu IP jest zapisywana wirtualna nazwa jednostki lub wirtualny adres IP. Gdy praca serwera zostanie przerwana, wartość VIP zostaje przeniesiona do drugiego z węzłów klastra. Ruch ze wszystkich jednostek klienckich, które próbują uzyskać dostęp do instancji Oracle na podstawie adresu VIP, jest przekazywany do serwera określanego wartością VIP. Jest to jeden z najważniejszych parametrów klastra RAC 10g.

Testowanie pracy sieciowej

Przed zainstalowaniem oprogramowania Oracle zawsze należy sprawdzić poprawność działania sieci.

Wystarczy w tym celu ustanowić połączenia SSH z każdym węzłem klastra, posługując się kontem użytkownika root. W teście należy wykorzystać zarówno adresy IP, jak i nazwy domenowe.

Warto również sprawdzić, czy możliwe jest kopiowanie jakichkolwiek plików między jednostkami za pomocą usługi SCP. Instalator Oracle będzie wykorzystywał mechanizm SCP do przenoszenia plików binarnych między węzłami.



Uwaga

Jeśli ustawienia sieciowe nie zostaną poprawnie zdefiniowane, instalacja oprogramowania Oracle zakończy się niepowodzeniem.

Sprawdzenie dostępności współdzielonej pamięci masowej

Ostatnim elementem wymagającym sprawdzenia przed rozpoczęciem instalacji oprogramowania jest dostępność współdzielonej pamięci masowej. Jeśli część węzłów nie będzie mogła korzystać ze wspólnego obszaru składowania danych, konfiguracja systemu Oracle 10g nie będzie mogła się rozpocząć. Oto kroki stosownej procedury:

1. Sprawdź, czy dla kontrolera został zainstalowany odpowiedni sterownik:

- ♦ HBA,
- ♦ SCSI,
- ♦ RAID.

2. Zainstaluj oprogramowanie dla urządzeń pamięci masowej.

3. Sprawdź konfigurację pamięci masowej:

- ♦ czy grupy RAID zostały utworzone?
- ♦ czy dyski zostały podzielone na partycje?

4. Sprawdź, czy każdy z węzłów ma dostęp do grup RAID (LUN):

```
cat /proc/partitions
```

Konfiguracja klastra RAC

W tej części rozdziału zostaną omówione zagadnienia związane z konfiguracją klastra RAC Oracle 10g. Oprogramowanie klastra RAC składa się z kilku komponentów, które muszą być parametryzowane niezależnie od siebie. Konfiguracja poszczególnych z nich — usług CRS, połączeń wewnętrznych oraz współdzielonych dysków OCFS, ASM i dysków nieformatowanych — jest właśnie tematem niniejszego podrozdziału.

Poniżej zostało zamieszczonych kilka wskazówek, o których warto pamiętać podczas implementowania aplikacji na bazie klastra RAC.

- ♦ Pliki danych systemu RAC zapisuj w dowolnej współdzielonej pamięci masowej. Zalecam, aby w klastrze RAC był do tego celu wykorzystywany system plików klastra OCFS.
- ♦ Jeżeli jest to możliwe, stosuj sprzętową pamięć podręczną — o ile jest ona odporna na awarie i nie ma ryzyka utraty zbuforowanych danych.
- ♦ Nie twórz bazy danych podczas instalowania oprogramowania. Procedura instalacyjna powinna obejmować jedynie oprogramowanie.
- ♦ Do tworzenia bazy danych wykorzystuj kreator konfiguracji baz danych (ang. *Database Configuration Assistant* — *DBCA*). Aplikacja DBCA uwzględnia obsługę mechanizmów klastrów RAC.
- ♦ Usługi skonfigurowanego środowiska również powinny być definiowane za pomocą kreatora DBCA i zarządzane za pomocą aplikacji Server Control (`srvctl`) oraz Oracle Enterprise Manager (OEM).
- ♦ Korzystaj z pliku parametrów serwera (`SPFILE`), który powinien być zapisany w obszarze współdzielonej pamięci masowej. Plik może być zapisany w systemie OCFS lub na nieformatowanym dysku.
- ♦ Wykorzystuj mechanizm automatycznego zarządzania segmentami pamięci (ang. *Automatic Segment-Space Management*).
- ♦ Wykorzystuj mechanizm automatycznego zarządzania operacjami wycofania (ang. *Automatic Undo Management*).
- ♦ Dodawanie kolejnych węzłów do klastra wcale nie oznacza natychmiastowej poprawy wydajności. Jeżeli bieżące instancje Oracle nie są obciążone, rozbudowa klastra mija się z celem.

Konfiguracja usług CRS

Usługi CRS są usługami, które umożliwiają poszczególnym węzłom klastra komunikowanie się ze sobą. Mają więc kluczowe znaczenie dla funkcjonowania systemu RAC. Przed zapoznaniem się z zasadami konfigurowania usług CRS niezbędne jest zapamiętanie kilku istotnych faktów z nimi związanych:

- ♦ Usługi CRS muszą być zainstalowane i uruchomione przed rozpoczęciem instalacji klastra RAC 10g.
- ♦ Usługi CRS mogą działać bez zewnętrznego oprogramowania zarządzania klastrem. Platforma Oracle 10g pozwala również na uruchomienie usług CRS na bazie oprogramowania dostarczanego przez zewnętrznego producenta, na przykład Veritas Cluster.
- ♦ Pliki binarne serwera Oracle i usług CRS muszą być przechowywane w różnych katalogach — zmienne środowiskowe `ORACLE_HOME` i `CRS_HOME` muszą wskazywać różne katalogi.

- ♦ Przed zainstalowaniem usług CRS musi być dostępna współdzielona pamięć masowa przeznaczona dla pliku głosowania (ang. *voting file*) i pliku rejestru OCR (ang. *Oracle Cluster Registry*). Plik głosowania powinien mieć rozmiar co najmniej 20 MB, a plik OCR — co najmniej 100 MB. Obydwa pliki muszą być zapisane na niesformatowanych dyskach. System OCFS nie jest w tym przypadku obsługiwany.
- ♦ Publiczne i prywatne interfejsy węzłów klastra RAC muszą być skonfigurowane przed zainstalowaniem usług CRS.
- ♦ Aby zagwarantować nieprzerwaną pracę systemu, plik OCR i plik głosowania powinny być przechowywane w macierzy RAID.

Płytkę CD usług CRS można zamontować jako dysk lokalny pierwszego węzła klastra. Przed uruchomieniem instalatora trzeba się również upewnić, że dostępne są następujące informacje:

- ♦ położenie katalogów `ORACLE_HOME` i `CRS_HOME`,
- ♦ lokalizacja współdzielonych dysków dla plików głosowania i OCR.

Aby skonfigurować usługi CRS, należy wykonać polecenie `runInstaller` w ramach sesji X-Window pierwszego węzła klastra (bieżącym katalogiem powinien być katalog płyty CD instalatora CRS — jest to oddzielna płyta w pakiecie Oracle 10g). Wprowadzenie instrukcji spowoduje uruchomienie instalatora. Usługi CRS pracują z prawami użytkownika `root`, natomiast komponenty klastra RAC 10g są uruchamiane w ramach konta `oracle`.

Publiczne adresy IP i wirtualne adresy IP poszczególnych węzłów jednego klastra muszą należeć do jednej podsięci.

Podczas instalacji aplikacja OUI w pierwszej kolejności kopiuje oprogramowanie do lokalnego systemu, a następnie przekazuje je do węzłów zdalnych. Po zakończeniu procedury instalacyjnej można sprawdzić jej poprawność przez wykonanie instrukcji `olsnodes` w katalogu `CRS_HOME/bin`. Składnia polecenia `olsnodes` jest następująca:

```
olsnodes [-n] [-l] [-v] [-g]
```

gdzie:

- ♦ opcja `-n` odpowiada za wyświetlenie numeru węzła i jego nazwy,
- ♦ opcja `-l` powoduje wyświetlenie lokalnej nazwy węzła,
- ♦ opcja `-v` uaktywnia tryb wyświetlania szczegółowych informacji wynikowych,
- ♦ opcja `-g` aktywuje rejestrację zdarzeń w dziennikach.

Oto przykład zastosowania instrukcji:

```
olsnodes -n
```

Wynikiem wykonania polecenia powinna być lista węzłów, w których usługi CRS zostały zainstalowane.

```
Node1 1  
Node2 2
```

Konfiguracja połączenia wewnętrznego

Połączenie wewnętrzne jest jednym z najważniejszych elementów systemu Oracle RAC. Firma Oracle zawarła w wymaganiach informację o tym, że musi to być połączenie o przepustowości minimalnej 100 Mb/s. Z doświadczeń praktycznych wynika jednak, że zastosowanie gigabitowego Ethernetu jest nieodzowne. Niska przepływność połączenia wewnętrznego może być przyczyną wielu problemów. Jeżeli stopień wykorzystania aplikacji jest względnie duży, a przepustowość połączenia niska, baza danych może zgłaszać błędy.

Połączenie wewnętrzne jest w zasadzie siecią prywatną, która powinna być oddzielona od innych segmentów sieci. Na jej obsługę warto przeznaczyć oddzielny przełącznik lub przynajmniej niezależną sieć wirtualną (VLAN).

Aby zwiększyć wydajność pracy klastra, można ustawić rozmiary nadawanych i odbieranych pakietów UDP na 256 KB.

Konfiguracja współdzielonych dysków

Współdzielone dyski mogą być konfigurowane na kilka różnych sposobów. Najczęściej wykorzystywane są rozwiązania bazujące na systemie plików klastra Oracle (OCFS), na nieformatowanych urządzeniach lub na mechanizmie ASM.

Konfiguracja współdzielonych dysków systemu OCFS

System plików klastra Oracle (ang. *Oracle Cluster File System* — *OCFS*) jest współdzielonym systemem plików wykorzystywanym w pracy klastra RAC. Choć nazwa sugeruje, że jest to pełny system plików, w praktyce umożliwia jedynie przechowywanie danych Oracle'a i plików archiwum dziennika. System OCFS jest zbliżony w działaniu do menedżera wolumenów logicznych (LVM). Składa się z dwóch komponentów — napędu dyskowego i usługi. Napęd dyskowy OCFS odpowiada za współpracę z systemem plików systemu operacyjnego. Usługa *cfs* koordynuje komunikację między poszczególnymi węzłami klastra. Odpowiada za utrzymanie synchronizacji węzłów i ich plików. Prawidłowe działanie systemu OCFS zależy w głównej mierze od dużej przepustowości połączeń wewnętrznych.

Konfiguracja współdzielonych nieformatowanych dysków

Oprogramowanie RAC dla większości platform wymaga udostępnienia każdej instancji zbioru niesformatowanych dysków we współdzielonym podsystemie dyskowym. Dyski te nazywa się również *czystymi dyskami* (ang. *raw devices*). Pracujące w klastrze RAC instancje Oracle wykorzystują niesformatowane dyski do przechowywania plików sterujących, plików parametrów serwera, plików danych oraz bieżących dzienników powtórzeń. Wymienione pliki są wspólne dla wszystkich instancji. Same dyski muszą zostać przygotowane za pomocą odpowiednich narzędzi systemu operacyjnego przed utworzeniem bazy danych.

Czysty *dysk* jest napędem, w którym nie został założony żaden system plików. Dyski tego typu są wykorzystywane w klastrach RAC, ponieważ umożliwiają współdzielenie pamięci masowej.

Czysta *partycja* jest fizycznym fragmentem dysku, którym można operować na najniższym z dozwolonych poziomów. Czysta partycja zostaje powołana w chwili, gdy utworzona zostaje partycja rozszerzona i przypisywane są do niej partycje logiczne (bez formatowania). Po sformatowaniu jest ona nazywana partycją *przygotowaną* (ang. *cooked*).

Nieformatowane dyski mają kilka zalet w porównaniu z systemem OCFS. Przede wszystkim mechanizm OCFS jest systemem plików, a systemy plików wnoszą pewien narzut, buforują dane, są trudne do wykorzystania w klastrze i przy przechowywaniu danych różnych systemów operacyjnych. W przypadku czystych dysków nie występuje synchronizacja na poziomie systemu operacyjnego. Oprogramowanie Oracle używa nieformatowanych dysków do przechowywania danych oraz do ustanawiania blokad danych. Zapewnienie funkcji synchronizacji danych w systemie OCFS jest przyczyną dodatkowego narzutu. Dlatego czyste dyski są od 3 do 7% efektywniejsze. Z drugiej strony system OCFS gwarantuje:

- ♦ uproszczenie zarządzania,
- ♦ wykorzystanie funkcji Oracle Managed Files w klastrze RAC,
- ♦ jednorazową instalację oprogramowania Oracle,
- ♦ automatyczne rozszerzenie plików danych Oracle'a,
- ♦ jednolity dostęp do archiwów dziennika w przypadku fizycznego uszkodzenia węzła.

Firma Oracle intensywnie rozbudowuje system OCFS przy jednoczesnym zaniechaniu jakichkolwiek innowacyjnych działań w odniesieniu do nieformatowanych dysków. Z tego względu stosowanie czystych dysków jest zalecane jedynie w przypadkach, w których konieczne jest maksymalne zwiększenie wydajności.

Konfiguracja współdzielonych dysków z wykorzystaniem mechanizmu ASM

Trzeci sposób konfiguracji współdzielonych dysków bazuje na wykorzystaniu mechanizmu automatycznego zarządzania pamięcią masową (ang. *Automatic Storage Management — ASM*). Funkcja ASM jest nowym rozwiązaniem firmy Oracle, które zwalnia administratora z obowiązku wykonywania szeregu żmudnych czynności. Oto zasada jego działania.

Dzięki systemowi ASM administrator nie musi zarządzać poszczególnymi plikami i napędami dyskowymi. Można bowiem tworzyć grupy dyskowe składające się z dysków oraz ze związanych z nimi plików. Połączenie technologii OMF i ASM w zasadzie zwalnia administratora z obowiązku definiowania nazw i lokalizacji plików baz danych zarówno podczas ich tworzenia, jak i w czasie wykonywania innych operacji bazodanowych. Wystarczy wskazać docelową grupę dyskową, a resztą zajmie się oprogramowanie Oracle. Mechanizm ASM może objąć swoim działaniem również inne zadania administracyjne, w tym zarządzanie dyskami oraz sporządzenie kopii zapasowych i odtwarzanie danych z archiwów.

Przed otwarciem bazy danych fizyczne pliki są zlokalizowane na podstawie informacji zapisanych w pliku sterującym. Jednak plik sterujący sam jest elementem grupy dyskowej. Proces ASM musi więc dysponować własną niezależną instancją, która jest powoływana przed uruchomieniem instancji bazy danych.

Jedną z operacji wykonywanych podczas uruchamiania instancji ASM jest operacja identyfikacji grup dyskowych i ich plików. System ASM montuje dyski, tworzy mapę ciągłych bloków pamięci dyskowej przydzielonych segmentowi bazy danych (ang. *extent*) i przekazuje ją do instancji bazy danych. Za realizowanie operacji wejścia-wyjścia jest już odpowiedzialna sama baza danych. Współdziałanie procesu ASM jest niezbędne jedynie w przypadku tworzenia lub usuwania plików oraz podczas zmiany konfiguracji dysku.

Gdy zachodzą jakiegokolwiek zmiany, instancja ASM równoważy obciążenie dysków i dostarcza informacji niezbędnych do odświeżenia mapy bloków w pamięci SGA instancji bazy danych. Z tego względu proces ASM i instancja bazy danych muszą pracować jednocześnie. Wyłączenie instancji ASM powinno nastąpić dopiero po zamknięciu bazy danych.

Wpływ mechanizmu ASM na wydajność pracy instancji bazy danych jest niewielki. Nie bierze on bowiem udziału w przetwarzaniu transakcji odnoszących się do poszczególnych obiektów bazodanowych. Dlatego średni rozmiar obszaru SGA przeznaczony na obsługę mechanizmu ASM nie przekracza 64 MB. Jeśli więc rozmiar pamięci serwera nie jest zbliżony do maksymalnego rozmiaru pamięci operacyjnej wymienionej w wymaganiach DBMS, zajętość 64 MB nie powinna mieć wpływu na dostępność pamięci dla instancji bazy danych.

Z uwagi na fakt, że mechanizm ASM sam zarządza grupami dyskowymi i plikami bazy danych, zadanie administratora sprowadza się jedynie do wskazania odpowiednich dysków, określenia grup awaryjnych oraz poziomu nadmiarowości zapisu danych. System ASM automatycznie rozkłada obciążenie na wszystkie dyski grupy. Dzięki temu rywalizacja o dostęp do dysku jest znacznie ograniczona, co skutkuje zwiększeniem wydajności pracy. Dodatkową zaletą mechanizmu ASM jest możliwość automatycznego i dynamicznego rozmieszczania plików w przypadku zmiany rozmiaru grupy dyskowej (dodania lub usunięcia dysku). Zadanie to jest realizowane bez potrzeby zamykania bazy danych.

W systemie ASM rozmiar grupy dyskowej może zostać określony podczas tworzenia nowego pliku danych. Na przykład aby utworzyć przestrzeń tabel z jednoczesną alokacją obszaru o rozmiarze 500 MB, wystarczy wykonać polecenie:

```
CREATE TABLESPACE przestrzentabel1 DATAFILE '+grupadanych1' SIZE 500MB;
```

Mechanizm ASM odpowiada za nazewnictwo, rozmieszczenie i dystrybucję plików danych. Administrator musi jedynie wskazać grupę dyskową, które będzie przechowywała plik danych. Rozwiązanie to eliminuje konieczność osobistego optymalizowania operacji wejścia-wyjścia i usuwania problemów związanych z fragmentacją danych.



Uwaga

Mechanizm ASM, podobnie jak OCFS, nie jest systemem plików ogólnego przeznaczenia. Nie może zastępować systemu plików Linuksa. Nie powinien być więc stosowany do zarządzania plikami, które są wykorzystywane przez funkcje śledzenia procesów drugoplanowych lub procesów użytkownika.

Największą zaletą systemu ASM jest jego zdolność do dynamicznej modyfikacji zasobów pamięci masowej, eliminacja ręcznej optymalizacji procedur wejścia-wyjścia oraz automatyczne rozkładanie obciążenia. Przy kilku tysiącach plików zadania wykonywane automatycznie przez system ASM mogą zaoszczędzić administratorowi od 30 do 40% czasu pracy. Zastosowanie tego rozwiązania pozwala przeznaczyć cenny czas na inne cele.

Testowanie klastra RAC i zarządzanie nim

Do najważniejszych czynności wykonywanych po zakończeniu instalacji jest testowanie klastra RAC i zarządzanie jego pracą. Zagadnienia opisywane w tym punkcie są odpowiedziami na pytania, dlaczego i w jaki sposób należy testować daną implementację klastra RAC systemu Oracle? Część testów wiąże się jednak z uruchamianiem i zatrzymywaniem instancji. Zatem w pierwszej kolejności zostaną omówione problemy zarządzania klastrem RAC.

Większość zadań związanych z zarządzaniem można wykonać za pomocą jednego narzędzia — `srvctl`. Program `Srvctl` jest interfejsem wiersza poleceń instancji i usług RAC.

Załóżmy, że została utworzona dwuwęzłowa baza danych RAC o nazwie `seed`. Bazie tej odpowiadają dwie instancje — `seed1` i `seed2`. Instancja `seed1` pracuje w węźle `rac1`, natomiast `seed2` — w węźle `rac2`. Aby poznać status bazy danych `seed`, wystarczy wykonać polecenie:

```
srvctl status database -d seed
```

Wynik powinien być następujący:

```
Instance seed1 running on node rac1
Instance seed2 running on node rac2
```

Polecenie `srvctl` może być wprowadzane w dowolnym węźle klastra.

Poniżej zostały przedstawione przykłady składni polecenia. Jego zastosowanie pozwala na sprawdzenie statusu usługi, aplikacji węzła, bazy danych lub samej instancji.

```
srvctl status database -d <nazwa_bazy_danych> [-f] [-v] [-S <poziom>]
srvctl status instance -d <nazwa_bazy_danych> -i <nazwa_instancji>
    [,<lista_nazw_instancji>] [-f] [-v] [-S <poziom>]
srvctl status service -d <nazwa_bazy_danych> -s <nazwa_uslugi> [,<lista_nazw_uslugi>] [-f]
    [-v] [-S <poziom>]
srvctl status nodeapps [-n <nazwa_wezla>]
srvctl status asm -n <nazwa_wezla>
```

Oto kilka przykładów wykorzystania instrukcji:

- ♦ sprawdzenie statusu bazy danych, wszystkich instancji i wszystkich usług:

```
srvctl status database -d seed -v
```

- ♦ sprawdzenie statusu wybranych instancji oraz ich usług:

```
srvctl status instance -d seed -i seed1,seed2 -v
```

- ♦ sprawdzenie statusu określonej usługi:

```
srvctl status service -d seed -s CRM -v
```

- ♦ sprawdzenie statusu wszystkich węzłów obsługujących aplikacje bazodanowe:

```
srvctl status node
```

Polecenie `srvctl` umożliwia również uruchamianie i zatrzymywanie baz danych oraz ich instancji. Składnia tego typu instrukcji została przedstawiona poniżej. Należy pamiętać, że podczas uruchamiania bazy danych wszystkie związane z nią instancje również są uruchamiane.

```

srvctl start database -d <nazwa_bazy_danych> [-o <opcje_startowe>] [-c <ciąg_połączenia>
| -q]
srvctl start instance -d <nazwa_bazy_danych> -i <nazwa_instancji>
[.<lista_nazw_instancji>] [-o <opcje_startowe>] [-c <ciąg_połączenia> | -q]

```

Za pomocą instrukcji `srvctl` można również uruchamiać usługi klastra, aplikacje węzła oraz system ASM.

```

srvctl start service -d <nazwa_bazy_danych> [-s <nazwa_usługi> [.<lista_nazw_usługi>]]
-i <nazwa_instancji> [.<lista_nazw_instancji>] [-o <opcje_startowe>]
[-c <ciąg_połączenia> | -q]
srvctl start nodeapps -n <nazwa_węzła>
srvctl start asm -n <nazwa_węzła> [-i <nazwa_instancji_asm>] [-o <opcje_startowe>]

```

Polecenie `srvctl` umożliwia także zatrzymywanie procesów CRS. Składnia różnych wersji instrukcji została przedstawiona poniżej.

```

srvctl stop database -d <nazwa_bazy_danych> [-o <opcje_wyłączenia>] [-c <ciąg_połączenia>
| -q]
srvctl stop instance -d <nazwa_bazy_danych> -i <nazwa_instancji> [.<lista_nazw_instancji>]
[-o <opcje_wyłączenia>] [-c <ciąg_połączenia> | -q]
srvctl stop service -d <nazwa_bazy_danych> -s <nazwa_usługi> [.<lista_nazw_usługi>]
[-o <opcje_wyłączenia>] [-c <ciąg_połączenia> | -q] [-f]
srvctl stop nodeapps -n <nazwa_węzła>
srvctl stop asm -n <nazwa_węzła> [-i <nazwa_instancji_asm>] [-o <opcje_wyłączenia>]

```

Oto kilka przykładów zastosowania polecenia:

- ♦ zatrzymanie bazy danych, wszystkich instancji i wszystkich usług:

```
srvctl stop database -d seed
```

- ♦ zatrzymanie określonych instancji:

```
srvctl stop instance -d seed -i seed1,seed2
```

- ♦ zatrzymanie usługi:

```
srvctl stop service -d seed -s CRM
```

- ♦ zatrzymanie usługi i określonej instancji:

```
srvctl stop database -d seed -s CRM -i seed2
```

- ♦ zatrzymanie aplikacji węzła:

```
srvctl stop nodeapps -n rac1
```



Zakończenie pracy aplikacji węzła powoduje jednoczesne zatrzymanie instancji i usług.

Kolejnymi prezentowanymi poleceniami są instrukcje modyfikacji sposobu działania usług CRS.

```

srvctl modify database -d <nazwa> [-n <nazwa_bazy_danych>] [-o <kat_oracle>] [-m <domena>]
[-p <plik_spfile>] [-r {PRIMARY | PHYSICAL_STANDBY | LOGICAL_STANDBY}]
[-s <opcje_startowe>]
srvctl modify instance -d <nazwa_bazy_danych> -i <nazwa_instancji> -n <nazwa_węzła>
srvctl modify instance -d <nazwa> -i <nazwa_instancji> {-s <nazwa_instancji_asm> | -r }
srvctl modify service -d <nazwa_bazy_danych> -s <nazwa_usługi> -i <nazwa_instancji>
-t <nazwa_instancji> [-f]

```



```

srvctl modify service -d <nazwa_bazy_danych> -s <nazwa_uslugi> -i <nazwa_instancji>
-r [-f]
srvctl modify nodeapps -n <nazwa_wezla> [-A <adres>] [-x]

```

Znaczenie poszczególnych opcji zostało opisane poniżej.

- ♦ -i <nazwa_instancji> -t <nazwa_instancji> — nazwa instancji [-i] jest zastępowana nazwą instancji [-t],
- ♦ -i <nazwa_instancji> -r — określona instancja jest wyznaczana jako instancja preferowana,
- ♦ -A lista_adresów — lista adresów aplikacji VIP na poziomie węzła,
- ♦ -s <nazwa_instancji_asm> — opcja wykorzystywana do dodawania lub usuwania zależności ASM.

Oto kilka przykładów wykorzystania instrukcji `srvctl` do zmiany sposobu pracy usług CRS:

- ♦ zmiana węzła, w którym uruchomiona jest instancja:

```
srvctl modify instance -d seed -n rac2
```

- ♦ zmiana węzła, w którym uruchomiony jest serwis:

```
srvctl modify service -d seed -s Hot_Backup -i rac2 -t rac3
```

- ♦ zmiana preferowanej instancji dla usługi:

```
srvctl modify service -d seed -s Hot_Backup -i rac2
```

Polecenia przedstawione poniżej włączają wykorzystanie zasobów CRS. Aby instancja mogła zostać uruchomiona, musi być wcześniej włączona. Funkcja może zostać wykorzystana, jeśli zasób jest włączony lub wyłączony.

```

srvctl enable database -d <nazwa_bazy_danych>
srvctl enable instance -d <nazwa_bazy_danych> -i <nazwa_instancji>
[.<lista_nazw_instancji>]
srvctl enable service -d <nazwa_bazy_danych> -s <nazwa_uslugi> [.<lista_nazw_uslug>]
[-i <nazwa_instancji>]

```

Oto kilka przykładowych instrukcji włączenia zasobów CRS:

- ♦ włączenie bazy danych:

```
srvctl enable database -d seed
```

- ♦ włączenie określonych instancji:

```
srvctl enable instance -d seed -i seed1,seed2
```

- ♦ włączenie usług:

```
srvctl enable service -d seed -s CRM,ERP
```

- ♦ włączenie usługi określonej instancji:

```
srvctl enable service -d seed -s ERP -i rac3
```

Wyłączanie zasobów CRS należy do zadań instrukcji prezentowanych poniżej. Funkcja może zostać wykorzystana, jeśli zasób jest włączony.

```

srvctl disable database -d <nazwa_bazy_danych>
srvctl disable instance -d <nazwa_bazy_danych> -i <nazwa_instancji>
[,<lista_nazw_instancji>]
srvctl disable service -d <nazwa_bazy_danych> -s <nazwa_uslugi> [,<lista_nazw_uslug>]
[-i <nazwa_instancji>]

```

Oto stosowne przykłady:

- ♦ globalne wyłączenie bazy danych:

```
srvctl disable database -d seed
```

- ♦ wyłączenie określonych instancji:

```
srvctl disable instance -d seed -i rac1,rac2
```

- ♦ globalne wyłączenie usługi:

```
srvctl disable service -d seed -s ERP
```

- ♦ wyłączenie usługi określonej instancji:

```
srvctl disable service -d seed -s CRM -i rac3,rac2
```

Czas wrócić do testów. Po wdrożeniu klastra Oracle RAC należy dokładnie przetestować jego działanie. Warto wykorzystać do tego celu procedury opisane w dalszej części podrozdziału.

Testy narzędzia srvctl

Test zawsze powinien obejmować operację pobrania informacji statusowych na temat baz danych wszystkich węzłów klastra. Administrator nie powinien mieć również problemu z uruchamianiem i zamykaniem baz danych w poszczególnych węzłach. Warto także przetestować możliwość uruchamiania instancji oraz usług (jeśli takowe są wykorzystywane) ze wszystkich węzłów.

Test awaryjnego przełączenia

Zakres procedury testowej w przypadku awaryjnego przełączania bazy danych (ang. *failover*) zależy w praktyce od rodzaju mechanizmu zabezpieczającego, który jest wykorzystany w danym rozwiązaniu. Jeśli na przykład jest stosowana funkcja Transparent Application Failover, niezbędne jest przeprowadzenie wielu testów z awaryjnym przełączaniem transakcji.

Podstawowy test, jaki można wykonać, polega na użyciu konsoli SQL*Plus do połączenia się z globalną instancją ze wszystkich węzłów, ustanowieniu połączenia z określoną instancją (na przykład seed1) ze wszystkich węzłów oraz zestawieniu połączenia z jednostki znajdującej się poza klastrem.

Kolejny opisywany test umożliwia szybkie sprawdzenie, czy plik *tnsnames.ora* został poprawnie skonfigurowany i czy gwarantuje awaryjne przełączenie baz danych. Pierwsza czynność polega na ustanowieniu połączenia z instancją za pomocą konsoli SQL*Plus (polecenie `sqlplus system/haslo@seed`). Dzięki poniższej instrukcji można zobaczyć, z którą instancją zostało ustanowione połączenie:

```
select instance_number, instance_name, host_name from v$instance;
```

Wynik powinien być zbliżony do następującego:

```
INSTANCE_NUMBER  INSTANCE_NAME  HOST_NAME
                1    seed1          rac1
```

Teraz wystarczy zakończyć działanie instancji wymienionej w listingu wynikowym i ponownie wykonać wcześniejszą instrukcję. Wynik powinien informować, że połączenie jest ustanowione z innym węzłem klastra.

```
INSTANCE_NUMBER  INSTANCE_NAME  HOST_NAME
                2    seed2          rac2
```

Jeżeli przełączenie nie spowoduje wyświetlenia jakiegokolwiek komunikatu o błędzie w konsoli SQL*Plus, można założyć, że ustawienia pliku *tnsnames.ora* są poprawne.

Można też przeprowadzić testy, które obejmowałyby takie czynności, jak w restart komputera w czasie połączenia z węzłem lub wykonanie instrukcji shutdown z opcjami *immediate*, *transactional* albo *abort* podczas połączenia z instancją. Wyniki poszczególnych operacji są zależne od rodzaju wykorzystanego polecenia SQL oraz od opcji awaryjnego przełączania zdefiniowanych w ciągu połączenia. Najważniejsze jest jednak to, aby pamiętać o przetestowaniu konfiguracji RAC 10g przed zastosowaniem jej w systemie użytkowym!

Dokumenty pomocy dla Oracle 10g RAC

Liczba przedstawionych dotychczas zagadnień jest dostatecznie duża, by utrudnić życie administratorowi. Ten rozdział nie miał służyć jako instrukcja instalacji oprogramowania RAC, lecz jako ogólny przewodnik, którego celem jest wskazanie kierunków wdrażania systemu Oracle 10g RAC.

W przypadku wystąpienia jakichkolwiek trudności poszukiwanie informacji należy rozpocząć od serwisu MetaLink. Firma Oracle zazwyczaj udostępnia w nim szczegółowe instrukcje postępowania przeznaczone dla każdej obsługiwanej platformy systemowej. Można również wykorzystać wyszukiwarkę, wprowadzając w niej słowo kluczowe *Linux* i numer wersji instalowanego oprogramowania. Kilka odsyłaczy na pewno zostanie zwróconych. W większości przypadków ktoś inny wcześniej zmagał się z danym problemem i go rozwiązał.

Oto kilka zasad, których należy przestrzegać przed rozpoczęciem instalacji oprogramowania RAC 10g:

- ♦ Zawsze sprawdzaj, czy dostawca sprzętu udostępnił certyfikowane (testowane) wersje sterowników lub oprogramowania.
- ♦ Upewnij się, że parametry pracy sieciowej systemu Linux zostały wyznaczone poprawnie. Błąd w konfiguracji sieci może być przyczyną wielu problemów.
- ♦ Zainstalowanie i uruchomienie usług CRS we wszystkich węzłach jest połową sukcesu. Nie należy kontynuować instalacji, dopóki usługi CRS nie pracują poprawnie.

- ♦ Upewnij się, że rozmiar współdzielonej pamięci masowej został właściwie dobrany. Większość nieefektywnie działających implementacji klastra RAC nie dysponuje dostatecznie dużą liczbą fizycznych napędów dyskowych lub bazuje na niewłaściwej wersji macierzy RAID. Zalecaną wersją macierzy dla plików danych Oracle jest RAID 1+0.
- ♦ Pamiętaj, by dobrze się bawić. Większość problemów wynika z błędów popełnianych w stresie. Zachowanie dobrego humoru w czasie pracy ułatwia całą procedurę.

Podsumowanie

Poprzedni rozdział miał charakter przeglądu funkcji RAC i zasad ich wykorzystywania. W tym rozdziale z kolei zostały przedstawione szczegółowe informacje na temat konfiguracji i instalacji oprogramowania RAC. Klaster Oracle RAC jest bardzo skomplikowanym produktem, którego praca w znacznie większym stopniu zależy od warstwy sprzętowej niż w przypadku niezależnego serwera Oracle. Wdrażanie klastra RAC zazwyczaj oznacza przeznaczenie około 40% czasu na instalację i 60% czasu na debugowanie. Przyjęcie takiego spojrzenia na procedurę implementacji pozwala wyeliminować stres pojawiający się, gdy coś nie do końca jest realizowane zgodnie z planem. Nie należy się wówczas poddawać, trzeba pamiętać o zwracaniu bacznej uwagi na najmniejsze szczegóły.