

XIV Konferencja PLOUG
Szczyrk
Październik 2008

Data Guard w praktyce

Jacek Sapiński
OPITZ CONSULTING Kraków. Sp. z o.o.

e-mail: jacek.sapinski@opitz-consulting.pl

Abstrakt. Ostatnie lata to narastająca informatyzacja przedsiębiorstw i organizacji. Naturalną konsekwencją tego procesu jest uzależnienie się przedsiębiorstw od systemów komputerowych. Jeszcze kilka lat temu, zagadnienie wysokiej dostępności systemów komputerowych wzbudzało zainteresowanie tylko nielicznych firm. Obecnie jest to kluczowy temat praktycznie w każdym przedsiębiorstwie. Dlatego też wszystkie wiodące firmy informatyczne skupiają się na poprawie „dostępności” swoich rozwiązań. Nie inaczej postępuje firma Oracle implementując coraz to nowe opcje w swoim oprogramowaniu. Należą do nich RMAN, Flashback, RAC i Data Guard. Technologia Data Guard pozwala na stworzenie i zarządzanie identyczną kopią bazy danych, znajdującą się na oddzielnym serwerze. Taka baza danych jest często określana mianem bazy Standby. Data Guard dba o to, by baza standby była na bieżąco aktualizowana poprzez przesył zmian z bazy produkcyjnej. W przypadku awarii serwera produkcyjnego Data Guard pozwala na aktywację bazy standby oraz przejście przez nią obsługi użytkowników, którzy wcześniej pracowali na bazie produkcyjnej. Niniejszy wykład omówi aspekty związane z bieżącym utrzymaniem Data Guard tak, by bazy produkcyjna i zapasowa pozostawały zsynchronizowane.

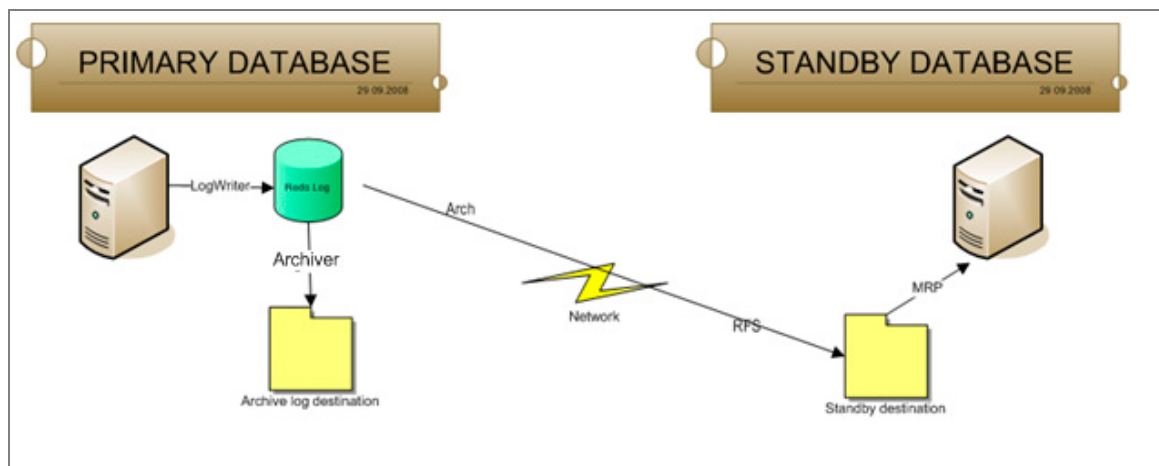
Informacja o autorze. Jacek Sapiński – Service Manager i Senior DBA w OPITZ CONSULTING Kraków sp. z o.o., Kieruje grupą Service Operations odpowiedzialną za monitorowanie i utrzymanie systemów. Posiada wieloletnie doświadczenie w administracji krytycznych systemów baz danych w jednym z największych cywilnych centrów przetwarzania danych w Europie.

1. DataGuard – Normalna praca

W tej prezentacji zakładam, że Data Guard został już skonfigurowany i działa poprawnie. Baza danych „standby” jest już stworzona, podjęto już również wszystkie decyzje co do trybu pracy oraz metody transportu logów. Przykładowa konfiguracja może wyglądać następująco:

- Physical Standby
- Tryb pracy Maximum performance,
- Logi są transportowane przez proces archivera,
- Brak Standby Redo Logs.

Oto graficzne przedstawienie tej konfiguracji:



Rys. 1. Podstawowa konfiguracja Data Guard

Baza Standby jest dokładną kopią bazy produkcyjnej. Baza standby jest na bieżąco aktualizowana poprzez aplikację informacji REDO otrzymywanych z bazy produkcyjnej. Jak dokładnie funkcjonuje ten proces w naszej prostej konfiguracji?

Proces Log Writer zapisuje zmiany do pliku dziennika powtórzeń. Po zapełnieniu takiego pliku proces archivera archiwizuje go we Flash Recovery Area. Jednocześnie kolejny proces archiver przesyła ten plik powtórzeń do bazy standby. Na bazie zapasowej plik ten jest odbierany przez proces RFS (Remote File Server). Proces ten jest automatycznie uruchamiany, gdy archiver łączy się po raz pierwszy z bazą standby. Nie wymaga on żadnej dodatkowej konfiguracji. Proces RFS odbiera plik przesyłany z bazy produkcyjnej i umieszcza go w katalogu określonym w parametrze standby_archive_dest. Tak powstały plik jest następnie pobierany przez proces MRP (Managed Recovery Process) i integrowany (wgrywany) do bazy danych. Proces MRP nie jest uruchamiany automatycznie. Aby go wystartować należy wydać następujące polecenie:

```
Alter Database recover manager standby Database disconnect from session;
```

Co dzieje się z przesłanymi logami, gdy nie działa proces MRP? Pozostają one w katalogu standby_archive_dest. Brak procesu MRP nie ma wpływu na przesył logów z bazy produkcyjnej. Są one nadal przesyłane i odbierane przez proces RFS.

2. Gaps

Co to jest Gap?

Gap nie jest niczym innym jak przerwą w sekwencji logów dziennika powtórzeń przesłanych do bazy standby. Logi są przesyłane z produkcji w kolejności ich tworzenia, więc taka przerwa w sekwencji nie powinna powstać. W praktyce jednak często mamy do czynienia z taką sytuacją. Spowodowane może to być na przykład wyłączeniem bazy danych standby lub problemami z siecią. Takie zdarzenia powodują naturalnie, że logi nie są przesyłane z zachowaniem kolejności chronologicznej.

Wg dokumentacji Oracle, postęp procesu aplikowania logów można odnaleźć w widoku v\$logarchived_log. W praktyce wydaje się w tym celu następujące zapytanie:

```
select sequence#,archived,applied from v$logarchived_log where dest_id=2 order by sequence#;
```

```
SEQUENCE# ARC APP
-----
451 YES YES
452 YES YES
453 YES YES
454 YES YES
455 YES YES
458 YES NO
459 YES YES
460 YES YES
```

Wynik tego zapytania może sugerować, że jeden z logów (seq. 197) nie został zaaplikowany na bazie standby. Czy jest to „Gap”?

Jak się okazuje nie jest to brak logów na bazie standby. To zapytanie zostało wykonane na bazie produkcyjnej. Zdarza się, że na bazie produkcyjnej pojawiają się takie „luki” w sekwencji logów. Zwykle pojawia się to po zlikwidowaniu prawdziwej przerwy w sekwencji logów. Informacja o logach, które zostały przesłane później, czasami nie jest poprawnie przesyłana do bazy primary. Dlatego też powyższego sprawdzenia należy zawsze dokonywać na bazie standby:

```
select sequence#,archived,applied from v$logarchived_log where dest_id=2 order by sequence#;
```

```
SEQUENCE# ARC APP
-----
451 YES YES
452 YES YES
453 YES YES
454 YES YES
455 YES YES
458 YES YES
459 YES YES
460 YES YES
```

W jaki sposób rozwiązywany jest problem braku niektórych logów w bazie standby? W większości przypadków dzieje się to automatycznie. Jednakże okazjonalnie wymagana jest manualna interwencja administratora.

2.1. Automatic Gap Resolution

Co to jest “Automatic Gap Resolution”?

Jest to zautomatyzowany proces uzupełniania brakujących logów dziennika powtórzeń na bazie standby. Proces ten może zostać zainicjowany zarówno przez proces RFS bazy standby, jak również przez proces Archivera bazy produkcyjnej.

Przeanalizujmy najpierw przypadek, kiedy inicjacji dokonuje proces RFS. Przy odbiorze pierwszego logu, po przerwie w przesyłce z bazy produkcyjnej, zostaje zauważony brak jednego lub więcej logów. Proces RFS wysyła do bazy produkcyjnej żądanie przesłania brakujących plików.

Drugim mechanizmem rozwiązywania tego problemu kontrolowany jest przez proces archivera. Jeden z procesów archivera (heartbeat archiver) ciągle odpytuje bazę standby i przesyła brakujące logi dziennika powtórzeń.

Te mechanizmy funkcjonują nie wymagają żadnej konfiguracji i w większości przypadków funkcjonują niezawodnie.

W tym kontekście warto też zwrócić uwagę na pojawiające się w alert logu wpisy FAL: Fetch Archive Log. Mechanizm ten pozwala rozwiązywać sytuację, której np. log nie został nigdy przesłany lub też został prawidłowo przesłany do bazy standby, jednakże przed jego użyciem został usunięty z dysku. Dzięki temu mechanizmowi baza standby może zażądać od bazy produkcyjnej ponownego przesłania logów. Aby ten mechanizm działał prawidłowo należy ustawić następujące parametry:

- `fal_server` – nazwa serwisu Oracle Net, który jest prawidłowo skonfigurowany na bazie standby i wskazuje na serwer dokąd należy wysłać żądanie przesłania pliku,
- `fal_client` – nazwa serwisu Oracle Net, który jest prawidłowo skonfigurowany na bazie produkcyjnej i wskazuje na serwer dokąd należy wysłać żądany plik.

Dla prawidłowego działania tego mechanizmu bardzo ważna jest prawidłowa konfiguracja odpowiednich serwisów Oracle Net. Parametry te powinny być ustawione na bazi standby. Jednakże, jeśli chcemy dokonywać zmiany ról baz danych (switchover lub failover) to warto skonfigurować te parametry również na bazie produkcyjnej.

Automatic Gap Resolution może funkcjonować prawidłowo tylko w przypadku, kiedy dysponujemy wystarczającą przepustowością łącza oraz procesy archivera na bazie produkcyjnej nie są zajęte archiwizowaniem aktualnych logów. Dlatego też zalecane jest skonfigurowanie wielu procesów. Nie zauważyłem, jak do tej pory, żadnych skutków ubocznych wynikających z uruchomienia „zbyt” wielu procesów archivera.

Poniżej ilustracja sytuacji, w której proces MRP czeka na przesłanie brakujących logów a te nie są przesyłane ponieważ archiver jest zajęty przesyłaniem aktualnych logów:

```
SQL> select process,status,sequence# from v$managed_standby;
```

PROCESS	STATUS	SEQUENCE#
ARCH	CLOSING	0
ARCH	CLOSING	0
RFS	WRITING	476

```
MRP0          WAIT_FOR_GAP          235
```

Z powyższego zapytania wynika, że na bazie produkcyjnej skonfigurowane są dwa procesy archivera. Jeden z nich jest odpowiedzialny za archiwizację lokalnie na dysk a drugi za przesył do bazy standby. Ponieważ dla każdego archivera kontaktującego się z baza standby jest startowany osobny proces RFS, dochodzimy do wniosku, że tylko jeden archiver przesyła pliki do bazy standby.

W celu ustalenia czy na bazie standby brakuje logów można również wykorzystać widok `v$archive_gap`. Informacje znajdujące się w tym widoku są aktualne, tylko jeśli proces MRP zauważył brak plików. W praktyce spotkałem się z sytuacją, że proces MRP nie działał, bo zakończył się po wykryciu uszkodzonego logu:

```
ORA-00317: file type 0 in header is not log file

ORA-00334:                                archived                                log:
'/oradata/sby/archive/2008_12_01/01_mf_1_418_4djms7gh_.arc'

MRP0: Background Media Recovery process shutdown (SBY)
```

W takiej sytuacji proces RFS nadal akceptował przychodzące logi i klient dopiero po pewnym czasie odkrył, że na bazie standby nie działa proces MRP i że powstał Gap (uszkodzony plik). Wtedy jednak log ten nie był już dostępny na maszynie produkcyjnej i trzeba było odtwarzać go z backupu.

2.2. Manualne “Gap Resolution”

Automatyczne rozwiązywanie problemu brakujących logów działa zwykle bardzo dobrze. W praktyce spotkałem się jednakże z sytuacjami, kiedy nawet przy ustawionych parametrach FAL, logi nie były kopiowane:

```
FAL[client]: Failed to request gap sequence
GAP - thread 1 sequence 400-400
DBID 3954948037 branch 662646405
FAL[client]: All defined FAL servers have been attempted.
```

Niestety nie udało mi się ustalić przyczyny takiego zachowania bazy danych. Jednakże również taki problem może zostać rozwiązany, wymaga to jednak manualnej interwencji administratora.

Jeżeli brakuje nam tylko kilku logów, najprostszym rozwiązaniem jest skopiowanie tych plików z serwera produkcyjnego na serwer standby i zarejestrowanie ich w bazie standby. Można to zrobić na dwa sposoby:

- `sqlplus`
 - o `alter Database register logfile '/kat/log_1_1234.arc';`
- `rman`
 - o `katalog start with '/katalog_gdzie_sa_logi';`

Użycie `RMAN`'a jest o wiele wygodniejsze, bowiem jednym poleceniem możemy zarejestrować wiele logów. W przypadku `sqlplus`'a musimy rejestrować każdy log osobno.

Jak tylko log zostanie zarejestrowany, proces MRP zainicjuje jego integrację do bazy. Nie musimy wydawać dodatkowego polecenia „`recover`”. Jest to prawda przy założeniu, że baza jest w trybie `Managed Recovery`.

Często jednak zdarza się, że mamy do czynienia z brakiem wielu logów. Tak późne rozpoznanie problemu zwykle jest to skutkiem ubocznym braku odpowiedniego monitoringu baz. Ręczne kopiowanie wszystkich brakujących logów może być w takiej sytuacji uciążliwe i czasochłonne. Zwykle logi zostały już usunięte z maszyny produkcyjnej i muszą być odtworzone z kopii zapasowej a ich sumaryczny rozmiar dochodzi do kilkuset GB. W takiej sytuacji łatwiejszym może być wykonanie backupu inkrementalnego bazy produkcyjnej i odtworzenie go na bazie standby.

```
BACKUP INCREMENTAL FROM SCN 233995 DATABASE FORMAT '/tmp/ForStandby_%U'
```

Jest to specjalny rodzaj backupu, który nie jest katalogowany w bazie produkcyjnej. Tak więc nie ma to wpływu na strategię tworzenia kopii zapasowych systemu produkcyjnego (retention policy, zależności pomiędzy kopiami inkrementalnymi i pełnymi).

Tak utworzoną kopię zapasową należy przenieść na system standby i odtworzyć ją przy pomocy następującego polecenia RMAN:

```
recover database noredo;
```

“Noredo” to specjalna opcja dla baz danych standby, które nie posiadają własnych plików dziennika powtórzeń.

Te kroki są opisane w dokumentacji. Brakuje tam jednak informacji o kolejnym kroku. Mianowicie po odtworzeniu backupu baza danych standby nadal czeka na brakujące logi, mimo że nie są one już potrzebne. W tej sytuacji należy utworzyć nowy „standby Controlfile” na bazie produkcyjnej i przenieść do bazy standby.

3. DataGuard Broker

Data Guard może być administrowany przy pomocy poleceń SQL. Oracle zaleca jednak użycie w tym celu programu Data Guard Broker. Program ten ma, podobnie jak RMAN, własny interfejs (dgmgrl) oraz własny zestaw poleceń. Każde polecenie SQL służące zarządzaniu bazami standby ma odpowiednik wśród poleceń Data Guard Brokera.

W tym kontekście pojawia się pytanie czy należy używać Data Guard Brokera? Jakie są jego zalety? Z mojego punktu widzenia Data Guard Broker ma trzy zasadnicze zalety.

Pierwszą z nich jest łatwość wykonywania operacji failover i switchover, czyli zamianę ról pomiędzy bazą produkcyjną i bazą standby. Wystarczy wydanie jednego polecenia w interfejsie dgmgrl i Data Guard Broker zajmuje się przeprowadzeniem całej operacji. Wykonanie operacji „switchover” przy pomocy poleceń SQL wymaga wydania ich w ściśle określonej kolejności na bazie produkcyjnej i bazie standby. Ta procedura jest podatna na błędy ludzkie, jako że zwykle wykonywana jest w sytuacji stresującej i pod presją czasu.

Data Guard Broker umożliwia również monitorowanie baz standby przy pomocy Grid Control. Ten aspekt omówiony będzie w następnym rozdziale.

Inną zaletą Data Guard Brokera jest to, że po starcie bazy standby automatycznie uruchamia on recovery na bazie standby oraz koryguje ustawienia parametrów na bazie produkcyjnej. Oszczędza to pisanie dodatkowych skryptów, które muszą być wykonane po restarcie serwera czy operacji failover.

Aktywacja procesu Data Guard Broker

W jaki sposób uruchamia się Data Guard Broker’a? Wystarczy ustawić następujący parametr i proces ten będzie automatycznie uruchamiany przy starcie instancji:

```
alter system set dg_broker_start=true;
```

Później można już korzystać z narzędzia dgmgrl do konfiguracji i zarządzania Data Guardem.

```
dgmgrl /
```

Istotne jest, że po uruchomieniu Data Guarda i skonfigurowaniu go przy pomocy Data Guard Brokera, należy zawsze używać interfejsu dgmgrl do zmiany parametrów. W przeciwnym wypadku Data Guard Broker będzie raportował ORA-16xxx błędy:

```
ORA-16792: configuration property value is inconsistent with database setting
```

Mimo takich błędów system zwykle funkcjonuje dalej, jednakże należy unikać niepotrzebnych i mylnych komunikatów o błędach. Tak więc, jeśli raz uruchomimy Data Guard Brokera nie powinniśmy używać poleceń SQL.

Dotyczy to parametrów, które na pierwszy rzut oka nie mają nic wspólnego z Data Guard Broker'em, np. `log_archive_max_processes`

Deaktywacja Data Guard Brokera

Deaktywacja jest tak samo prosta jak aktywacja:

```
alter system set dg_broker_start=false;
```

Po deaktywacji Data Guard Brokera powinno się również usunąć jego pliki konfiguracyjne. Znajdują się one w lokalizacji określonej parametrami:

```
dg_broker_config_file1
```

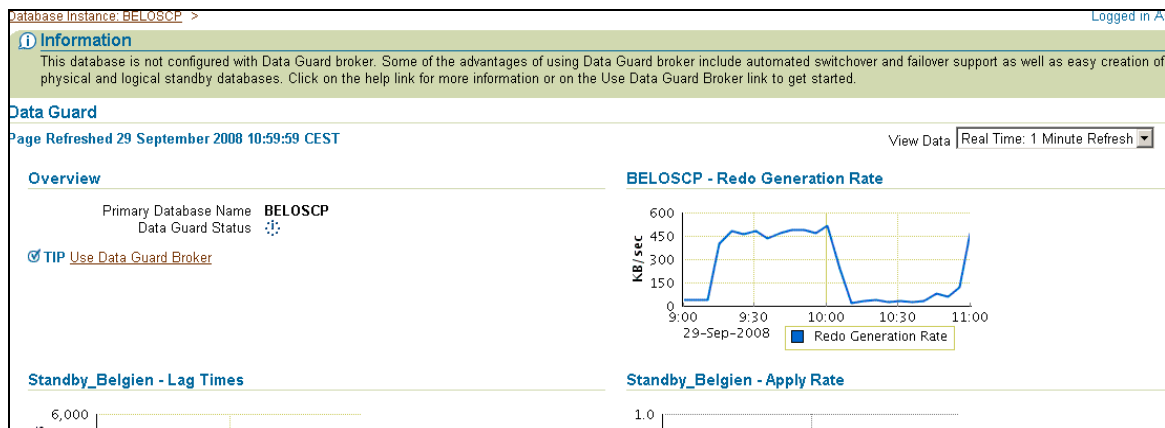
```
dg_broker_config_file2
```

Jeśli nie usunie się tych plików, przy uruchomieniu Data Guard Brokera w przyszłości zostaną one wczytane, co może spowodować zamieszanie.

4. Monitorowanie DataGuard za pomocą GridControl

Oracle zaleca monitorowanie Data Guard'a przy pomocy Enterprise Manage Grid Control. DB Console nie jest wystarczająca. Również niezbędny jest Data Guard Broker, jako że GC używa go do monitorowania bazy standby.

Bez działającego Brokera strona Data Guard'a w GC wygląda następująco:



Rys. 2. Strona Data Guard w GC bez Data Guard Brokera

W szczególności włączony Data Guard Broker umożliwia monitorowanie transportu i aplikowania logów na bazie standby. Przekroczenie definiowalnych poziomów powoduje wygenerowanie ostrzeżenia lub alarmu.

Odpowiednie metryki są dostarczane wraz ze standardową instalacją. Są one widoczne na stronie bazy standby. Jednakże nie mają one zdefiniowanych poziomów alarmowych. Odpowiednie reguły powiadamiania również nie są zdefiniowane. Aby otrzymywać powiadomienia o alarmach użytkownik musi więc dokonać odpowiedniej konfiguracji.

Gdy tylko wszystkie powyżej opisane wymagania zostały spełnione, strona w GC wygląda następująco:



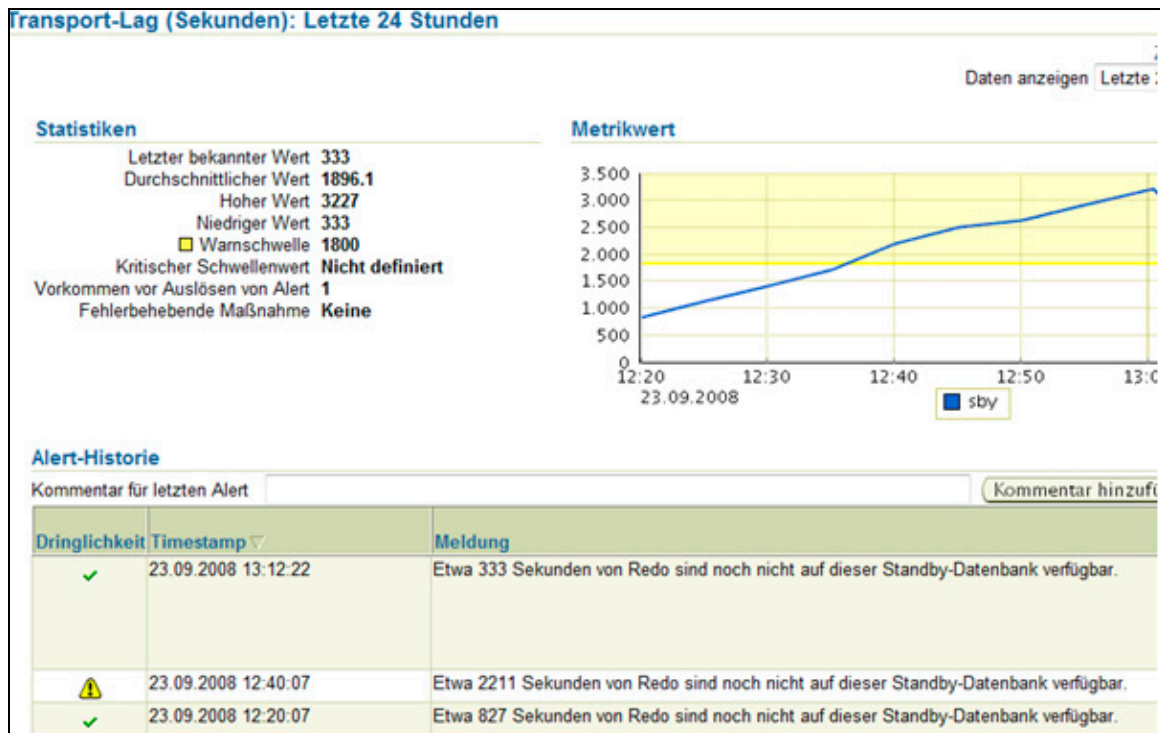
Rys. 3. Strona Data Guard w GC z Data Guard Broker'em

Widoczny tutaj alert był spowodowany zmianą parametru `log_archive_max_processes` przy użyciu SQL'a a nie polecenia `dgmgrl`:

```
alter system set log_archive_max_processes=10;
```

Jest to jeden z parametrów, które powinny być zmieniane przy pomocy poleceń `dgmgrl`.

Inny typowy alert wygląda następująco:



Rys. 4. Przykładowa metryka – Transport Lag

Metryka „Transport-Lag” pokazuje odległość w czasie od ostatnio przetransportowanego logu. Taka sytuacja może być spowodowana przez

- przerwanie przesyłu logów z bazy produkcyjnej np. z powodu problemów z siecią,
- lub też przez fakt, że na bazie produkcyjnej nie nastąpiło przełączenie się pomiędzy logami. Najczęstszą przyczyną jest niewielka liczba transakcji i nieustawiony parametr `archive_lag_target`

7. Podsumowanie

Praktyka pokazuje, że technologia Data Guard jest niezawodnym narzędziem pozwalającym na podniesienie dostępności systemów baz danych. Prawidłowo skonfigurowana i monitorowana nie wymaga dużych nakładów administracyjnych.

Oprócz zastosowania do celów Disaster Recovery, baza standby może być również wykorzystywana jako baza do generacji raportów (w trybie Read-Only) i/lub do tworzenia kopii zapasowych bazy bez obciążania systemów produkcyjnego.

Bibliografia

- [1] Dokumentacja Oracle – Administrator's Guide
- [2] Dokumentacja Oracle – Data Guard Broker
- [3] Dokumentacja Oracle – Data Guard Concepts and Administration