



Integracja systemów transakcyjnych

Robert Wrembel
Politechnika Poznańska
Instytut Informatyki
Robert.Wrembel@cs.put.poznan.pl
www.cs.put.poznan.pl/rwrembel

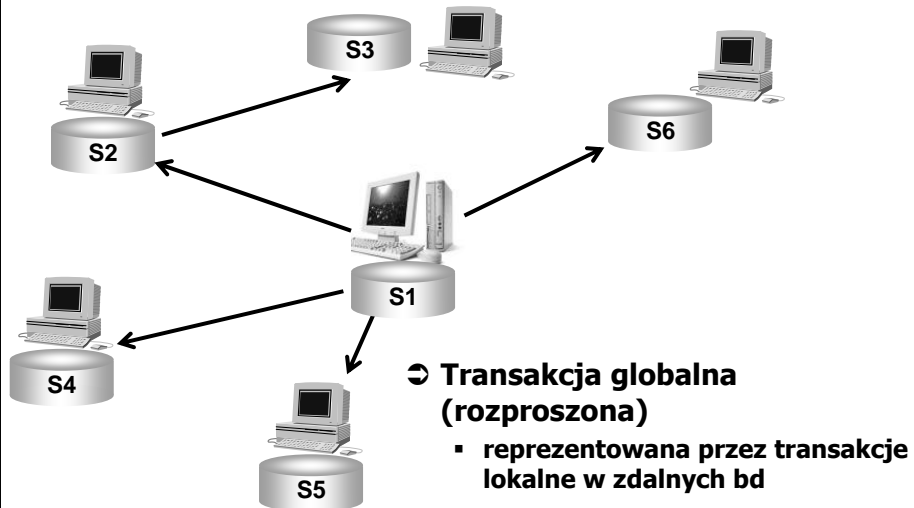


Zarządzanie transakcjami rozproszonymi

- ⇒ Protokół 2PC
- ⇒ Protokół 3PC
- ⇒ Protokół 2PC w implementacji Oracle



Transakcje w środowisku RSBD



Awarie w środowisku RSBD

- ↻ Zgubienie komunikatu
- ↻ Awaria medium
- ↻ Awaria węzła
- ↻ Brak łączności pomiędzy S1 a S2
 - S2 uległ awarii
 - medium komunikacyjne uległo awarii
 - S2 jest zbyt obciążone aby odpowiedzieć



Wymagania dot. przetwarzania

- **Atomowość**
- **Spójność**
- **Odtwarzanie po awarii**
- **Awaria węzła nie powinna wpływać na (blokować) inny węzeł**
- **Zaawansowany protokół zatwierdzania transakcji rozproszonej**
 - **Two-Phase Commit (2PC)**
 - **Trhee-Phase Commit (3PC)**
 - **Aktorzy**
 - **koordynator**
 - **uczestnik**
 - **Każdy węzeł posiada własny log transakcyjny**



2PC

- **Faza głosowania**
- **Faza decyzji**



2PC - koordynator

⇒ Faza głosowania

1. zapis do logu (na dysku) `begin_commit`; wysłanie komunikatu `PREPARE` do wszystkich uczestników; oczekiwanie na odpowiedź przez zadany czas (timeout)

⇒ Faza decyzji

2. jeśli odebrano komunikat `READY_COMMIT`, następuje uaktualnienie listy uczestników, którzy odpowiedzieli; jeśli wszyscy odpowiedzieli `READY_COMMIT` następuje zapis do logu (na dysku) `commit`; wysłanie komunikatu `GLOBAL_COMMIT` do wszystkich uczestników; oczekiwanie na odpowiedź przez zadany czas (timeout)
3. jeśli odebrano od wszystkich potwierdzenia zatwierdzenia transakcji następuje zapis do logu `end_of_transaction`; jeśli węzeł nie odpowiedział następuje ponowne wysłanie decyzji do momentu otrzymania odpowiedzi



2PC - koordynator

⇒ Faza decyzji

4. jeśli odebrano komunikat `ABORT` następuje zapis do logu (na dysku) `abort`; wysłanie komunikatu `GLOBAL_ABORT` do wszystkich uczestników; oczekiwanie na odpowiedź przez zadany czas (timeout)



2PC - uczestnik

➤ Faza decyzji

1. Otrzymanie komunikatu PREPARE

- zapis ready_commit do logu; zapis buforów na dysk; wysłanie READY_COMMIT do koordynatora
- lub
- zapis abort to do logu; wycofanie transakcji; wysłanie ABORT do koordynatora
- oczekiwanie na komunikat od koordynatora przez zadany czas (timeout)

2. Jeśli odebrano komunikat GLOBAL_COMMIT, następuje zapis commit do logu; zatwierdzenie transakcji i zwolnienie zasobów; wysłanie potwierdzenia do koordynatora

3. Jeśli odebrano GLOBAL_ABORT, następuje zapis abort do logu; wycofanie transakcji; zwolnienie zasobów; wysłanie potwierdzenia do koordynatora



2PC - podsumowanie

Koordynator

Uczestnik

- zapis begin_commit
- wysłanie PREPARE →
- oczekiwanie na odpowiedź ←
- wszyscy odpowiedzieli READY
 - zapis commit do logu
 - wysłanie GLOBAL_COMMIT →
 - oczekiwanie na potwierdzenie
- oczekiwanie na GLOBAL_COMMIT lub GLOBAL_ABORT
- zapis commit do logu
- zatwierdzenie transakcji
- ← ➤ wysłanie potwierdzenia
- wszyscy potwierdzili
 - zapis end_of_transaction do logu



2PC - podsumowanie

Koordynator

- ➔ zapis begin_commit
- ➔ wysłanie PREPARE →
- ➔ oczekiwanie na odpowiedź ←
- ➔ conajmniej 1 odpowiedzią ABORT
 - zapis abort do logu
 - wysłanie GLOBAL_ABORT →
 - oczekiwanie na potwierdzenie

Uczestnik

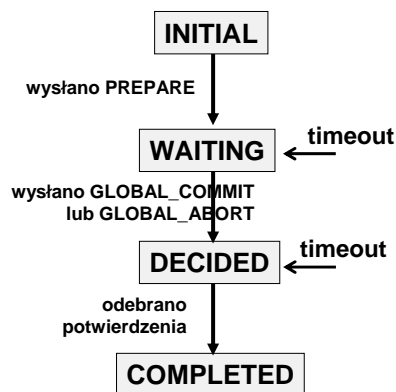
- ➔ zapis abort do logu
- ➔ wysłanie ABORT
- ➔ wycofanie transakcji

- ➔ wszyscy potwierdzili
 - zapis end_of_transaction do logu

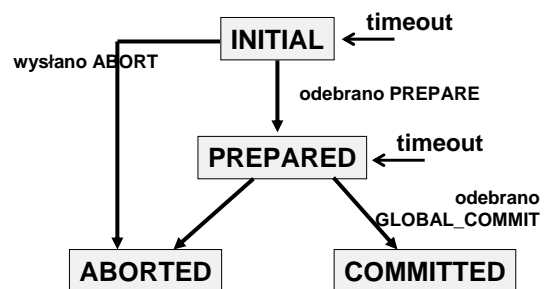


2PC - oczekiwanie

Koordynator



Uczestnik





Reakcja na timeout

- ➔ Brak komunikatu w określonym czasie - timeout
- ➔ Obsługa timeout za pomocą protokołu timeout (termination protocol)



Protokół timeout - koordynator

- ➔ W stanie **WAITING**
 - niemożliwa decyzja o zatwierdzeniu
 - możliwa decyzja o wycofaniu
- ➔ W stanie **DECIDED**
 - ponowne wysłanie decyzji do uczestników, którzy nie wysłali potwierdzenia





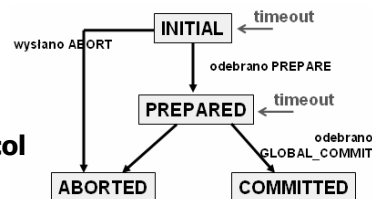
Protokół timeout - uczestnik

➔ W stanie INITIAL

- wycofanie transakcji
- jeśli pojawiłby się komunikat PREPARE po wycofaniu trans. uczestnik odpowiada ABORT

➔ W stanie PREPARED

- uczestnik przygotował się do zatwierdzenia ⇒ nie może już wycofać transakcji i nie może zatwierdzić ⇒ uczestnik blokuje dane
- zmniejszenie prawdopodobieństwa blokowania ⇒ zastosowanie cooperative termination protocol



Cooperative termination protocol

➔ Jeśli awarii ulega koordynator ⇒ uczestnicy wybierają nowego koordynatora

- konieczna znajomość wszystkich uczestników
- koordynator załącza listę uczestników w komunikatach
- nowy koordynator doprowadza transakcję rozproszoną do końca



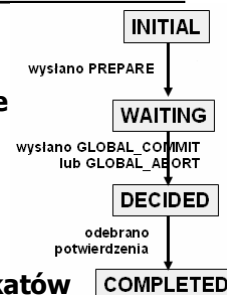
Reakcja na awarię

- ⇒ Węzeł ulega awarii
- ⇒ Po usunięciu awarii następuje odtwarzanie (recovery) stanu węzła
- ⇒ Protokół odtwarzania (recovery protocol)



Protokół odtwarzania - koordynator

- ⇒ W stanie **INITIAL**
 - procedura zatwierdzania nie została jeszcze uruchomiona ⇒ uruchom procedurę
- ⇒ W stanie **WAITING**
 - oczekiwanie na brakujące komunikaty **READY_COMMIT** lub **ABORT**
 - dalsza akcja zależy od odebranych komunikatów
- ⇒ W stanie **DECIDED**
 - decyzja została już podjęta, komunikaty wysłane
 - koordynator otrzymał wszystkie potwierdzenia ⇒ koniec pracy
 - brakuje potwierdzeń ⇒ odtwarzanie z wykorzystaniem protokołu timeout w fazie **DECIDED**





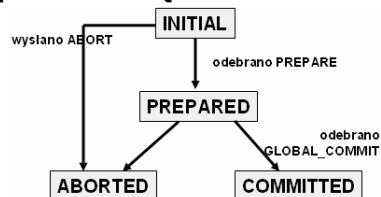
Protokół odtwarzania - uczestnik

- ⇒ Uczestnik po odtworzeniu wykonuje takie same akcje jak inni uczestnicy
- ⇒ Procedura odtwarzania nie wymaga kontaktów z innymi węzłami



Protokół odtwarzania - uczestnik

- ⇒ W fazie **INITIAL**
 - wycofanie transakcji ponieważ koordynator nie mógł podjąć innej decyzji niż wycofanie
- ⇒ W fazie **PREPARED**
 - uczestnik wysłał przed awarią komunikat ⇒ odtwarzanie z wykorzystaniem protokołu timeout w fazie **PREPARED**
- ⇒ W fazie **ABORTED/COMMITTED**
 - transakcja została zakończona przed awarią ⇒ nie wymaga działań





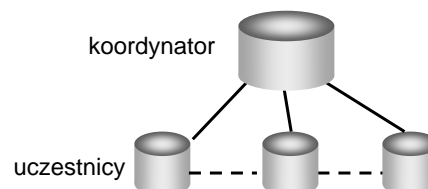
Topologia komunikacyjna 2PC

- ⇒ Odzwierciedla "ścieżkę" wymiany komunikatów pomiędzy koordynatorem i uczestnikami
- ⇒ Scentralizowana
- ⇒ Liniowa
- ⇒ Zdecentralizowana / rozproszona



Topologie komunikacyjne 2PC

- ⇒ Scentralizowana
 - koordynator zna adresy wszystkich uczestników
 - protokół wyboru nowego koordynatora ⇔ rozesłanie listy adresów w komunikatach do uczestników
 - musi istnieć możliwość komunikacji pomiędzy uczestnikami

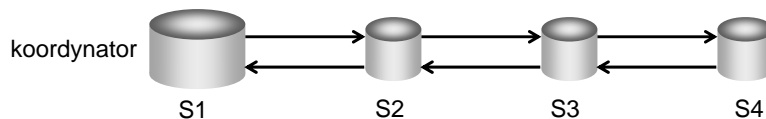




Topologie komunikacyjne 2PC

➤ Liniowa

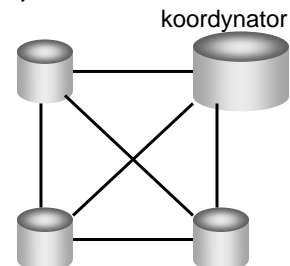
- węzły posiadają numery (1 - koordynator, ...), odzwierciedlające kolejność wysyłania komunikatów
- faza głosowania koordynator → uczestnik
- faza decyzji uczestnik → koordynator
- każdy uczestnik dodaje do komunikatu swoją odpowiedź ⇨ komunikat zwrotny zawiera decyzję globalną
- modyfikacja
 - faza głosowania: koordynator → uczestnik → uczestnik → ...
 - faza decyzji: uczestnik wysyła decyzję do wszystkich



Topologie komunikacyjne 2PC

➤ Zdecentralizowana/rozproszona

- koordynator rozsyła PREPARE do wszystkich
- uczestnik rozsyła odpowiedź do wszystkich
- uczestnik czeka z decyzją na odpowiedzi od wszystkich pozostałych uczestników ⇨ faza decyzji jest niepotrzebna
 - uczestnik może podjąć spójną decyzję sam, niezależnie od pozostałych uczestników





Podsumowanie 2PC

⇒ Protokół blokujący

- np. uczestnik odpowiada `ready_commit` ale nie otrzymuje odpowiedzi od koordynatora ⇒ blokowanie danych w węźle
- prawdopodobieństwo zablokowania jest w praktyce małe (2PC implementowany w praktyce)



3PC

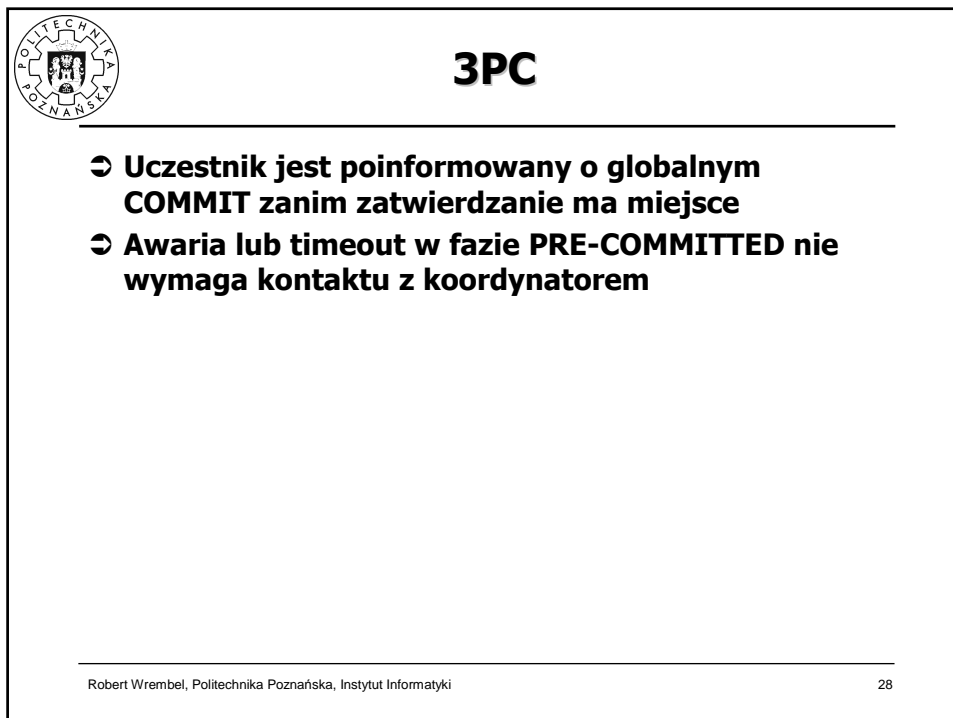
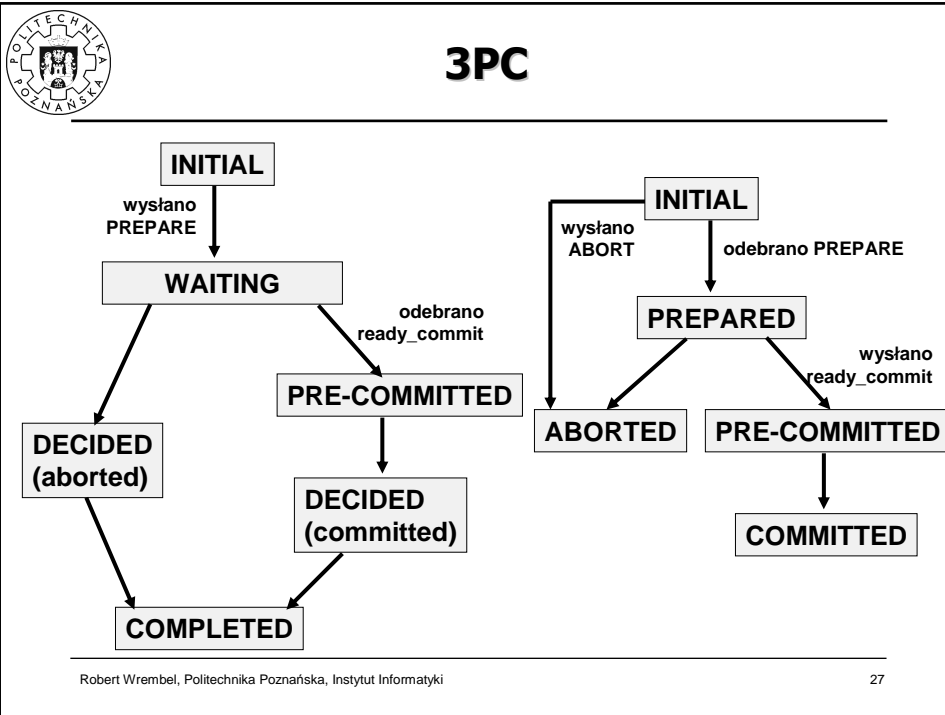
⇒ Protokół nieblokujący

- przynajmniej 1 węzeł jest sprawny

⇒ Minimalizacja czasu niepewności uczestnika, który odpowiedział `ready_commit` i oczekuje na `GLOBAL_COMMIT` lub `GLOBAL_ABORT`

⇒ Koordynator

- po odebraniu od wszystkich `ready_commit` rozsyła komunikat `PRE-COMMIT` do wszystkich uczestników
 - uczestnik odbierając `PRE_COMMIT` zna decyzję pozostałych
- po odebraniu potwierdzenia od uczestników `PRE-COMMIT`, koordynator rozsyła `COMMIT`
- `ABORT` obsługiwany identycznie jak w 2PC



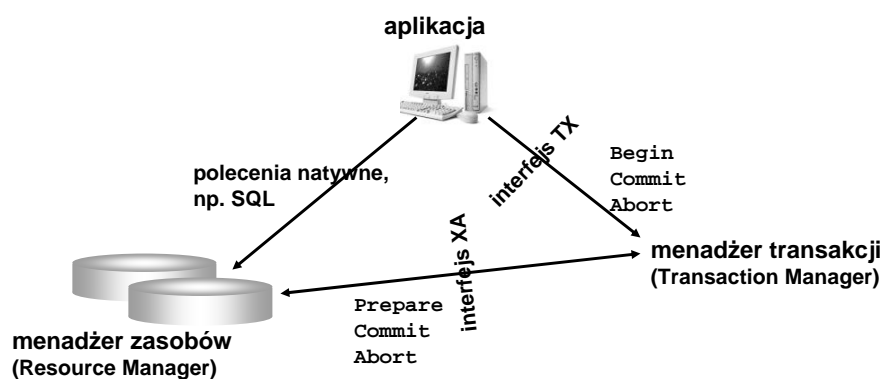


Środowisko heterogeniczne

- ⇒ Wsparcie dla 2PC/3PC
- ⇒ Różna implementacja 2PC/3PC
- ⇒ Konieczność współpracy systemów ⇨ standaryzacja interfejsu komunikacji
 - LU6.2 (IBM)
 - X/Open Distributed Transaction Processing (OSI-TP)



X/Open DTP



- ⇒ zadania TM
 - definiowanie zakresu transakcji (poleceń składających się na transakcję)
 - zarządzanie identyfikatorami transakcji
 - komunikacja z innymi TM



X/Open DTP

⇒ Application

- żądanie od TM rozpoczęcia transakcji rozproszonej
- wysłanie poleceń do RM
- żądanie od TM zakończenia transakcji

⇒ TM komunikuje się z RM w ramach transakcji



X/Open DTP

⇒ Interfejs TX: aplikacja → TM

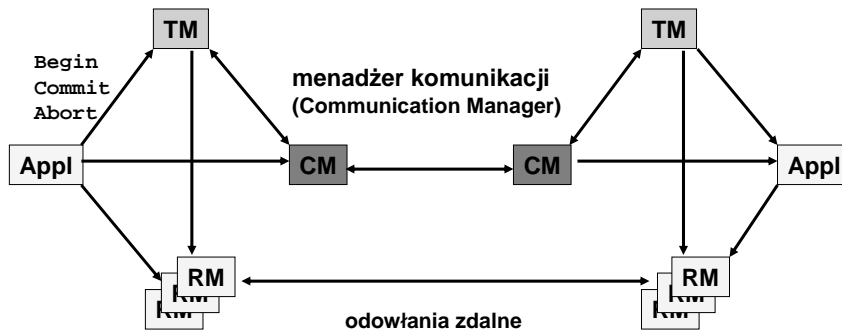
- tx_open, tx_close – otwarcie, zamknięcie sesji
- tx_begin – rozpoczęcie transakcji
- tx_commit, tx_abort – commit, abort

⇒ Interfejs XA: TM → RM

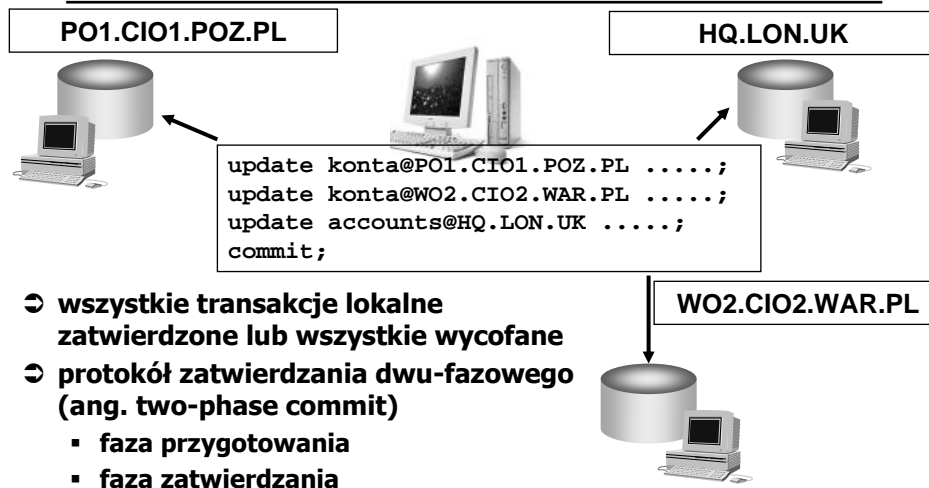
- xa_open, xa_close – dołączenie, odłączenie od RM
- xa_start, xa_end – rozpoczęcie, zakończenie transakcji o wskazanym ID
- xa_rollback – wycofanie transakcji o wskazanym ID
- xa_prepare – przygotowanie transakcji o wskazanym ID
- xa_commit – zatwierdzenie transakcji o wskazanym ID
- xa_forget – zwolnienie zasobów



X/Open DTP



Transakcja rozproszona





Aktorzy

- **Koordinator globalny**
 - węzeł, w którym zainicjowano transakcję rozproszoną
- **Uczestnik**
 - węzeł z transakcją lokalną
- **Węzeł zatwierdzania (ang. commit point site)**
 - inicjowanie zatwierdzania lub wycofywania transakcji zgodnie z komunikatem od koordynatora globalnego
 - wybierany przez administratora systemu
 - parametr konfiguracyjny COMMIT_POINT_STRENGTH
 - wartość 0-255; wartość domyślna zależna od systemu operacyjnego
 - odzwierciedla ilość danych krytycznych w węźle
 - odzwierciedla niezawodność węzła
 - węzeł o najwyższej wartości COMMIT_POINT_STRENGTH jest węzłem zatwierdzania



Aktorzy

- **Węzeł zatwierdzania**
 - zawiera status zatwierdzania transakcji rozproszonej
 - transakcja rozproszona jest uznawana za zatwierdzoną jeżeli zostanie zatwierdzona w węźle zatwierdzania, nawet jeśli pozostałe węzły jeszcze nie zatwierdziły swoich transakcji lokalnych



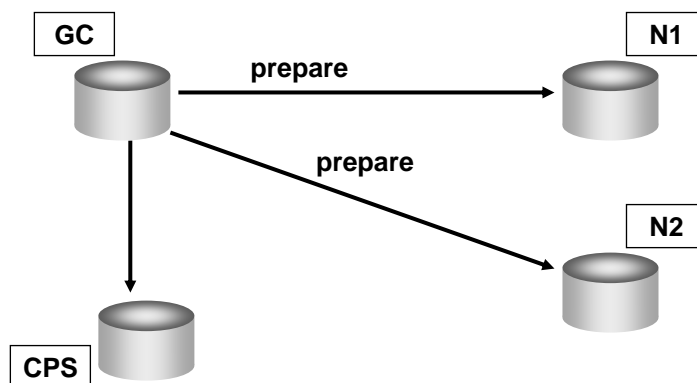
2PC

- Faza przygotowania (prepare)
- Faza zatwierdzania (commit)
- Faza zakończenia (forget)



2PC - faza przygotowania (koordynator)

- koordynator globalny określa węzeł zatwierdzania
- koordynator globalny wysyła do uczestników żądanie przygotowania do zatwierdzania





Faza przygotowania - uczestnik

- odbiór komunikatu od koordynatora globalnego żądającego przygotowania do zatwierdzenia
- wysłanie żądania przygotowania do zdalnych węzłów, do których odwołuje się uczestnik
- w przypadku braku modyfikacji danych -> wysłanie do koordynatora globalnego komunikatu READ-ONLY
- zapisanie zawartości bufora dziennika powtórzeń do pliku dziennika (redo log)
- inne zdalne węzły dołączone do uczestnika zgłosiły gotowość i sam uczestnik jest gotów -> wysłanie komunikatu PREPARED do koordynatora globalnego
 - w przeciwnym przypadku
 - wycofanie lokalnej transakcji
 - wysłanie ABORT



2PC - faza zatwierdzenia

- koordynator globalny odbiera potwierdzenia od uczestników
 - PREPARED
 - READ-ONLY (brak modyfikacji)
 - ABORT (niemożliwość przygotowania do zatwierdzenia)
- jeśli wszyscy odpowiedzieli PREPARED -> koordynator globalny wysyła żądanie zatwierdzenia transakcji do węzła zatwierdzenia
 - węzeł zatwierdzenia zatwierdza transakcję i wysyła komunikat do koordynatora globalnego
 - koordynator globalny wysyła żądanie zatwierdzenia do pozostałych węzłów



2PC - faza zatwierdzania

- ⇒ jeśli choć jeden uczestnik odpowiedział ABORT -> koordynator wysłał żądanie wycofania transakcji do węzła zatwierdzania
 - węzeł zatwierdzania wycofuje transakcję i wysłał komunikat do koordynatora globalnego
 - koordynator wysłał żądanie wycofania do uczestników

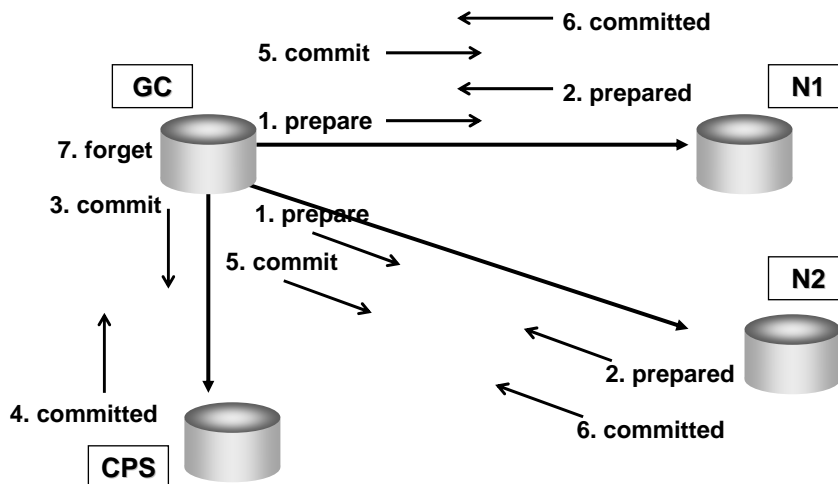


Faza zatwierdzania - uczestnik

- ⇒ odbiór od koordynatora globalnego komunikatu żądającego zatwierdzenia transakcji
- ⇒ zatwierdzenie lokalnej transakcji
- ⇒ zwolnienie blokad
- ⇒ zapis informacji o zatwierdzeniu w pliku dziennika powtórzeń



2PC - podsumowanie



Ograniczenie liczby transakcji rozproszonych

- ⇒ parametr konfiguracyjny **DISTRIBUTED_TRANSACTIONS**
- ⇒ **DISTRIBUTED_TRANSACTIONS=0**
 - brak transakcji rozproszonych
 - nie startuje proces drugoplanowy RECO



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



Blokowanie

- 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



Blokowanie

- ⇒ transakcja rozproszona żąda w zdalnym węźle blokady danych zablokowanych wcześniej przez inną transakcję
 - czas oczekiwania na zwolnienie blokad -> parametr konfiguracyjny `DISTRIBUTED_LOCK_TIMEOUT`, wyrażony w sekundach
 - wartość: 1- nieograniczona; domyślnie 60
 - po przekroczeniu czasu oczekiwania polecenie jest wycofywane i może być powtórzone
 - `ORA-02049: time-out distributed transaction waiting for lock`



Blokowanie

- ⇒ zablokowane dane muszą być natychmiast zwolnione
- ⇒ blokowanie segmentu wycofania
- ⇒ czas usunięcia awarii sprzętowej bardzo długi



Manualne zatwierdzenie/wycofanie

- ➔ uzyskanie informacji, czy transakcja lokalna wchodząca w skład transakcji rozproszonej powinna być zatwierdzona, czy wycofana -> perspektywa SYS.DBA_2PC_PENDING

```
ALTER SESSION ADVISE COMMIT;
INSERT INTO emp@LAB.WORLD ... ;
/* zalecane zatwierdzenie transakcji lokalnej w węźle LAB.WORLD */
```

```
ALTER SESSION ADVISE ROLLBACK;
DELETE FROM emp@ORC1.WORLD ... ;
/* zalecane wycofanie transakcji lokalnej w węźle ORC1.WORLD */
```

```
ALTER SESSION ADVISE NOTHING;
```

```
DBA_2PC_PENDING.ADVISE
```



```
R
```

```
C
```



Manualne zatwierdzenie/wycofanie

- ➔ uzyskanie informacji na temat transakcji rozproszonej inicjującej COMMIT -> perspektywa SYS.DBA_2PC_PENDING

```
COMMIT COMMENT 'komentarz';
```

```
DBA_2PC_PENDING.TRAN_COMMENT
```

➔ komentarz

- może zawierać np. rodzaj aplikacji inicjującej COMMIT, rodzaj operacji
- max. 50 znaków

```
set transaction name 'modyfikacja salda';
```

```
SQL> select name, status from v$transaction;
NAME                                STATUS
-----
modyfikacja salda                   ACTIVE
```



Manualne zatwierdzenie/wycofanie

- ➔ użytkownik realizujący transakcję lokalną otrzymuje komunikat:

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

↑
identyfikator lokalnej transakcji będącej częścią trans. rozproszonej

- ➔ analiza zawartości DBA_2PC_PENDING w bazie lokalnej

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



Manualne zatwierdzenie/wycofanie

	nazwa globalna bd koordynatora	identyfikator bd koordynatora	
LOCAL_TRAN_ID	1.21.17		
GLOBAL_TRAN_ID	PO1.CIO1.POZ.PL	55d1c563	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



Manualne zatwierdzenie/wycofanie

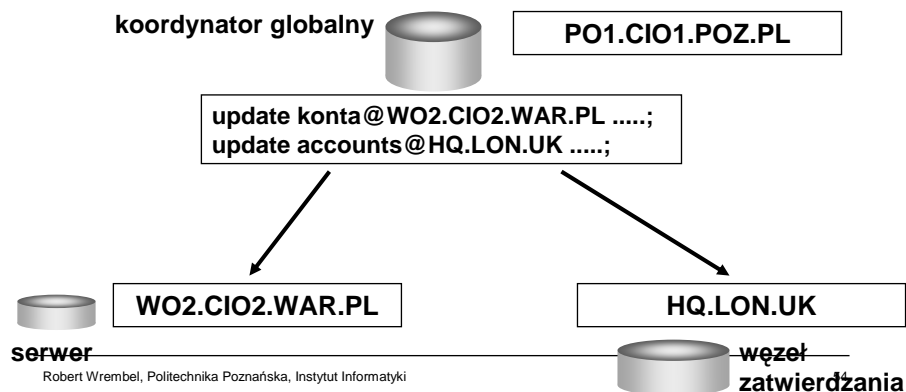
➤ atrybut STATE:

- **collecting**
 - wystąpi tylko dla koordynatora
 - gromadzenie potwierdzeń z węzłów
- **prepared**
 - transakcja przygotowana do zatwierdzenia
 - potwierdzenie przygotowania wysłane do koordynatora lub nie
- **committed**
 - transakcja w węźle zatwierdzona
- **forced commit**
 - transakcja w węźle zatwierdzona manualnie
- **forced abort**
 - transakcja w węźle wycofana manualnie



Manualne zatwierdzenie/wycofanie

- **znalezienie węzła zatwierdzania -> zawiera informację czy transakcja rozproszona powinna zostać zatwierdzona (wycofana)**
- **perspektywa SYS.DBA_2PC_NEIGHBORS**





Manualne zatwierdzenie/wycofanie

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

węzeł jest serwerem
żądania bazy danych

dołączenie
zrealizowane z konta

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



Manualne zatwierdzenie/wycofanie

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



Manualne zatwierdzenie/wycofanie

PO1.CIO1.POZ.PL

- znalezienie identyfikatora transakcji lokalnej w węźle PO1.CIO1.POZ.PL na podstawie id transakcji globalnej (zob. slajd "in-doubt" 4 i 5)

```
SELECT local_tran_id FROM sys.dba_2pc_pending
WHERE global_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';
```



Manualne zatwierdzenie/wycofanie

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	SCOTT	
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	← HQ.LON.UK jest węzłem zatwierdzania
DBUSER_OWNER	ALLEN	
INTERFACE	C	←
DBID	00000390	
SESS#	1	
BRANCH	1	



Manualne zatwierdzenie/wycofanie

HQ.LON.UK

➤ odczytanie statusu transakcji w węźle zatwierdzania

```
SELECT local_tran_id, global_tran_id, state, commit#  
FROM 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



Manualne zatwierdzenie/wycofanie

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

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

```
DBA_2PC_PENDING.LOCAL_TRAN_ID
```

➤ uprawnienia systemowe

- FORCE TRANSACTION, FORCE ANY TRANSACTION

➤ DISTRIBUTED_RECOVERY_CONNECTION_HOLD_TIME

- czas (w sekundach) przez który dołączenie bazodanowe pozostaje aktywne w przypadku niemożliwości zakończenia transakcji rozproszonej
- domyślnie 200 sekund



Crash testy

- symulacja 10 typów awarii transakcji rozproszonej

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

- 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

Uwaga: wyłączyć automatyczne odtwarzanie transakcji rozproszonej we wszystkich węzłach uczestniczących w tej transakcji

```
ALTER SYSTEM DISABLE DISTRIBUTED RECOVERY;
```