

XII Konferencja PLOUG
Zakopane
Październik 2006

Rozproszona replikacja danych z wykorzystaniem Oracle Streams

Maciej Zakrzewicz

*PLOUG, Politechnika Poznańska
mzakrz@cs.put.poznan.pl*

Streszczenie

Realizacja wielu typów zaawansowanych systemów baz danych wymaga rozpraszania i replikowania danych. Najczęściej powodem takich działań jest zapotrzebowanie na niezawodność lub skalowalność systemów. Jedną z technologii, które umożliwiają implementację złożonych mechanizmów rozproszonej replikacji danych jest Oracle Streams 10g. Oracle Streams 10g umożliwia m.in. przechwytywanie operacji wykonywanych przez użytkowników w systemie bazy danych, ich przesyłanie do zdalnych węzłów, a następnie aplikowanie w docelowych systemach baz danych. Celem referatu jest przedstawienie praktycznych zastosowań funkcjonalności Oracle Streams 10g.

1. Wprowadzenie

Replikacja danych to technologia informatyczna, która znajduje wiele zastosowań w dziedzinach m.in. hurtowni danych, zapewniania niezawodności systemów baz danych, optymalizacji wydajności baz danych. Realizacja replikacji danych w środowisku systemów zarządzania bazami danych Oracle była od wielu lat wspomagana przez takie mechanizmy jak łączniki bazodanowe (Database Links), migawki (Snapshots), perspektywy materializowane (Materialized Views), narzędzia Export/Import, Data Pump, Data Guard, itp. Począwszy od wersji 9i systemu zarządzania bazą danych Oracle zbiór tych metod został rozszerzony o kolejną: Oracle Streams. Oracle Streams to uniwersalna technologia zdarzeniowej replikacji danych pomiędzy wieloma bazami danych Oracle. Oracle Streams umożliwia m.in. budowę środowisk replikacyjnych o topologii jeden-do-jednego, jeden-do-wielu, wiele-do-jednego, wiele-do-wielu, środowisk klasy Single-Master i Multimaster, środowisk replikacji pośredniej, itp.

Podstawowa idea technologii Oracle Streams opiera się na użyciu trzech typów komponentów programowych: Capture, Propagate, Apply. Komponent Capture prowadzi nasłuch operacji wykonywanych przez użytkowników w systemie źródłowym, komponent Propagate przesyła wychwycone operacje do systemu docelowego, a komponent Apply powiela te operacje w docelowej bazie danych. Do obserwacji (nasłuchu) operacji wykonywanych w źródłowej bazie danych przez użytkowników wykorzystuje się dziennik powtórzeń. Do tymczasowego przechowywania wychwyconych operacji służą struktury kolejkowe Oracle Advanced Queueing. Replikowane operacje mogą być automatycznie filtrowane i transformowane w każdym punkcie architektury systemu.

Celem tego artykułu jest przegląd architektury i funkcjonalności Oracle Streams 10g w kontekście implementacji środowisk replikacji danych. Struktura artykułu jest następująca. W rozdziale 2 dokonano przeglądu rozwiązań technicznych, na których bazuje technologia Oracle Streams: dzienniki powtórzeń, obiekty LCR i kolejki Oracle Advanced Queueing. Rozdział 3 omawia trzy podstawowe komponenty Oracle Streams: Capture, Propagate i Apply. Rozdział 4 przedstawia prostą przykładową implementację replikacji w środowisku Oracle Streams. W rozdziale 5 przedstawiono zaawansowane rozwiązania Oracle Streams: reguły filtrujące, transformację regułową, replikację symetryczną. Rozdział 6 jest podsumowaniem.

2. Technologie podstawowe

2.1. Dziennik powtórzeń

Podstawowym źródłem informacji o operacjach modyfikacji danych wykonywanych przez użytkowników w systemie źródłowym jest dziennik powtórzeń (Redo Log). Dziennik powtórzeń to plikowy bufor cykliczny, w którym każdy system zarządzania bazą danych Oracle rejestruje realizowane operacje DML i DDL. Podstawowym zastosowaniem dziennika powtórzeń jest ochrona zawartości bazy danych przed uszkodzeniem w wyniku utraty zapisów z bufora danych (np. w wyniku zaniku zasilania serwera). Istnieją narzędzia, jak np. Log Miner, które umożliwiają rekonstrukcję oryginalnych poleceń SQL na podstawie zapisów umieszczonych w dzienniku powtórzeń.

W wielu systemach baz danych Oracle dzienniki powtórzeń podlegają archiwizacji w celu umożliwienia pełnego odtwarzania stanu bazy danych po awarii. Za archiwizację plików dziennika powtórzeń odpowiada specjalizowany proces o nazwie Archiver (ARCH).

2.2. Obiekty LCR

Zmiany dokonywane w bazie danych przez użytkowników posługujących się poleceniami DML są reprezentowane przez Oracle Streams w formie obiektów nazywanych Row Logical Change Records (Row LCR). Każdy obiekt Row LCR opisuje pojedynczą zmianę danych za pomocą następujących atrybutów:

- `source_database_name` – nazwa bazy danych, w której nastąpiła zmiana,
- `command_type` – typ polecenia DML, które spowodowało zmianę, np. „INSERT”, „UPDATE”,
- `object_owner` – identyfikator właściciela tabeli, w której nastąpiła zmiana,
- `object_name` – nazwa tabeli, w której nastąpiła zmiana,
- `tag` – opcjonalny marker wykorzystywany do filtrowania obiektów LCR,
- `transaction_id` – identyfikator transakcji, w ramach której nastąpiła zmiana,
- `scn` – numer SCN z chwili zapisu zmiany do pliku dziennika powtórzeń,
- `old_values` – wcześniejsze wartości pól modyfikowanego rekordu,
- `new_values` – późniejsze wartości pól modyfikowanego rekordu.

Zmiany strukturalne dokonywane w bazie danych w związku z realizacją poleceń DDL są opisywane przez Oracle Streams za pomocą obiektów nazywanych DDL Logical Change Records (DDL LCR). Każdy obiekt DDL LCR opisuje pojedynczą zmianę struktury danych posługując się następującymi atrybutami:

- `source_database_name` – nazwa bazy danych, w której nastąpiła zmiana,
- `command_type` – typ polecenia DDL, które spowodowało zmianę, np. „ALTER TABLE”,
- `object_owner` – identyfikator właściciela obiektu, którego dotyczy zmiana,
- `object_name` – nazwa obiektu, którego dotyczy zmiana,
- `object_type` – typ obiektu, którego dotyczy zmiana, np. „TABLE”, „INDEX”,
- `ddl_text` – treść oryginalnego polecenia SQL DDL,
- `logon_user` – identyfikator użytkownika, w którego sesji wykonano polecenie DDL,
- `current_schema` – identyfikator domyślnego schematu dla polecenia SQL DDL,
- `base_table_owner` – identyfikator właściciela tabeli, od której zależne było wykonane polecenie SQL DDL,
- `base_table_name` – nazwa tabeli, od której zależne było wykonane polecenie SQL DDL,
- `tag` – opcjonalny marker wykorzystywany do filtrowania obiektów LCR,
- `transaction_id` – identyfikator transakcji, w ramach której nastąpiła zmiana,
- `scn` – numer SCN z chwili zapisu zmiany do pliku dziennika powtórzeń.

2.3. Struktury kolejkowe Oracle AQ

Obiekty LCR służą jako nośniki informacji o operacjach wykonywanych przez użytkowników w bazie danych. Do ich tymczasowego przechowywania wykorzystuje się struktury kolejkowe FIFO Oracle Advanced Queueing (Oracle AQ). Każda kolejka AQ umożliwia przechowywanie obiektów jednego typu danych. Ponieważ obiekty LCR występują w dwóch odmianach – Row

LCR i DDL LCR – to konieczne było zaimplementowanie rozwiązania kolejki polimorficznej. Efekt takiej kolejki uzyskano poprzez wprowadzenie metatypu danych o nazwie SYS.AnyData, który może być uniwersalnym nośnikiem dla dowolnych struktur obiektowych. Wszystkie kolejki AQ wykorzystywane w środowisku Oracle Streams operują na obiektach typu SYS.AnyData.

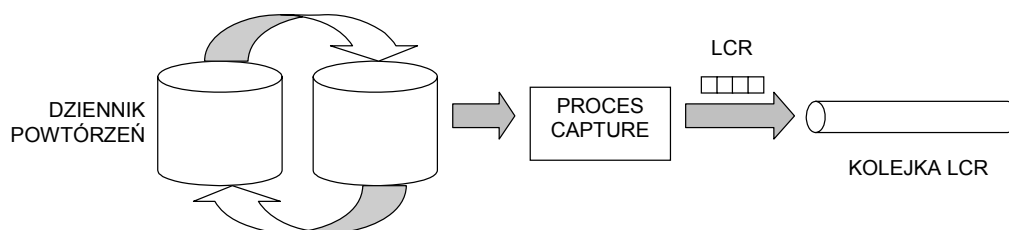
Podstawowymi operacjami wykonywanymi w odniesieniu do kolejek AQ są operacje umieszczenia w kolejce (Enqueue) i pobrania z kolejki (Dequeue). Zawartość kolejek jest przechowywana w sposób trwały w tabelach bazy danych. Do tworzenia kolejek i ich tabel-nosicieli służy procedura SET_UP_QUEUE pakietu standardowego DBMS_STREAMS_ADM.

3. Komponenty Oracle Streams

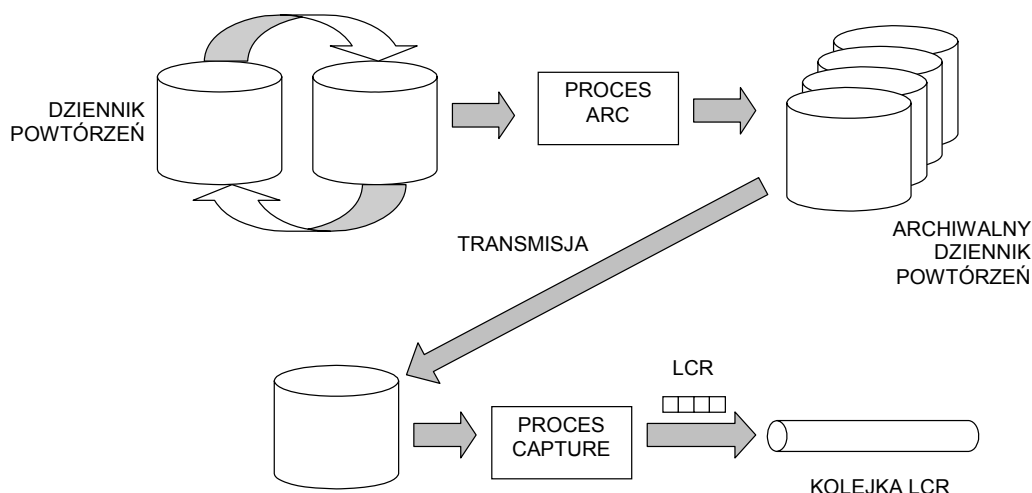
3.1. Detekcja zmian

Detekcja zmian dokonywanych przez użytkowników w źródłowej bazie danych jest realizowana przez specjalizowany proces o nazwie Capture – proces ten posługuje się w systemie operacyjnym identyfikatorami *Cnnn*. Proces Capture prowadzi analizę plików dziennika powtórzeń w celu ekstrakcji opisów zmian DML i DDL dokonywanych przez użytkowników bazy danych. Na podstawie tych opisów generuje obiekty Row LCR i DDL LCR, które następnie umieszcza we wskazanej strukturze kolejkowej. Do ekstrakcji opisów zmian z plików dziennika powtórzeń wykorzystywana jest infrastruktura narzędzia Log Miner.

Proces Capture może być wykorzystywany w dwóch konfiguracjach: detekcji lokalnej (Local Capture) i detekcji zdalnej (Downstream Capture). Z detekcją lokalną mamy do czynienia wtedy, kiedy proces Capture pracuje na tym samym komputerze, na którym znajduje się system zarządzania źródłową bazą danych. Obiekty LCR są wtedy tworzone wprost na podstawie zawartości plików bieżącego dziennika powtórzeń. Detekcja zdalna polega na osadzeniu procesu Capture po stronie docelowego systemu zarządzania bazą danych i na wykorzystaniu mechanizmów transportu plików dziennika powtórzeń LGWR/RFS lub ARC. W ramach tego rozwiązania, pliki dziennika powtórzeń powstające po stronie źródłowej bazy danych są w niezmienionej postaci transferowane do systemu docelowego i dopiero tam podlegają analizie, w wyniku której są generowane i kolej-kowane obiekty LCR. Architektura procesu detekcji zmian pracującego w konfiguracji lokalnej przedstawiono na rys. 1, a pracującego w konfiguracji zdalnej – na rys. 2.



Rys. 1. Proces detekcji w konfiguracji lokalnej



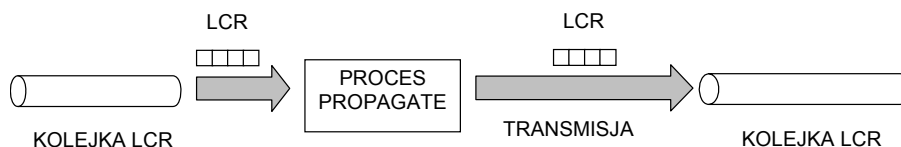
Rys. 2. Proces detekcji w konfiguracji zdalnej

Zaletami wykorzystywania zdalnej konfiguracji procesu detekcji zmian są m.in.: odciążenie systemu źródłowego, gdyż przetwarzanie plików dziennika powtórzeń odbywa się w środowisku docelowym, a także prostota architektury w przypadku, gdy replikacja następuje z wielu systemów źródłowych do jednego systemu docelowego. Należy jednak podkreślić występujące tu zagrożenia dla poufności danych, gdyż transferowane dzienniki powtórzeń mogą zawierać także te informacje, które nie podlegają replikacji i nie powinny być udostępniane.

Do konfiguracji i zarządzania procesem Capture służy pakiet standardowy o nazwie `DBMS_CAPTURE_ADM`. Obecna w tym pakiecie procedura `CREATE_CAPTURE` służy do definiowania nowej instancji procesu Capture. W pojedynczym systemie może równocześnie funkcjonować wiele procesów Capture. Instancje procesów Capture są również tworzone jako efekt uboczny stosowania procedur pakietu `DBMS_STREAMS_ADM`.

3.2. Propagacja zmian

Opisy zmian dokonywanych w bazie danych przez użytkowników, reprezentowane w formie obiektów LCR, są zapisywane w strukturach kolejkowych. W przypadku wykorzystywania detekcji lokalnej istnieje potrzeba przesłania zawartości lokalnych struktur kolejkowych do systemu docelowego. Za taką propagację zmian odpowiada proces o nazwie Propagate. Proces Propagate odczytuje obiekty LCR z kolejki w systemie źródłowym, a następnie zapisuje je do kolejki w systemie docelowym. Przesłane obiekty są usuwane z kolejki w systemie źródłowym. Proces Propagate funkcjonuje jako zadanie wsadowe (Job) po stronie systemu źródłowego (rys. 3).



Rys. 3. Architektura procesu propagacji zmian

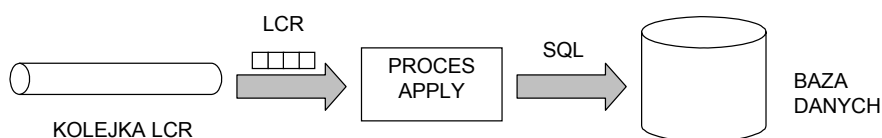
Proces Propagate pracuje w trybie zegarowym – jest uruchamiany w regularnych odstępach czasu określonych przez administratora. Do konfiguracji i zarządzania procesem Propagate służy pakiet standardowy o nazwie `DMBS_PROPAGATION_ADM`. Obecna w tym pakiecie procedura `CREATE_PROPAGATION` służy do definiowania nowego zadania procesu Propagate. Zadania

procesów Propagate są również tworzone jako efekt uboczny stosowania procedur pakietu DBMS_STREAMS_ADM.

3.3. Aplikowanie zmian

W celu poprawnej replikacji danych, operacje DML i DDL reprezentowane w formie obiektów LCR powinny być zaaplikowane w docelowym systemie bazy danych. Do aplikowania zmian w systemie docelowym służy specjalizowany proces o nazwie Apply – w systemie operacyjnym posługuje się on identyfikatorami *Annn*.

Proces Apply pobiera obiekty LCR ze struktury kolejkowej i wykonuje opisane w nich operacje. Jeżeli operacje te nie mogą być poprawnie wykonane, np. z powodu braku tabeli o nazwie takiej, jak w systemie źródłowym, to cała wadliwa transakcja (jej obiekty LCR) jest przepisywana do wyróżnionej struktury kolejkowej nazywanej kolejką błędów (Error Queue). Architektura procesu Apply została zilustrowana na rys. 4.

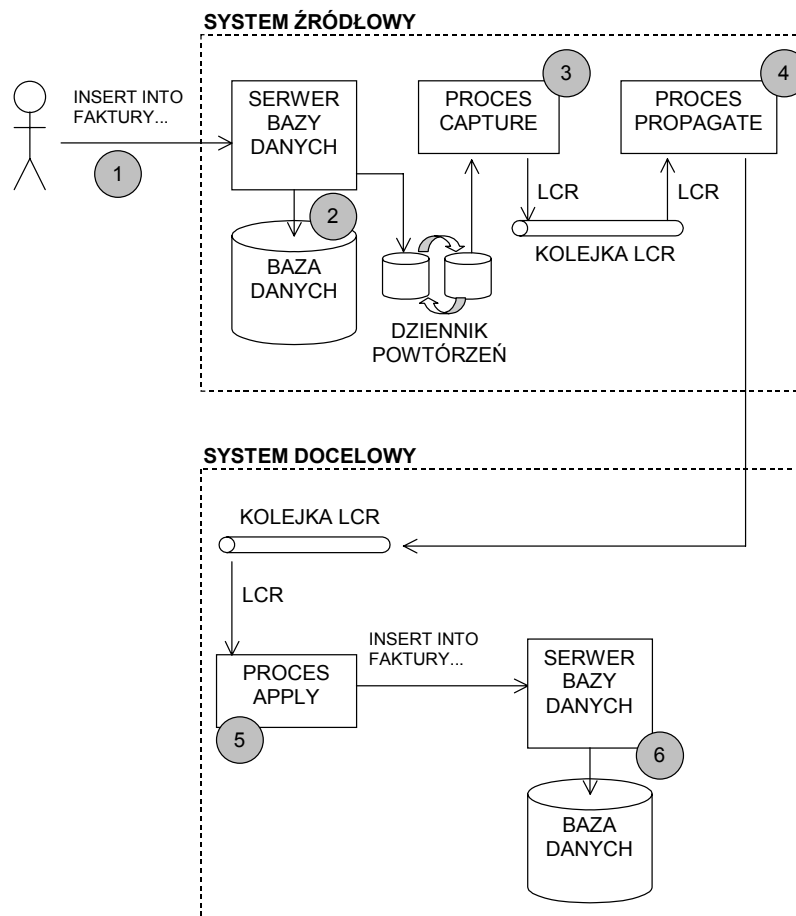


Rys. 4. Architektura procesu aplikowania zmian

Do konfiguracji i zarządzania procesem Apply służy pakiet standardowy o nazwie DBMS_APPLY_ADM. Obecna w tym pakiecie procedura CREATE APPLY służy do definiowania nowej instancji procesu Apply. Instancje procesów Apply są również tworzone jako efekt uboczny stosowania procedur pakietu DBMS_STREAMS_ADM.

4. Implementacja replikacji danych w środowisku Oracle Streams

Implementacja replikacji danych w środowisku Oracle Streams polega na definicji i aktywacji procesów Capture, Propagate i Apply w różnych punktach rozproszonej architektury. Na rys. 5 przedstawiono przykładowe rozwiązanie replikacji. Znaczenie wyróżnionych elementów rysunku jest następujące. W kroku (1) użytkownik systemu źródłowego generuje polecenie SQL DML. W kroku (2) polecenie to jest przetwarzane przez system zarządzania źródłową bazą danych i powoduje zapis nowego rekordu do bazy danych oraz rejestrację operacji w pliku dziennika powtórzeń. W kroku (3) proces Capture odczytuje z pliku dziennika powtórzeń zarejestrowane polecenie i umieszcza odpowiadający mu obiekt LCR w strukturze kolejkowej w systemie źródłowym. Krok (4) wiąże się z przeniesieniem obiektu LCR z kolejki w systemie źródłowym do kolejki w systemie docelowym. Zadanie to wykonywane jest przez proces Propagate. W kroku (5) proces Apply pracujący w systemie docelowym odbiera z kolejki obiekt LCR i na jego podstawie generuje polecenie SQL DML. W kroku (6) polecenie SQL DML jest wykonywane przez system zarządzania docelową bazą danych. Replikacja została zakończona.



Rys. 5. Przykładowa implementacja replikacji danych

5. Rozwiązania zaawansowane

5.1. Reguły filtrujące

O ile administrator systemu nie wskaże inaczej, procesy Capture, Propagate i Apply przetwarzają wszystkie nadchodzące informacje – proces Capture generuje obiekty LCR dla wszystkich operacji DML i DDL, proces Propagate propaguje wszystkie obiekty LCR, proces Apply aplikuje wszystkie zmiany opisane w formie obiektów LCR. W większości zastosowań jest to podejście dalece niepraktyczne – zwykle replikacji danych powinny podlegać tylko wybrane tabele, a czasem tylko wybrane rekordy. Do implementacji selektywnej replikacji danych stosuje się tzw. reguły filtrujące (Rules).

Reguły filtrujące to wyrażenia logiczne, operujące na atrybutach obiektów LCR, ewaluowane przez procesy Capture, Propagate i Apply. Mogą się odnosić np. do nazwy tabeli źródłowej, wartości pól modyfikowanego rekordu, rodzaju operacji SQL, itd. Reguły filtrujące są grupowane w tzw. zbiory reguł filtrujących (Rule Sets), które są wskazywane procesom Capture, Propagate i Apply. Zbiory reguł filtrujących mogą być przez te procesy traktowane jako tzw. zbiory pozytywne (Positive Rule Sets) lub zbiory negatywne (Negative Rule Sets). Spełnienie przez obiekt LCR reguły filtrującej ze zbioru pozytywnego powoduje dalsze przetwarzanie obiektu. Spełnienie reguły filtrującej ze zbioru negatywnego powoduje zaprzestanie dalszego przetwarzania obiektu.

Definiowanie zbiorów reguł filtrujących jest zadaniem dla administratora środowiska replikacji danych. Do tworzenia nowych reguł służą procedury pakietu standardowego DBMS_STREAMS_ADM: ADD_TABLE_RULES (dla procesów Capture i Apply), ADD_SCHEMA_RULES (dla procesów Capture i Apply), ADD_GLOBAL_RULES (dla procesów Capture i Apply), ADD_SUBSET_RULES (dla procesów Capture i Apply), ADD_TABLE_PROPAGATION_RULES (dla procesu Propagate), ADD_SCHEMA_PROPAGATION_RULES (dla procesu Propagate), ADD_GLOBAL_PROPAGATION_RULES (dla procesu Propagate), ADD_SUBSET_PROPAGATION_RULES (dla procesu Propagate).

5.2. Transformacje

Istnieje wiele przypadków, w których replikacja danych nie ma charakteru 1:1, tzn. nie następuje dokładne powielanie rekordów pomiędzy identycznymi tabelami w systemach źródłowym i docelowym. Często wartości danych podlegają konwersji, normalizacji czy translacji. Zdarza się, że nazwa tabeli docelowej jest odmienna od nazwy tabeli źródłowej. W celu umożliwienia implementacji konwersji replikowanych danych, Oracle Streams udostępnia mechanizm tzw. transformacji regułowej (Rule-Based Transformations).

Implementacja transformacji regułowej polega na realizacji funkcji PL/SQL modyfikujących obiekty LCR. Z funkcjami tymi związane są reguły określające ich aktywację. Jeżeli spełniona jest reguła aktywacyjna, to obiekt LCR jest przekazywany do funkcji PL/SQL, która może zmienić jego treść przed kontynuacją przetwarzania. Do definiowania reguł aktywacyjnych służy procedura SET_RULE_TRANSFORM_FUNCTION pakietu standardowego DBMS_STREAMS_ADM.

5.3. Replikacja symetryczna

Replikacja danych może mieć postać replikacji symetrycznej, w ramach której replika może być w systemie docelowym modyfikowana przez użytkowników, a zmiany przekazywane zwrótnie do systemu źródłowego. W takich rozwiązaniach mogą występować kolizje związane ze współbieżną modyfikacją tych samych danych w dwóch systemach – źródłowym i docelowym. Oracle Streams umożliwia deklaratywne lub programowe rozwiązywanie takich kolizji. Deklaratywne rozwiązywanie kolizji może być stosowane tylko w stosunku do kolizji typu UPDATE - polega na stosowaniu predefiniowanych procedur obsługi: OVERWRITE, DISCARD, MAXIMUM i MINIMUM. Procedura OVERWRITE powoduje nadpisanie wartości aktualnej przez wartość replikowaną. Procedura DISCARD powoduje odrzucenie wartości replikowanej. Procedury MAXIMUM i MINIMUM porównują wartości wskazanych pól w kolidujących wersjach rekordu i dokonują wyboru tej wersji, w której występuje wartość większa/mniejsza.

6. Podsumowanie

Oracle Streams jest interesującą technologią implementacji rozproszonej replikacji danych wykorzystującą model zdarzeniowy. Budowa środowiska replikacji polega na konfiguracji i aktywacji trzech komponentów systemowych – procesów Capture, Propagate i Apply. Zastosowania Oracle Streams obejmują: realizację zadań ETL w hurtowniach danych, zabezpieczanie danych przed utratą w wyniku awarii, audyt użytkowników, itp.

Literatura

- [1] "Oracle Streams Concepts and Administration 10g Release 2", February 2006, Oracle
- [2] "Oracle Streams Replication Administrator's Guide 10g Release 2", February 2006, Oracle
- [3] "Oracle Streams Advanced Queueing User's Guide and Reference 10g Release 2", June 2005, Oracle