

JLoXiM Transaction Manager

bez Loga

Łukasz Chmielewski

Wydział Matematyki Informatyki i Mechaniki

26 listopada 2009

Plan prezentacji

- 1 Wprowadzenie
- 2 Problemy dotychczasowej wersji
- 3 Co zostało zrobione?
- 4 Ile jeszcze przed nami?
- 5 Plany na przyszłość

ARIES - to już było, ale... jeszcze o WAL

- rozwinięcie skrótu: Write-Ahead Logging
- rekordy dziennika opisujące zmianę na danych muszą zostać zapisane, zanim zmiana zostanie utrwalona na dysku
- system nie może nadpisać danych póki przynajmniej informacje undo nie zostaną zapisane
- o WAL dba Transaction Manager

Transaction Manager - struktury

- Przede wszystkim: Kolekcja transakcji
- Dla każdej transakcji kolekcja stron (TransactionBufferPage), z których ona korzysta
- TransactionBufferManager - przydziela strony (i zakłada locki)
- Mapowania wszystkich struktur obecnych w komponencie Bufferspace (TransBufferspace itp.)

Czym zajmuje się Transaction Manager?

- Tworzy transakcje
- Zawiaduje dostępem do stron, bufferspace'ów
- Odpowiada za ogólnie pojętą współbieżność (pomiędzy transakcjami i w ich obrębie)
- Loguje zmiany na danych czynione przez transakcję (przy pomocy Loga)

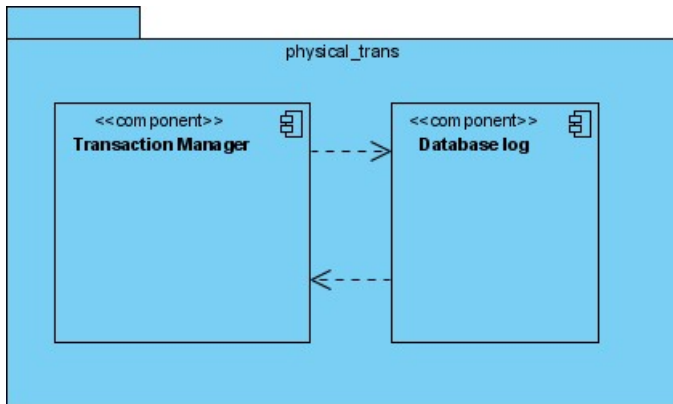
Co łączy Managera Transakcji z logiem?

- Manager odpowiada za transakcje
- Transakcja to ACID
- Aby zagwarantować ACID - Log

Plan prezentacji

- 1 Wprowadzenie
- 2 Problemy dotychczasowej wersji**
- 3 Co zostało zrobione?
- 4 Ile jeszcze przed nami?
- 5 Plany na przyszłość

Cały mechanizm nierozdzielony na pakiety



Wzajemne zależności między modułami

Niedociągnięcia obecnej wersji od strony formalnej...

- Problem z testami
 - Niezpełne testy jednostkowe
 - Niewyczerpujące testy integralności
 - Słabe logowanie w trakcie testów
- Praktycznie niekomentowany, mało przejrzysty kod

...

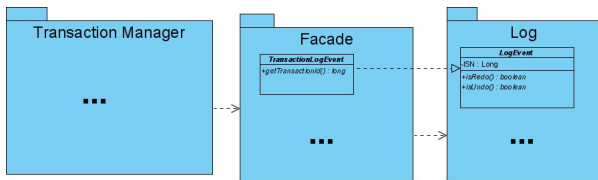
... i merytorycznej

- Obsługa współbieżnego dostępu do strony
 - nie zawsze poprawne momenty zakładania locków
 - złe poziomy locków
- Problem z wielowątkowością transakcji
 - tego po prostu nie ma

Plan prezentacji

- 1 Wprowadzenie
- 2 Problemy dotychczasowej wersji
- 3 Co zostało zrobione?**
- 4 Ile jeszcze przed nami?
- 5 Plany na przyszłość

Podział na 3 komponenty



- Rozbicie elementu na 3 moduły pozwoliło na uniknięcie dwustronnych zależności
- Każdy moduł to 3 podmoduły (API, implementacja, testy integracyjne)

Transaction Manager

- Implementacje mechanizmów związanych z logiem przeniesione do Fasady (LogEventProcessor, RollbackHelper)
 - Eventy
 - TransactionLogEventBegin
 - TransactionLogEventChangePage
 - TransactionLogEventRollback
 - TransactionLogEventCommit
 - ...
- przeniesione do Fasady
- Odpowiednia modyfikacja importów
 - Użycie metod fasady zamiast powyższych elementów

Log

- Przeniesienie interfejsów zawierających logikę logu (LogRollbackHelper, LogRecoverer itp.) do fasady
- Pozostają proste mechanizmy:
 - LogEvent, TransactionLogEvent
 - iteratory
 - reader
 - writer
- Zwykła przesuwalna taśma

Fasada

- Elementy przeniesione z innych modułów
 - Powiązane z logiem elementy TMA (m.in. eventy)
 - Cała logika z Loga
- Zapewniony odpowiedni interfejs dla TMA

```
public interface DBLogTransactionHelper {  
    void commitTransaction(long transactionId) throws DBLogException;  
  
    void logChangePage(boolean isRedo, boolean isUndo, long transactionId, int bufferPageId,  
        ByteBuffer pagesLogStruct, int sizeofPagesLogStruct, int actionNum);  
  
    long createSavePoint(boolean isRedo, boolean isUndo, long transactionId) throws DBLogException;  
  
    void beginLogTransaction(long transactionId) throws DBLogException;  
    ...  
}  
public interface DBLogRollbackHelper {  
    void rollbackTransaction(long transactionId) throws DBLogException;  
    ...  
}
```

Fasada c.d.

Podstawowe operacje, przy użyciu API Fasady:

- Commit
 - Odpowiednia ilość wywołań `logChangePage()`
 - Wywołanie `commitTransaction()` // - wpisuje event commit do loga
- Rollback
 - Po prostu wywołanie metody `rollbackTransaction`, która loguje event rollback i przywraca zmiany

Plan prezentacji

- 1 Wprowadzenie
- 2 Problemy dotychczasowej wersji
- 3 Co zostało zrobione?
- 4 Ile jeszcze przed nami?**
- 5 Plany na przyszłość

TODO:

- Refaktoryzacja kodu
- Testy
 - dopasowanie do istniejącej struktury
 - uzupełnienie testów jednostkowych
 - całościowe testy integracyjne
 - Mockito?

Współbieżność transakcji

- Przemysłenie koncepcji (`beginWorkingOnPage`, `changeUsePurpose`, `finishWorkingOnPage`)
- Odpowiednie użycie locków Lock Managera
- Wyczerpujące testy

Transakcja wielowątkowa

- Dane: jedna transakcja, jedna strona, wiele wątków
- Zdefiniowanie problemu: jak przekazywać kolejnym wątkom stronę bez zdjęcia locka
- Rozwiązanie: dodatkowy poziom locków
- Mechanizm czytelników i pisarzy
- Na deser: wyczerpujące testy

Plan prezentacji

- 1 Wprowadzenie
- 2 Problemy dotychczasowej wersji
- 3 Co zostało zrobione?
- 4 Ile jeszcze przed nami?
- 5 **Plany na przyszłość**

Odtwarzanie bazy danych

- brak mechanizmu odtwarzania stanu bazy po awarii

Savepoint

- Mamy zbyt mało danych
- jak aktualnie używane strony
- czy stan transakcji z założonymi lockami
- co z wieloma wątkami jednej transakcji

Wprowadzenie
Problemy dotychczasowej wersji
Co zostało zrobione?
Ile jeszcze przed nami?
Plany na przyszłość

Koniec

Jakieś pytania?