

Согласованное чтение и долгие транзакции

Самофатов Н.А.



Проблемы

- ◆ Исторически (до Firebird 3.0 включительно), режим изоляции READ COMMITTED позволяет запросу видеть подтвержденные данные по состоянию на момент чтения, т. е. не целостную картину
- ◆ Хранимые процедуры и триггеры работают по-разному в зависимости от режима изоляции транзакции, в которой они вызываются
- ◆ Возникающие конфликты обновления данных должны обрабатываться программистом приложения
- ◆ Длительные транзакции блокируют сборку мусора

Проблема (пример)

Подключение 1

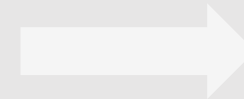
```
INSERT INTO T ...
INSERT INTO T ...
INSERT INTO T ...
-- 1000 записей
COMMIT;
INSERT INTO T ...
...
INSERT INTO T ...
-- 1000 записей
COMMIT;
...
INSERT INTO T ...
INSERT INTO T ...
...
```

Подключение 2

```
...
SELECT COUNT(*) FROM T
...
```

Подключение 2

```
...
SELECT COUNT(*) FROM T
...
```



Ожидание :)

0

1000

Реальность :(

229

1356

Функционал

- ◆ Новый режим изоляции READ COMMITTED READ CONSISTENCY
 - Снимок данных на момент начала запроса
 - Перезапуск запроса в случае конфликта обновления
- ◆ Новый алгоритм сборки мусора, эффективно работающий при наличии долгих транзакций
- ◆ API для работы с общими снимками

История вопроса

- ◆ первая реализация (Самофатов, Ред База Данных 2.5, 2014 год)
- ◆ вторая реализация (Самофатов, патч для Firebird 3, 2014 год)
- ◆ доработка, интеграция кода в Firebird (Хорсун и Симаков, Firebird 4, 2014-2020 год)
- ◆ API для работы с общими снимками (Фернандес, Firebird 4, 2019 год)

Режим READ COMMITTED READ CONSISTENCY

- ◆ Используется по умолчанию без изменения приложений (ReadConsistency=1 в firebird.conf)
- ◆ Также можно указать режим явно
 - SQL: SET TRANSACTION READ COMMITTED READ CONSISTENCY
 - API: вызов `isc_start_transaction` с указанием `isc_tpb_read_consistency` в TPB
- ◆ Показывается в `MON$TRANSACTIONS.MON$ISOLATION_MODE` (4)

Внутренняя реализация

- ◆ новый способ снятия снимков (по CN - номеру коммита)
- ◆ таблица в общей памяти для учёта номеров коммита для транзакций
- ◆ таблица в общей памяти для хранения списка снимков
- ◆ новый алгоритм сборки мусора, использующий список снимков

API для работы с общими снимками

- ◆ Возможность иметь несколько транзакций с одинаковым снимком данных
 - Предназначена для распараллеливания больших операций чтения данных
 - Режим изоляции concurrency (один снимок для транзакции)
- ◆ Получить номер снимка
 - SQL: `RDB$GET_CONTEXT('SYSTEM', 'SNAPSHOT_NUMBER')`
 - API: вызов `isc_transaction_info` с `fb_info_tra_snapshot_number`
- ◆ Задать номер снимка
 - SQL: `SET TRANSACTION SNAPSHOT AT NUMBER <snapshot number>`
 - API: вызов `isc_start_transaction` с `isc_tpb_at_snapshot_number <snapshot number length> <snapshot number>`

Более подробное видео с примерами

<https://youtu.be/sHsEOltYfd0>





Вопросы

nikolay.samofatov@red-soft.ru