Relacyjne Bazy Danych

Andrzej M. Borzyszkowski
PJATK/ Gdańsk

materiały dostępne elektronicznie
http://szuflandia.pjwstk.edu.pl/~amb

Transakcje

Współbieżność

- Przykład: przelewanie pieniędzy z konta A na konto B, początkowe stany obu kont równe 100, więc suma 200:

czas	użytkownik 1	użytkownik 2
0 min	czyta konto A, wynik 100	
1 min		odejmuje 50 z konta A
2 min		dodaje 50 do konta B
3 min	czyta konto B, wynik 150	

  - a więc użytkownik 1 sądzi, że na obu kontach jest razem 250

Współbieżność, c.d.

- Przykład: każdy z użytkowników dodaje swój wkład do konta, stan początkowy 50:

czas	użytkownik 1	użytkownik 2
0 min	czyta stan konta, wynik 50	
1 min		czyta stan konta, wynik 50
2 min	nowa wartość konta 110	
3 min		nowa wartość konta 125

  - każdy z użytkowników sądzi, że nowa wartość konta jest powiększona o jego wpłatę, odp. 60 i 75 (i umożliwi w przyszłości wypłatę)

Współbieżność, III

- Przykład: przelew raz jeszcze

czas	użytkownik 1	SYSTEM
1 min	odejmuje 50 z konta A	
2 min		AWARIA
1 godz	stan konta A pomniejszony o 50, stan konta B bez zmian	

  - tak więc suma obu kont będzie mniejsza niż przed awarią byłaby większa, gdyby przelew zacząć od wpłaty

Współbieżność – wyzwania

- Trzy problemy
  - niespójna analiza
  - utracona modyfikacja
  - niezatwierdzona wartość

Transakcja – niepodzielna jednostka działań

  - albo wykonają się wszystkie operacja w transakcji, albo żadna
  - tzn. nowe wartości muszą być zatwierdzone
  - transakcja zajmuje zero czasu !

Operacje na danych

- Rodzaj operacji
  - odczyt – read(X)
  - zapis – write(X)
- Wielkość operacji
  - atomowa dana (komórka w tabeli)
  - wiersz tabeli (pojedyncza encja)
  - cała tabela
  - wielkość wyznaczona przez implementację (blok w systemie plików itp.)
- Również operacje zatwierdzenia (*commit*) i wycofania (*rollback*, *abort*)

Wycofanie transakcji – przyczyny

- Transakcja musi wykonać wszystkie operacje
  - a jeśli to niemożliwe, to musi wycofać już dokonane (*rollback, undo*)
- Przyczyny wycofania
  - przerwanie wykonania – błędy pamięci, przesyłania danych, spowodowane poza SZBD
  - błędy operacji z transakcji, jawna operacja wycofania
  - konieczność spowodowana współbieżnością (o tym będzie wykład)
  - również upływ czasu powoduje wycofanie niezatwierdzonej transakcji
  - awarie trwałych danych (pamięć dyskowa, ...)

Wycofanie transakcji – narzędzia

- Narzędzia wycofania
  - dziennik – zapis każdego ruchu (start(T), read(T,X), write(T,X,old,new), commit(T), abort(T))
  - mogą być prostsze dzienniki
  - w razie wycofania transakcji dziennik posłuży do odtworzenia poprzedniego stanu
  - po zatwierdzeniu transakcji i utrwaleniu jej wyników fragmenty dziennika są usuwane
- Wycofanie transakcji jest co najmniej tak czasochłonne jak sama transakcja, a raczej dużo bardziej

Wycofanie transakcji – wariacje

- Niektóre systemy przewidują ustawienie w transakcji punktów kontrolnych (savepoints)
  - wycofanie następuje do ostatniego takiego punktu
  - PostgreSQL od wersji 8.* posiada też to narzędzie
  - w zasadzie jest to sprzeczne z ideą atomowości transakcji
- Transakcja może obejmować kilka systemów (long transaction)
  - zatwierdzanie dwufazowe: każdy z systemów posiada dziennik pozwalający odtworzyć stan poprzedni i nowy
  - jeśli wszystkie systemy zakończyły pomyślnie ten etap, koordynator zaleca przyjęcie nowego stanu, wpp. odtworzenie poprzedniego stanu przez wszystkie systemy
  - idea transakcji zależy mocno od pewności, że koordynator będzie w stanie skutecznie porozumieć się, ze wszystkimi uczestnikami

Transakcja – własności ACID

- Cztery własności charakteryzujące transakcje
  - A – niepodzielność (atomic) – transakcji nie da się podzielić na podoperacje
  - C – spójność (consistency) – po zatwierdzeniu transakcji (i po wycofaniu transakcji) baza danych jest w stanie spójnym, tak jak była przed rozpoczęciem transakcji
  - I – odizolowanie (isolated) – transakcja przebiega tak, jak by w danym momencie była jedyną transakcją w systemie
  - D – trwałość (durable) – zatwierdzenie transakcji oznacza, że jej wyniki są trwale widoczne w bazie

Przebiegi transakcji

- Przebieg (wykonanie, historia) transakcji T1,...,Tn
  - ciąg operacji z T1, ..., Tn, t.że operacje z każdej transakcji występują w przebiegu w tej samej kolejności co w transakcji
- Przykład (T1 i T2, czytają i zapisują X i Y, a=rollback)
  - r1(X);r2(X);w1(X);r1(Y);w2(X);w1(Y);
  - r1(X);w1(X);r2(X);w2(X);r1(Y);a1;
- Operacje w konflikcie
  - jeśli należą do różnych transakcji, oraz
  - dotyczą tego samego obiektu, oraz
  - co najmniej jedna z operacji zapisuje ten obiekt
- Przebieg nie musi koniecznie być ciągiem
  - musi być ustalona kolejność operacji w konflikcie
  - oraz kolejność operacji z jednej transakcji

Przebiegi odtwarzalne

- Transakcja T1 czyta z transakcji T2 w danym przebiegu, jeśli
  - istnieje obiekt X, t.ż. w2(X) jest wcześniej niż r1(X)
- Przebieg jest odtwarzalny, jeśli
  - każda transakcja T1 czytająca z transakcji T2 jest zatwierdzona dopiero po zatwierdzeniu T2
  - kontrprzykład: r2(X);w2(X);r1(X);r2(Y);w1(X);c1;a2; – T1 odczytała X z T2, ale T2 została wycofana
- Problemy
  - utracona modyfikacja nadal możliwa
  - wycofania mogą powodować kolejne wycofania (kaskada):
  - r2(X);w2(X);r1(X);r2(Y);w1(X);a2;a1; – po wycofaniu T2 okazało się, że przeczytana wartość X jest nieaktualna

Przebiegi odtwarzalne, c.d.

- Przebieg jest bez kaskad, jeśli
  - żadna transakcja nie czyta obiektów zapisanych przez niezatwierdzone inne transakcje
- Przebieg jest ścisły, jeśli
  - żadna transakcja nie czyta ani nie zapisuje obiektów zapisanych przez niezatwierdzone inne transakcje
- Dla przebiegu ścisłego odtwarzanie jest łatwe, wystarczy przywrócić poprzednią wartość obiektów
  - dla innych przebiegów istnieją algorytmy, ale prosty pomysł nie wystarcza
  - przykład: X=1: w1(X,5);w2(X,7);a1;c2; po wycofaniu T1 przywracamy X=1, ale X powinno być 7

Szeregowalność

- Przebieg jest sekwencyjny, jeśli transakcje wykonywane są po kolei, bez przeplatania operacji
  - przebieg jest szeregowalny, jeśli w pewnym sensie jest równoważny sekwencyjnemu
- Równoważność przebiegów
  - dają ten sam rezultat – ale jak to sprawdzić?
  - operacje w konflikcie wykonywane są w tej samej kolejności
  - jest jeszcze trzecia definicja, mniej ograniczająca
- Przykład: r1(X);w1(X);r1(Y);w1(Y);r2(X);w2(X); – T1 potem T2
  - r1(X);w1(X);r2(X);w2(X);r1(Y);w1(Y); – równoważny, T1<T2
- r1(X);r2(X);w1(X);w2(X);r1(Y);w1(Y); – nie jest szeregowalny, bo T2<T1<T2
- Przebieg jest szeregowalny, jeśli *nie ma cyklu* w grafie kolejności

SQL/ PostgreSQL

- BEGIN (można użyć BEGIN WORK) w SQL nie występuje, ponieważ każde wyrażenie SQL rozpoczyna transakcję
  - PostgreSQL transakcja rozciąga się na jedną instrukcję, jeśli ma być dłuższa, trzeba użyć BEGIN
- COMMIT (można użyć COMMIT WORK) zatwierdzenie – kończy transakcję pozytywnie, wszystkie dane od tego momentu należy uważać za zatwierdzone, w szczególności dostępne dla innych transakcji
- ROLLBACK (również w wersji ROLLBACK WORK) wycofanie – kończy transakcję niepowodzeniem, dane tymczasowe są przywrócone do poprzedniego stanu

Transakcje w PostgreSQL

- PostgreSQL działa domyślnie w trybie chained (niejawnych transakcji), instrukcja jest całą transakcją chyba, że jest częścią bloku BEGIN ROLLBACK/COMMIT
  - np. SQL server Microsoftu wymaga podania SET IMPLICIT_TRANSACTIONS
  - standard SQL wymaga jawnego zakończenia transakcji
- Nie wolno zagnieżdżać transakcji
  - tzn. BEGIN musi mieć do pary COMMIT albo ROLLBACK nim nastąpi następny BEGIN
- Transakcje powinny być w miarę krótkie
  - w szczególności należy pilnować, by częścią transakcji nie był dialog z użytkownikiem – najpierw dane, potem transakcja