

Rozproszone bazy danych – 2

Zarządzanie transakcjami rozproszonymi

Laboratorium przygotował:

Robert Wrembel



Plan laboratorium

- Transakcja rozproszona - podstawowe cechy
- Uczestnicy transakcji
- Zatwierdzanie transakcji - protokół 2PC
- Awarie transakcji rozproszonej
- "Ręczne" odtwarzanie transakcji
- Studium przypadku

ZSBD – laboratorium 1 (2)

Celem laboratorium jest przedstawienie podstawowych zagadnień zarządzania transakcjami rozproszonymi. Laboratorium obejmuje:

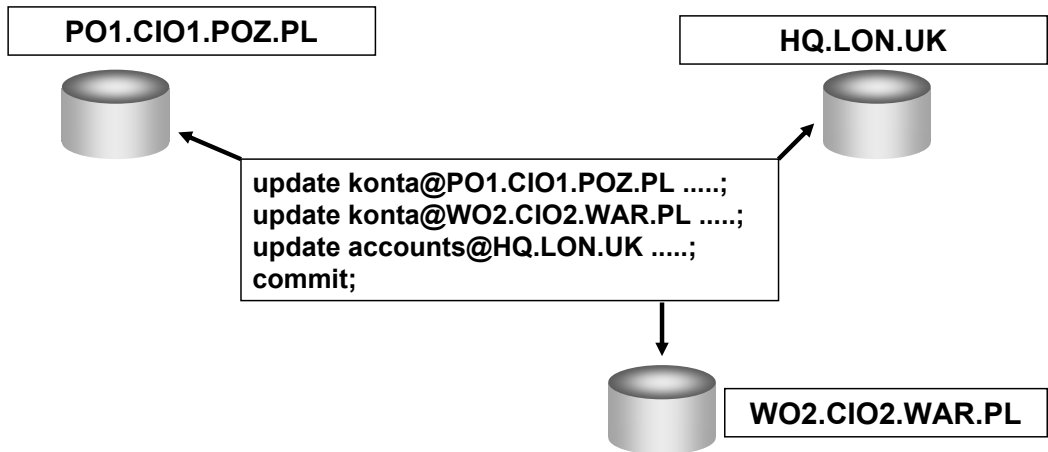
- przedstawienie definicji i podstawowych cech transakcji rozproszonej,
- omówienie ról baz danych biorących udział w transakcji rozproszonej,
- przedstawienie implementacji protokołu zatwierdzania transakcji rozproszonej (protokół 2-Phase-Commit 2PC),
- omówienie problemów związanych z awariami rozproszonej bazy danych w trakcie zatwierdzania transakcji rozproszonej,
- omówienie procedury odtwarzania transakcji rozproszonej, która uległa awarii, zilustrowane studium przypadku.

Wszystkie powyższe zagadnienia zostaną zilustrowane implementacją w systemie Oracle9i/10g.



Transakcja rozproszona (1)

- Transakcja, której polecenia INSERT, UPDATE, DELETE, SELECT odwołują się do tabel znajdujących się co najmniej w dwóch węzłach rozproszonej bazy danych



ZSBD – laboratorium 1 (3)

Transakcja rozproszona to transakcja, której polecenia DML (insert, update, delete) i select odwołują się do tabel znajdujących się co najmniej w dwóch węzłach rozproszonej bazy danych. Transakcja rozproszona jest często nazywana transakcją globalną.

Na slajdzie przedstawiono przykładową transakcję, która wykonuje polecenie *update* na tabeli *konta* w bazie danych wskazywanej łącznikiem PO1.CIO1.POZ.PL, polecenie *update* na tabeli *konta* w bazie danych wskazywanej łącznikiem WO2.CIO2.WAR.PL, *update* na tabeli *accounts* w bazie danych wskazywanej łącznikiem HQ.LON.UK.



Transakcja rozproszona (2)

- Transakcja rozproszona jest reprezentowana przez zbiór transakcji lokalnych
- W każdej z baz danych, do której odwołuje się transakcja rozproszona tworzona jest jedna transakcja lokalna
- Cechy transakcji rozproszonej
 - trwałość
 - spójność
 - izolacja
 - atomowość

ZSBD – laboratorium 1 (4)

Transakcja rozproszona jest reprezentowana przez zbiór **transakcji lokalnych**. W każdej z baz danych, do której odwołuje się transakcja rozproszona tworzona jest jedna transakcja lokalna. Zarówno każda z transakcji lokalnych jak i rozproszona mają cechy trwałości, spójności, izolacji i atomowości.



Transakcja rozproszona (3)

- Atomowość
 - wszystkie transakcje lokalne zatwierdzone lub wszystkie wycofane
- Atomowość zapewniana przez protokół zatwierdzania dwu-fazowego (ang. two-phase commit - 2PC)

ZSBD – laboratorium 1 (5)

Cecha atomowości w odniesieniu do transakcji rozproszonej oznacza, że wszystkie transakcje lokalne wchodzące w skład transakcji rozproszonej muszą zostać zatwierdzone. Jeśli choć jedna z transakcji lokalnych nie może zostać zatwierdzona, wówczas całą transakcję rozproszoną należy wycofać.

Zatwierdzanie lub wycofywanie transakcji rozproszonej, gwarantujące atomowość jest realizowane za pomocą tzw. **protokołu zatwierdzania dwu-fazowego** — *2PC* (ang. *two-phase commit*).



"Aktorzy" (1)

- Koordynator globalny (KG)
 - węzeł sieci, w którym zainicjowano transakcję rozproszoną
- Koordynator lokalny (KL)
 - węzeł sieci, któremu podlegają inne węzły
- Uczestnik (U)
 - węzeł sieci z transakcją lokalną

ZSBD – laboratorium 1 (6)

Transakcja rozproszona odwołuje się do kilku węzłów rozproszonej bazy danych. Każdy z tych węzłów pełni ściśle określoną funkcję. Wyróżnia się następujące rodzaje węzłów: koordynator globalny, koordynator lokalny, węzeł zatwierdzania, uczestnik.

Koordynator globalny (ang. global coordinator) jest tym węzłem z którego zainicjowano transakcję rozproszoną. Zadaniem koordynatora globalnego jest zarządzanie całą transakcją, tj. doprowadzenie jej w całości do zatwierdzenia lub wycofania.

Koordynator lokalny (ang. local coordinator) jest węzłem który otrzymuje żądanie, przetwarza je i wysyła kolejne żądania do innych podległych mu węzłów. *KL* nie koordynuje całej transakcji globalnej a jedynie te żądania, które sam wysłał.

Uczestnik (ang. node) jest węzłem, do którego jest kierowane żądanie transakcji rozproszonej, a sam węzeł nie wysyła żadnych żądań.



"Aktorzy" (2)

- Węzeł zatwierdzania (WZ)
 - inicjowanie zatwierdzania lub wycofywania transakcji zgodnie z komunikatem od koordynatora globalnego
 - zawiera status zatwierdzania transakcji rozproszonej
 - odczytywany przez transakcje lokalne

Węzeł zatwierdzania (ang. commit point site) jest węzłem, który jako pierwszy wykonuje zatwierdzenie swojej transakcji lokalnej w ramach całej transakcji rozproszonej. Dodatkowo, węzeł zatwierdzania przechowuje informacje o stanie transakcji rozproszonej. Informacje te są wykorzystywane w czasie manualnego kończenia transakcji po awarii systemu rozproszonego. Jeżeli stan transakcji rozproszonej przechowywany w węźle zatwierdzania przyjmuje wartość *zatwierdzona*, wówczas całą transakcję rozproszoną uznaje się za zatwierdzoną.



"Aktorzy" (3)

- Węzeł zatwierdzania
 - wybierany przez administratora systemu
 - parametr konfiguracyjny
COMMIT_POINT_STRENGTH
 - wartość 0-255
 - odzwierciedla ilość danych krytycznych w węźle
 - odzwierciedla niezawodność węzła
 - węzeł o najwyższej wartości
COMMIT_POINT_STRENGTH jest węzłem
zatwierdzania

Węzeł zatwierdzania jest wybierany przez koordynatora globalnego spośród wszystkich węzłów biorących udział w transakcji rozproszonej. Kryterium wyboru jest wartość parametru konfiguracyjnego instancji bazy danych COMMIT_POINT_STRENGTH. Spośród zbioru węzłów węzłem zatwierdzania staje się ten, który posiada najwyższą wartość parametru COMMIT_POINT_STRENGTH.

Wartość COMMIT_POINT_STRENGTH jest ustalana przez administratora systemu rozproszonego. Parametr ten może przyjąć wartości od 0 do 255. Węzeł najbardziej niezawodny powinien mieć największą wartość tego parametru.



Stan systemu w przypadku awarii

- Transakcja rozproszona jest uznawana za zatwierdzoną jeżeli zostanie zatwierdzona w węźle zatwierdzania, nawet jeśli pozostałe węzły jeszcze nie zatwierdziły swoich transakcji lokalnych



Protokół zatwierdzania dwu-fazowego (2PC)

- Fazy realizacji
 - przygotowanie (prepare)
 - zatwierdzanie (commit)
 - zakończenie (forget)

ZSBD – laboratorium 1 (10)

Zatwierdzanie lub wycofywanie transakcji rozproszonej, gwarantujące atomowość jest realizowane za pomocą tzw. **protokołu zatwierdzania dwu-fazowego** — *2PC* (ang. *two-phase commit*). Protokół *2PC* składa się z dwóch głównych faz przetwarzania, tj. **przygotowanie** i **zatwierdzanie** oraz z jednej fazy pomocniczej, tj. **zakończenie**.

W fazie przygotowania następuje przygotowanie transakcji lokalnych do zatwierdzania lub wycofywania. W fazie zatwierdzania następuje zatwierdzanie lub wycofywanie transakcji lokalnych. W fazie zakończenia następuje usuwanie z systemu informacji o transakcji rozproszonej i zwalnianie niezwolnionych w fazie zatwierdzania zasobów.



2PC - faza przygotowania (KG)

1. KG wybiera węzeł zatwierdzania
2. KG wysyła do uczestników żądanie przygotowania do zatwierdzania (uwaga: komunikat ten nie jest wysyłany do WZ)
3. Uczestnik przygotowuje się i wysyła komunikat PREPARED do KG
4. KG odbiera komunikaty od uczestników

Faza przygotowania jest realizowana w następujących krokach.

1. Koordynator globalny wybiera węzeł zatwierdzania spośród spośród wszystkich węzłów biorących udział w transakcji rozproszonej.
2. KG wysyła do wszystkich węzłów z wyjątkiem węzła zatwierdzania komunikat PREPARE, wymuszający przygotowanie się transakcji lokalnych do zatwierdzenia.
3. Po zakończeniu operacji przygotowania do zatwierdzenia transakcji, każdy węzeł przesyła do koordynatora globalnego komunikat PREPARED jeżeli się przygotował do zatwierdzenia swojej transakcji lokalnej.
4. W ostatnim kroku tej fazy koordynator globalny odbiera komunikaty od wszystkich węzłów, z wyjątkiem węzła zatwierdzania. Po ich odebraniu następuje przejście do drugiej fazy, tj. zatwierdzenia.



2PC - faza przygotowania (uczestnik) (1)

1. Odbiór komunikatu od KG żądającego przygotowania do zatwierdzenia
2. Zapis do plików dziennika powtórzeń
3. Wysłanie komunikatu PREPARE do podległych uczestnikowi węzłów
4. Odbiór komunikatów od węzłów podległych
5. Wysłanie komunikatu do koordynatora globalnego

ZSBD – laboratorium 1 (12)

W fazie przygotowania w węźle uczestnika są wykonywane następujące operacje:

1. Odbiór komunikatu PREPARE od koordynatora globalnego.
2. Dokonanie zapisów w bieżących plikach dziennika powtórzeń (ang. *online redo logs*). W plikach tych znajdują się wszystkie zmiany wykonane w bazie danych (np. wstawienie, zmodyfikowanie, usunięcie rekordu) wprowadzone zarówno przez zatwierdzone, jak i niezatwierdzone transakcje. Informacje z plików dziennika powtórzeń są wykorzystywane w czasie odtwarzania bazy danych po awarii.
3. Wysłanie komunikatu PREPARE do węzłów podległych.
4. Odbiór komunikatów od węzłów podległych.
5. Wysłanie komunikatu do koordynatora globalnego.



2PC - faza przygotowania (uczestnik) (2)

- W przypadku braku modyfikacji danych -> wysłanie do koordynatora globalnego komunikatu READ-ONLY
- Jeśli inne zdalne węzły podległe danemu węzłowi uczestnika zgłosiły gotowość i sam uczestnik jest gotów -> wysłanie komunikatu PREPARED do koordynatora globalnego
 - w przeciwnym przypadku
 - wycofanie lokalnej transakcji
 - wysłanie ABORT

ZSBD – laboratorium 1 (13)

Jeżeli w węźle uczestnika i w żadnym z węzłów mu podległych nie dokonano modyfikacji danych, wówczas uczestnik wysyła do koordynatora globalnego komunikat READ-ONLY.

Jeśli sam uczestnik i wszystkie węzły mu podległe przygotowały się do zatwierdzenia, uczestnik wysyła do koordynatora globalnego komunikat PREPARED. Jeśli natomiast albo uczestnik albo przynajmniej jeden z węzłów mu podległych nie mogły się przygotować, wówczas uczestnik wycofuje swoją transakcję lokalną i wysyła do koordynatora globalnego komunikat ABORT.



2PC - faza zatwierdzania (zatwierdzenie)

1. KG odbiera potwierdzenia od uczestników
 - PREPARED
 - READ-ONLY (brak modyfikacji)
 - ABORT (niemożliwość przygotowania do zatwierdzania)
2. Jeśli wszyscy odpowiedzieli PREPARED -> KG wysyła żądanie zatwierdzenia transakcji do węzła zatwierdzania
3. WZ zatwierdza transakcję i wysyła komunikat do KG
4. KG wysyła żądanie zatwierdzenia do pozostałych węzłów

ZSBD – laboratorium 1 (14)

W fazie zatwierdzania są realizowane następujące kroki prowadzące do zatwierdzenia transakcji.

1. Koordynator globalny odbiera potwierdzenia od uczestników.
2. Jeśli wszyscy odpowiedzieli komunikatem PREPARED, wówczas KG wysyła do węzła zatwierdzania komunikat COMMIT, czyli żądanie zatwierdzenia transakcji rozproszonej.
3. Węzeł zatwierdzania zatwierdza transakcję i wysyła do KG komunikat potwierdzający zatwierdzenie (COMMITTED).
4. Po otrzymaniu od WZ komunikatu COMMITTED, KG wysyła komunikat COMMIT do pozostałych węzłów (uczestników).



2PC - faza zatwierdzania (wycofanie)

1. KG odbiera potwierdzenia od uczestników
 - PREPARED
 - READ-ONLY (brak modyfikacji)
 - ABORT (niemożliwość przygotowania do zatwierdzania)
2. Jeśli choć jeden uczestnik odpowiedział ABORT -> KG wysłała żądanie wycofania transakcji do WZ
3. WZ wycofuje transakcję i wysłała komunikat do KG
4. KG wysłała żądanie wycofania do pozostałych węzłów

ZSBD – laboratorium 1 (15)

W fazie zatwierdzania są realizowane następujące kroki prowadzące do wycofania transakcji.

1. Koordynator globalny odbiera potwierdzenia od uczestników.
2. Jeśli przynajmniej jeden węzeł odpowiedział komunikatem ABORT (niemożliwe przygotowanie do zatwierdzenia), wówczas KG wysłała do węzła zatwierdzania komunikat ABORT, czyli żądanie wycofania transakcji rozproszonej.
3. Węzeł zatwierdzania wycofuje transakcję i wysłała do KG komunikat potwierdzający wycofanie (ABORTED).
4. Po otrzymaniu od WZ komunikatu ABORTED, KG wysłała komunikat ABORT do pozostałych węzłów (uczestników).



Faza zatwierdzania (uczestnik)

1. Odbiór od koordynatora globalnego komunikatu żądającego zatwierdzenia transakcji
2. Zatwierdzenie lokalnej transakcji
3. Zwolnienie blokad
4. Zapis informacji o zatwierdzeniu w pliku dziennika powtórzeń

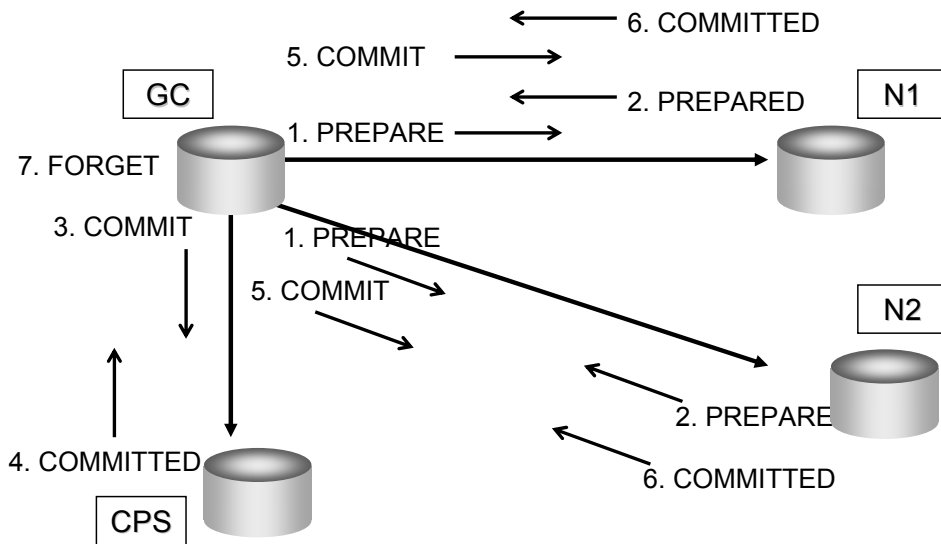


Faza zakończenia

1. Usunięcie z systemu informacji o zakończonej transakcji rozproszonej
2. Zwolnienie wszystkich niezwolnionych jeszcze zasobów systemowych



2PC - podsumowanie



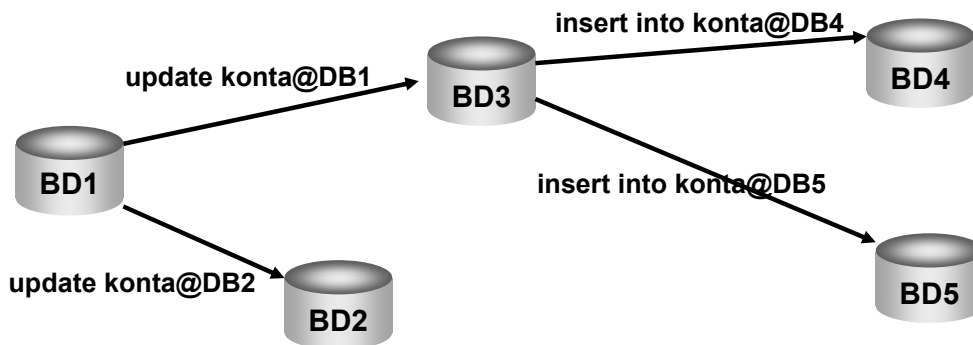
ZSBD – laboratorium 1 (18)

Na slajdzie przedstawiono komunikaty wysyłane pomiędzy węzłami w czasie zatwierdzania transakcji rozproszonej protokołem 2PC.



Graf wywołań transakcji

- Węzły - bazy danych adresowane przez transakcję rozproszoną
- Łuki - żądania wykonania poleceń w węzłach



ZSBD – laboratorium 1 (19)

Transakcja rozproszona adresuje wiele baz danych umieszczonych w różnych węzłach sieci. Jak wspomniano, transakcja taka jest reprezentowana przez transakcje lokalne w każdym z adresowanych węzłów. Oznacza to, że jedna transakcja lokalna jest powiązana z inną, znajdującą się w odległym węźle. Transakcje te tworzą tzw. **graf wywołań transakcji**, zwany również **drzewem sesji** (ang. *session tree*).

Przykładowy graf wywołań transakcji przedstawiono na slajdzie. Składa się on z 5 węzłów, baz danych BD1 do BD5. Łuki stanowią żądania wykonania poleceń z jednej bazy w innej.



Problemy sprzętowo-programowe

- W czasie fazy COMMIT (ROLLBACK) następuje awaria sieci, węzła lub zdalnej bazy danych
 - nie wszystkie węzły zatwierdziły (wycofały)
 - nie wszystkie węzły potwierdziły zakończenie operacji
 - transakcja rozproszona w stanie “in-doubt”
- Automatyczne odtwarzanie transakcji rozproszonej (proces RECO) w stanie “in-doubt” po usunięciu awarii
 - wynik: wszystkie węzły zatwierdzą lub wszystkie wycofają

ZSBD – laboratorium 1 (20)

Jeżeli awaria któregoś z węzłów nastąpi w czasie realizowania transakcji rozproszonej, wówczas taka transakcja będzie w tzw. **stanie zawieszenia** (ang. *in-doubt*) do momentu usunięcia awarii, nawiązania połączenia ze zdalnym węzłem i doprowadzenia transakcji do końca. Doprowadzenie do końca transakcji rozproszonej będącej w stanie zawieszenia będziemy nazywać **odtworzeniem transakcji**.

Transakcję w stanie zawieszenia można odtworzyć na dwa sposoby:
automatycznie - uruchamiając dedykowany do tego celu proces systemowy o nazwie RECO, lub "ręcznie" - przeszukując węzły i zatwierdzając lub wycofując ich lokalne transakcje.



Blokowanie przez transakcję rozproszoną

- Transakcja rozproszona w stanie “in-doubt” blokuje dane
- Inna transakcja żąda blokady na tych danych
 - ORA-01591: lock held by in-doubt distributed transaction <id>
 - polecenie żądające blokady jest wycofywane i może być powtórzone



"Ręczne" odtwarzanie transakcji w stanie "in-doubt"

- Stosowane gdy
 - blokowane dane muszą być natychmiast zwolnione
 - czas usunięcia awarii sprzętowej bardzo długi
 - nie działa proces automatycznego odtwarzania (RECO)

Transakcję w stanie "in-doubt" należy odtworzyć (zatwierdzić lub wycofać) "ręcznie" w przypadku gdy:

- dane zablokowane przez taką transakcję muszą być natychmiast zwolnione,
- czas usunięcia awarii sprzętowej i możliwość uruchomienia procesu automatycznego odtwarzania transakcji są bardzo długie,
- nie działa proces automatycznego odtwarzania.



Procedura "ręcznego" odtwarzania transakcji (1)

- Założenia
 - transakcja rozproszona uległa awarii (stan "in-doubt")
 - część danych w zdalnych bazach jest blokowana
 - użytkownik lokalny jednej ze zdalnych baz otrzymuje następujący komunikat w wyniku wykonania polecenia INSERT, UPDATE, DELETE lub SELECT

ORA-01591: lock held by in-doubt distributed transaction 1.21.17

↑
identyfikator lokalnej transakcji będącej częścią trans. rozproszonej

ZSBD – laboratorium 1 (23)

Na kolejnych slajdach zostanie omówiona procedura "ręcznego" odtwarzania transakcji w stanie "in-doubt". Zadaniem tej procedury jest odblokowanie danych, na których taka transakcja operuje. Odblokowanie danych nastąpi po zatwierdzeniu lub wycofaniu transakcji.

W celu zachowania atomowości transakcji rozproszonej należy stwierdzić, czy transakcję w stanie "in-doubt" należy wycofać, czy zatwierdzić. W tym celu należy znaleźć węzeł zatwierdzania i odczytać status transakcji w tym węźle.

Przyjmijmy, że użytkownik lokalnej bazy danych wykonując polecenie DML lub SELECT otrzymał komunikat:

ORA-01591: lock held by in-doubt distributed transaction 1.21.17

1.21.17 jest identyfikatorem transakcji lokalnej w stanie "in-doubt". Transakcja ta wchodzi w skład transakcji rozproszonej i blokuje dane.



Procedura "ręcznego" odtwarzania transakcji (2)

- Znalezienie opisu transakcji o identyfikatorze 1.21.17
- Perspektywy słownikowe
 - SYS.DBA_2PC_PENDING
 - SYS.DBA_2PC_NEIGHBORS
- Analiza zawartości DBA_2PC_PENDING w bazie lokalnej

```
SELECT * FROM sys.dba_2pc_pending  
WHERE local_tran_id = '1.21.17';
```

ZSBD – laboratorium 1 (24)

Informacje na temat transakcji rozproszonych w stanie "in-doubt", znajdujących się w systemie są dostępne za pomocą dwóch perspektyw słownika bazy danych: *DBA_2PC_PENDING* i *DBA_2PC_NEIGHBORS*.

Obie perspektywy znajdują się w schemacie użytkownika *SYS*. Każdy inny użytkownik bazy danych musi otrzymać od użytkownika *SYS* uprawnienie obiektowe *SELECT* to tych perspektyw, w celu odczytywania z nich informacji.

Na slajdzie przedstawiono przykładowe zapytanie do perspektywy *DBA_2PC_PENDING* w celu wyświetlenia opisu transakcji o identyfikatorze 1.21.17



Procedura "ręcznego" odtwarzania transakcji (3)

	nazwa globalna bd koordynatora	identyfikator bd koordynatora	
LOCAL_TRAN_ID	1.21.17		
GLOBAL_TRAN_ID	PO1.CIO1.POZ.PL	55d1c563	1.93.29 ← identyfikator lokalnej transakcji w bd koordynatora
STATE	prepared		
MIXED	no		
ADVICE			
TRAN_COMMENT	Sales/New Order/Trans_type	10B	
FAIL_TIME	31-MAY-91		
FORCE_TIME			
RETRY_TIME	31-MAY-91		
OS_USER	SWILLIAMS		
OS_TERMINAL	TWA139:		
HOST	system1		
DB_USER	SWILLIAMS		
COMMIT#			

identyczne wartości wystąpią tylko w bd koordynatora

ZSBD – laboratorium 1 (25)

Perspektywa DBA_2PC_PENDING jest wykorzystywana m.in. do uzyskania informacji nt. stanu transakcji rozproszonej oraz identyfikatorów transakcji lokalnej i rozproszonej. Szczegółowe znaczenie atrybutów tej perspektywy przedstawiono poniżej.

LOCAL_TRAN_ID - jest identyfikatorem transakcji lokalnej wchodzącej w skład transakcji rozproszonej;

GLOBAL_TRAN_ID - jest identyfikatorem transakcji rozproszonej; dla wszystkich transakcji lokalnych wchodzących w skład tej samej transakcji rozproszonej wartość tego atrybutu jest identyczna;

STATE - określa fazę przetwarzania (por. następny slajd);

MIXED - jeżeli przyjmuje wartość YES, oznacza to, że w niektórych węzłach transakcje lokalne (wchodzące w skład rozproszonej w zawieszeniu) zostały zatwierdzone manualnie, a w niektórych — wycofane;

ADVICE - zawiera wartość wskazówki przekazaną poleceniem *alter session advise*;



Procedura "ręcznego" odtwarzania transakcji (4)

LOCAL_TRAN_ID	1.21.17
GLOBAL_TRAN_ID	PO1.CIO1.POZ.PL.55d1c563.1.93.29
STATE	prepared
MIXED	no
ADVICE	
TRAN_COMMENT	Sales/New Order/Trans_type 10B
FAIL_TIME	31-MAY-91
FORCE_TIME	
RETRY_TIME	31-MAY-91
OS_USER	SWILLIAMS
OS_TERMINAL	TWA139:
HOST	system1
DB_USER	SWILLIAMS
COMMIT#	

ZSBD – laboratorium 1 (26)

TRAN_COMMENT - tekst wyspecyfikowany w poleceniu *commit comment*, lub *set transaction name*; jeżeli dla danej transakcji wyspecyfikowano oba polecenia, wówczas wartością tego atrybutu jest tekst z *set transaction name*;

FAIL_TIME - data i czas awarii transakcji;

FORCE_TIME - data i czas manualnego zatwierdzenia lub wycofania transakcji;

RETRY_TIME - data i czas ostatniej próby nawiązania połączenia z uszkodzonym węzłem;

OS_USER - nazwa użytkownika systemu operacyjnego, z którego konta uruchomiono aplikację korzystającą z bazy danych;

OS_TERMINAL - nazwa terminalu systemu operacyjnego, z którego uruchomiono aplikację korzystającą z bazy danych;

HOST - nazwa komputera, z którego uruchomiono aplikację korzystającą z bazy danych;

DB_USER - nazwa użytkownika bazy danych realizującego transakcję;

COMMIT# - globalny identyfikator zatwierdzenia transakcji nadawany przez system w momencie wydania polecenia *commit*.



Procedura "ręcznego" odtwarzania transakcji (5)

- Atrybut STATE może przyjąć wartość:
 - collecting
 - prepared
 - committed
 - forced commit
 - forced abort

ZSBD – laboratorium 1 (27)

Jednym z ważniejszych atrybutów perspektywy DBA_2PC_PENDING jest STATE. Jego wartość określa fazę przetwarzania, w której znajduje się transakcja lokalna; atrybut ten może przyjąć jedną z pięciu wartości:

- collecting — wartość ta pojawia się wyłącznie w węzłach koordynatora globalnego lub lokalnego; oznacza ona oczekiwanie na komunikaty od uczestników w fazie przygotowania;
- prepared — węzeł zakończył przygotowanie do zatwierdzenia swojej transakcji; informacja o tym została przesłana lub nie do koordynatora globalnego;
- committed — węzeł zatwierdził swoją transakcję lokalną;
- forced commit — transakcja w węźle, poprzednio w stanie zawieszenia, została manualnie zatwierdzona;
- forced abort — transakcja w węźle, poprzednio w stanie zawieszenia, została manualnie wycofana;



Procedura "ręcznego" odtwarzania transakcji (6)

- Analiza zawartości DBA_2PC_NEIGHBORS w bazie lokalnej

```
SELECT * FROM sys.dba_2pc_neighbors  
WHERE local_tran_id = '1.21.17';
```

LOCAL_TRAN_ID	1.21.17
IN_OUT	in
DATABASE	PO1.CIO1.POZ.PL
DBUSER_OWNER	SCOTT
INTERFACE	N
DBID	000003F4
SESS#	1
BRANCH	0100

ZSBD – laboratorium 1 (28)

Perspektywa DBA_2PC_NEIGHBORS przechowuje graf wywołań transakcji. Jej zawartość jest wykorzystywana do odnajdowania poszczególnych węzłów, m.in. zatwierdzania, biorących udział w transakcji. Szczegółowe znaczenie atrybutów perspektywy DBA_2PC_NEIGHBORS przedstawiono na następnym slajdzie.



Procedura "ręcznego" odtwarzania transakcji (7)

LOCAL_TRAN_ID	1.21.17	węzeł jest serwerem żądania bazy danych
IN_OUT	in	
DATABASE	PO1.CIO1.POZ.PL	dołączenie zrealizowane z konta
DBUSER_OWNER	SCOTT	
INTERFACE	N	WO2.CIO2.WAR.PL nie jest węzłem zatwierdzania; żaden z węzłów podległych nie jest węzłem zatwierdzania
DBID	000003F4	
SESS#	1	
BRANCH	0100	

ZSBD – laboratorium 1 (29)

LOCAL_TRAN_ID - identyfikator transakcji lokalnej w danym węźle; transakcja ta wchodzi w skład transakcji rozproszonej, której identyfikator można z kolei odczytać z DBA_2CP_PENDING.GLOBAL_TRAN_ID;

IN_OUT - wskazuje kierunek żądania; wartość in oznacza żądanie kierowane (wchodzące) do bieżącego węzła, a wartość out — żądanie kierowane (wychodzące) z bieżącego węzła do zdalnego;

DATABASE - dla żądania wchodzącego określa nazwę bazy danych kierującej żądanie, natomiast dla żądania wychodzącego określa nazwę łącznika bazy danych, za pomocą którego jest kierowane żądanie do zdalnej bazy;

DBUSER_OWNER - dla żądania przychodzącego określa nazwę użytkownika wyspecyfikowanego w łączniku bazy danych wykorzystywanym do skierowania żądania, natomiast dla żądania wychodzącego określa nazwę właściciela łącznika, za pomocą którego jest kierowane żądanie do zdalnej bazy;



Procedura "ręcznego" odtwarzania transakcji (8)

LOCAL_TRAN_ID	1.21.17
IN_OUT	in
DATABASE	PO1.CIO1.POZ.PL
DBUSER_OWNER	SCOTT
INTERFACE	N
DBID	000003F4
SESS#	1
BRANCH	0100

ZSBD – laboratorium 1 (30)

INTERFACE - wartość C oznacza, że bieżący węzeł jest węzłem zatwierdzania lub któryś z węzłów mu podległych jest węzłem zatwierdzania; wartość N oznacza, że ani węzeł bieżący, ani żaden z mu podległych nie są węzłami zatwierdzania;

DBID - systemowy identyfikator bazy danych;

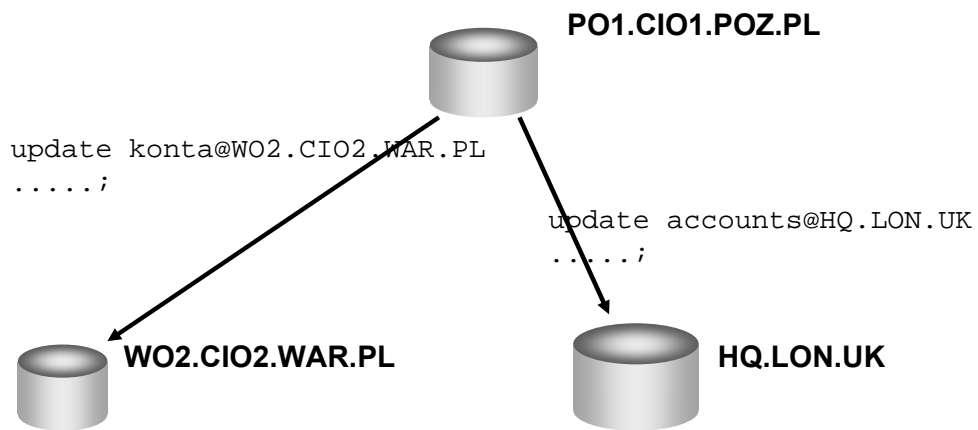
SESS# - systemowy identyfikator sesji w bieżącej bazie danych;

BRANCH - systemowy identyfikator bieżącego węzła.



Studium przypadku (1)

- Graf wywołań transakcji



Studium przypadku zostanie omówione dla grafu wywołań transakcji przedstawionego na slajdzie. Koordynatorem globalnym jest baza danych PO1.CIO1.POZ.PL.



Studium przypadku (2)

WO2.CIO2.WAR.PL

ORA-01591: lock held by in-doubt distributed transaction 1.21.17

```
SELECT * FROM sys.dba_2pc_neighbors
WHERE local_tran_id = '1.21.17';
```

LOCAL_TRAN_ID	1.21.17	węzeł jest serwerem
IN_OUT	in	żądania bazy danych
DATABASE	PO1.CIO1.POZ.PL	
DBUSER_OWNER	SCOTT	dołączenie
INTERFACE	N	zrealizowane z konta
DBID	000003F4	
SESS#	1	
BRANCH	0100	

WO2.CIO2.WAR.PL nie jest węzłem zatwierdzania;
żaden z węzłów podległych
nie jest węzłem
zatwierdzania

ZSBD – laboratorium 1 (32)

Przyjmijmy, że w bazie danych WO2.CIO2.WAR.PL jest aktywna transakcja w stanie "in-doubt" o identyfikatorze 1.21.17.

Kolejne kroki będą prowadziły do znalezienia węzła zatwierdzania. W tym celu wyświetlamy fragment grafu wywołań transakcji w węźle WO2.CIO2.WAR.PL, dla transakcji o identyfikatorze 1.21.17. Służy do tego celu polecenie SELECT ze slajdu.

Wynikiem tego polecenia jest fragment grafu, pokazany na slajdzie. Wynika z niego, że do węzła WO2.CIO2.WAR.PL jest kierowane żądanie (IN_OUT='in') z bazy PO1.CIO1.POZ.PL. Wartość atrybutu INTERFACE='N' oznacza, że WO2.CIO2.WAR.PL nie jest węzłem zatwierdzania.

W następnym kroku należy zbadać stan transakcji w węźle który kieruje żądanie do WO2.CIO2.WAR.PL, czyli w węźle PO1.CIO1.POZ.PL.



Studium przypadku (3)

WO2.CIO2.WAR.PL

Znalezienie identyfikatora transakcji globalnej w węźle WO2.CIO2.WAR.PL na podstawie identyfikatora transakcji lokalnej

```
SELECT local_tran_id, global_tran_id
FROM sys.dba_2pc_pending
WHERE local_tran_id = '1.21.17';
```

LOCAL_TRAN_ID	GLOBAL_TRAN_ID
1.21.17	PO1.CIO1.POZ.PL.55d1c563.1.93.29

ZSBD – laboratorium 1 (33)

Zanim przejdziemy do węzła PO1.CIO1.POZ.PL należy znaleźć identyfikator transakcji globalnej w skład którego wchodzi lokalna o identyfikatorze 1.21.17, czyli transakcja w stanie "in-doubt". W tym celu korzystamy z zapytania przedstawionego na slajdzie.



Studium przypadku (4)

PO1.CIO1.POZ.PL

Znalezienie identyfikatora transakcji lokalnej w węźle
PO1.CIO1.POZ.PL na podstawie id transakcji globalnej

```
SELECT local_tran_id FROM sys.dba_2pc_pending
WHERE lobal_tran_id='PO1.CIO1.POZ.PL.55d1c563.1.93.29';
```

LOCAL_TRAN_ID	GLOBAL_TRAN_ID
1.93.29	PO1.CIO1.POZ.PL.55d1c563.1.93.29

Wyświetlenie zawartości SYS.DBA_2PC_NEIGHBORS

```
SELECT * FROM dba_2pc_neighbors
WHERE local_tran_id='1.93.29';
```

ZSBD – laboratorium 1 (34)

Znając identyfikator transakcji globalnej, w węźle PO1.CIO1.POZ.PL odczytujemy identyfikator transakcji lokalnej, wchodzącej w skład transakcji rozproszonej. Następnie, z wykorzystaniem identyfikatora transakcji lokalnej odczytujemy fragment grafu wywołań transakcji.



Studium przypadku (5)

		PO1.CIO1.POZ.PL
LOCAL_TRAN_ID	1.93.29	
IN_OUT	OUT	← węzeł zgłasza żądanie do serwera
DATABASE	WO2.CIO2.WAR.PL	←
DBUSER_OWNER	SWILLIAMS	
INTERFACE	N	← nie jest węzłem zatwierdzania
DBID	55d1c563	
SESS#	1	
BRANCH	1	
LOCAL_TRAN_ID	1.93.29	
IN_OUT	OUT	← węzeł zgłasza żądanie do serwera
DATABASE	HQ.LON.UK	←
DBUSER_OWNER	ALLEN	
INTERFACE	C	← HQ.LON.UK jest węzłem zatwierdzania
DBID	00000390	
SESS#	1	
BRANCH	1	

ZSBD – laboratorium 1 (35)

Z otrzymanego grafu wywołań transakcji wynika, że z węzła PO1.CIO1.POZ.PL wychodzą 2 żądania, jedno do węzła WO2.CIO2.WAR.PL (odwiedzanego już wcześniej) i jedno do węzła HQ.LON.UK. Atrybut INTERFACE='C' występuje przy drugim żądaniu, więc w bazie HQ.LON.UK (lub bazie jej podległej) należy poszukiwać węzła zatwierdzania.

Udajemy się zatem do bazy HQ.LON.UK.



Studium przypadku (6)

HQ.LON.UK

Odczytanie statusu transakcji w węźle zatwierdzania

```
SELECT local_tran_id, global_tran_id, state, commit#
FROM sys.dba_2pc_pending
WHERE global_tran_id =
      'PO1.CIO1.POZ.PL.55d1c563.1.93.29';
```

LOCAL_TRAN_ID	1.45.13
GLOBAL_TRAN_ID	'PO1.CIO1.POZ.PL.55d1c563.1.93.29'
STATE	COMMIT
COMMIT#	129314

Należy zatwierdzić transakcje lokalne we wszystkich węzłach

ZSBD – laboratorium 1 (36)

Korzystając z identyfikatora transakcji globalnej odczytujemy z perspektywy SYS.DBA_2PC_PENDING status transakcji. Ponieważ wartość atrybutu STATE wynosi 'COMMIT', więc został znaleziony węzeł zatwierdzania. Przypomnijmy, że STATE='COMMIT' wystąpi wyłącznie w węźle zatwierdzania. Ponieważ węzeł zatwierdzania zatwierdził swoją transakcję lokalną, więc w celu zapewnienia atomowości transakcji globalnej, należy zatwierdzić niezatwierdzone transakcje lokalne we wszystkich węzłach uczestniczących w transakcji rozproszonej.



Studium przypadku (7)

Zatwierdzenie transakcji

```
COMMIT FORCE  
'identyfikator.transakcji.lokalnej';
```

```
COMMIT FORCE '1.21.17';
```

Wycofanie transakcji

```
ROLLBACK FORCE  
'identyfikator.transakcji.lokalnej';
```

Uprawnienia systemowe FORCE ANY TRANSACTION

ZSBD – laboratorium 1 (37)

Manualne zatwierdzanie i wycofywanie transakcji lokalnej realizuje się odpowiednio poleceniami:

```
commit force 'id_trans_lokalnej';
```

```
rollback force 'id_trans_lokalnej';
```

W naszym przypadku należy zatwierdzić transakcję o identyfikatorze 1.21.17.

Użytkownik, który symuluje awarie transakcji rozproszonych musi posiadać uprawnienie FORCE ANY TRANSACTION. Umożliwia ono również zatwierdzanie lub wycofywanie dowolnych transakcji w systemie poleceniami *commit force* i *rollback force* oraz wydawanie polecenia *commit comment*.



Programowe symulowanie awarii (1)

- Symulacja 10 typów awarii transakcji rozproszonej

- 1 : Crash commit point site after collect
- 2 : Crash non-commit point site after collect
- 3 : Crash before prepare (non-commit point site)
- 4 : Crash after prepare (non-commit point site)
- 5 : Crash commit point site before commit
- 6 : Crash commit point site after commit
- 7 : Crash non-commit point site before commit
- 8 : Crash non-commit point site after commit
- 9 : Crash commit point site before forget
- 10: Crash non-commit point site before forget

ZSBD – laboratorium 1 (38)

Producent SZBD Oracle dostarcza mechanizm programowego symulowania awarii transakcji rozproszonych. Administrator ma do dyspozycji 10 różnego typu awarii transakcji rozproszonych (ang. crash tests). Ich opis umieszczono poniżej.

- 1 - Awaria węzła zatwierdzania po odebraniu przez koordynator globalny komunikatów prepared.
- 2 - Awaria węzła uczestnika po odebraniu przez koordynator globalny komunikatów prepared.
- 3 - Awaria węzła uczestnika przed przygotowaniem się do zatwierdzania.
- 4 - Awaria węzła uczestnika po przygotowaniu się do zatwierdzania.
- 5 - Awaria węzła zatwierdzania przed zatwierdzeniem transakcji.
- 6 - Awaria węzła zatwierdzania po zatwierdzeniu transakcji.
- 7 - Awaria węzła uczestnika przed zatwierdzeniem transakcji.
- 8 - Awaria węzła uczestnika po zatwierdzeniu transakcji.
- 9 - Awaria węzła zatwierdzania przed przejściem do fazy zakończenia.
- 10 - Awaria węzła uczestnika przed przejściem do fazy zakończenia.



Programowe symulowanie awarii (2)

- Ogólna składnia polecenia

```
COMMIT COMMENT 'ORA-2PC-CRASH-TEST-n' ;
```

- Przykład: symulowanie awarii nr 5

```
COMMIT COMMENT 'ORA-2PC-CRASH-TEST-5' ;
```

Awarie symuluje się wydając polecenie:

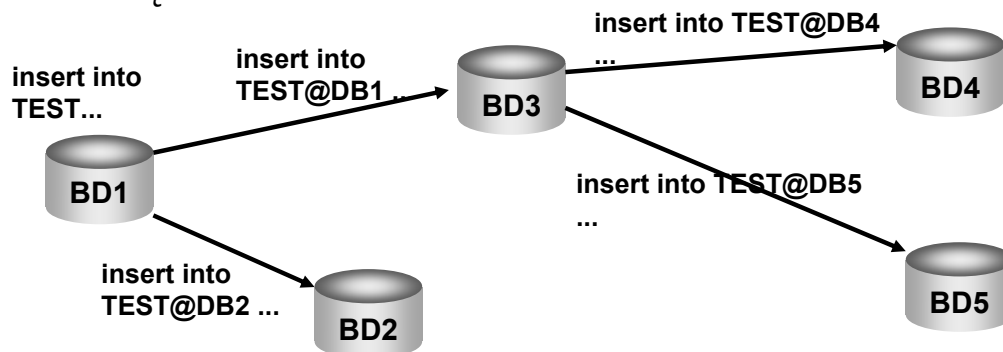
```
commit comment 'ORA-2PC-CRASH-TEST-nr';
```

gdzie nr jest numerem testu opisanym na poprzednim slajdzie. Tekst '*ORA-2PC-CRASH-TEST*' musi być pisany dużymi literami. Przykładowo, polecenie ze slajdu symuluje awarię numer 2.



Ćwiczenie (1)

1. Zasymulować awarię numer 6 transakcji, której graf wywołań przedstawiono poniżej
2. Zakończyć transakcję rozproszoną zgodnie z informacją w węźle zatwierdzenia



ZSBD – laboratorium 1 (40)

Celem ćwiczenia jest symulowanie awarii transakcji rozproszonej i jej manualne doprowadzenie do końca.

W ramach ćwiczenia zostanie zasymulowana awaria nr 6.



Ćwiczenie (2)

- Przygotowanie środowiska
 - w każdej z baz danych BD1 do BD5 utworzyć tabelę TEST z jednym atrybutem typu numerycznego
 - w BD3 utworzyć wyzwalacz AFTER INSERT na tabeli TEST; wyzwalacz ten ma propagować rekordy wstawione do tabel TEST w bazach DB4 i BD5

Przed rozpoczęciem ćwiczenia należy przygotować środowisko zgodnie z opisem na slajdzie.



Ćwiczenie (3)

- Wydać poniższe polecenia z bazy DB1

```
insert into TEST values (1);  
insert into TEST@BD2 values (2);  
insert into TEST@BD3 values (3);  
commit comment 'ORA-2PC-CRASH-TEST-6';
```

- Z niezależnych sesji (lokalnych dla baz DB1 do DB5) wydać polecenie SELECT * do tabeli TEST
- Po uzyskaniu błędu "ORA-01591: lock held by in-doubt distributed transaction ..." rozpocząć procedurę "ręcznego" odtwarzania transakcji