Библиотека транзакционного доступа к файлам из PostgreSQL

Семенов Александр Сергеевич

Научный руководитель:
доц. каф. СП, к.ф.-м.н. Д.В. Луцив

13.05.2024

Актуальность (1)

- Одна из составляющих любой информационной системы — данные
- Для хранения часто используются базы данных
- Один из типов хранимых данных — бинарные файлы
  - pdf-документы, изображения, архивы, медицинские данные...
- С ростом объемов бинарных данных становится необходимым их эффективное хранение

Актуальность (2)

Для многих информационных систем важна:

- Транзакционная поддержка операций над данными
- Возможность версионировать данные
- Стандартный интерфейс работы с данными

Постановка задачи

Целью работы является разработка библиотеки, с помощью которой появляется возможность реализовать доступ к бинарным данным на уровне SQL с ACID гарантиями

Задачи:

- Изучить существующие подходы к решению задачи хранения бинарных данных в PostgreSQL
- Определить подход к хранению бинарных данных
- Реализовать библиотеку, предоставляющую API для транзакционного доступа к бинарным данным в PostgreSQL
- Провести тестирование полученного решения

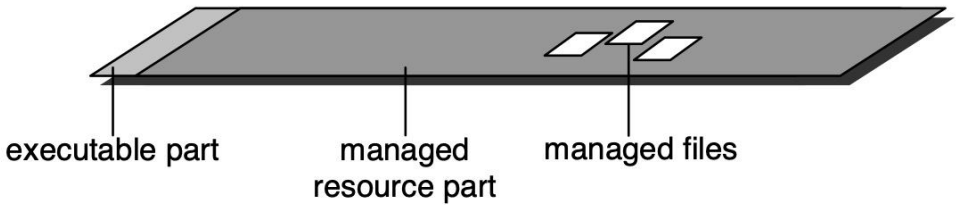
PostgreSQL: тип bytea

- bytea — тип данных для хранения произвольного набора байтов
- Применяется механизм TOAST[1] для управления данными > 8Кб
- Недостатки TOAST:
  - Снижение производительности работы с большими объектами данных
  - Требует дополнительного места для хранения TOAST таблиц
  - Избыточное обновление
- Не поддерживается версионирование

[1] The Oversized-Attribute Storage Technique

Smart files[2]

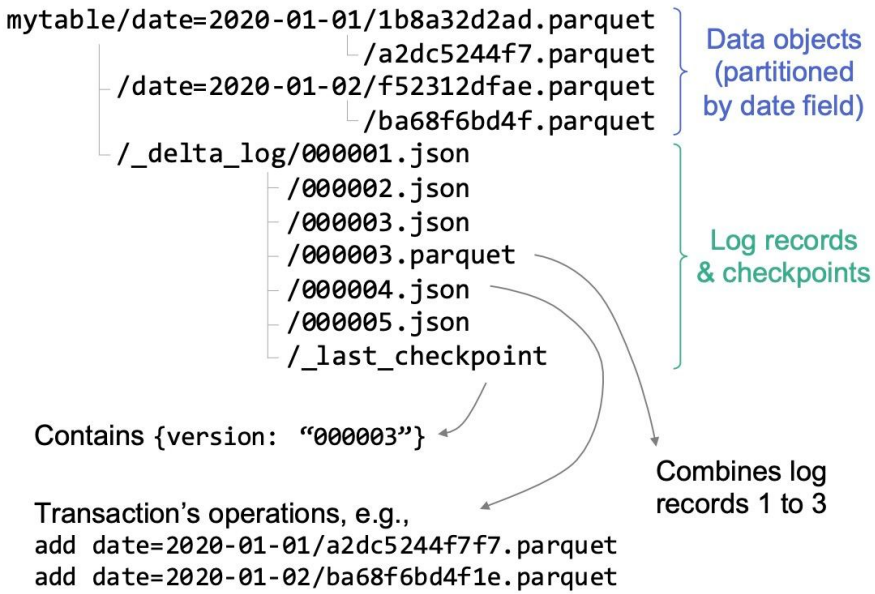
- Умные файлы являются исполняемыми, внутри есть своя файловая система, которая даёт возможность протоколировать все операции и поддерживает механизм версионирования
- Однако такой подход нельзя в полной мере назвать подходящим для решения задачи
  - требует пересмотра процессов работы с файлам
  - размер таких файлов будет больше



[2] Smart files: combining the advantages of DBMS and WfMS with the simplicity and flexibility of spreadsheets

Delta Lake[3]

- Слой хранения данных, который поддерживает ACID гарантии
- Использует формат файлов Parquet для хранения данных
- Помимо этого используется транзакционный журнал



[3]Delta Lake: High-Performance ACID Table Storage over Cloud Object Stores

Выбор файловой системы (1)

- Выбор между EXT4, Btrfs, ZFS
- Критерии отбора:
  - Наличие библиотеки для взаимодействия с файловой системой из программного кода
  - Поддержка транзакционного механизма взаимодействия через механизм copy-on-write
  - Поддержка создания снапшотов за константное время для реализации версионирования

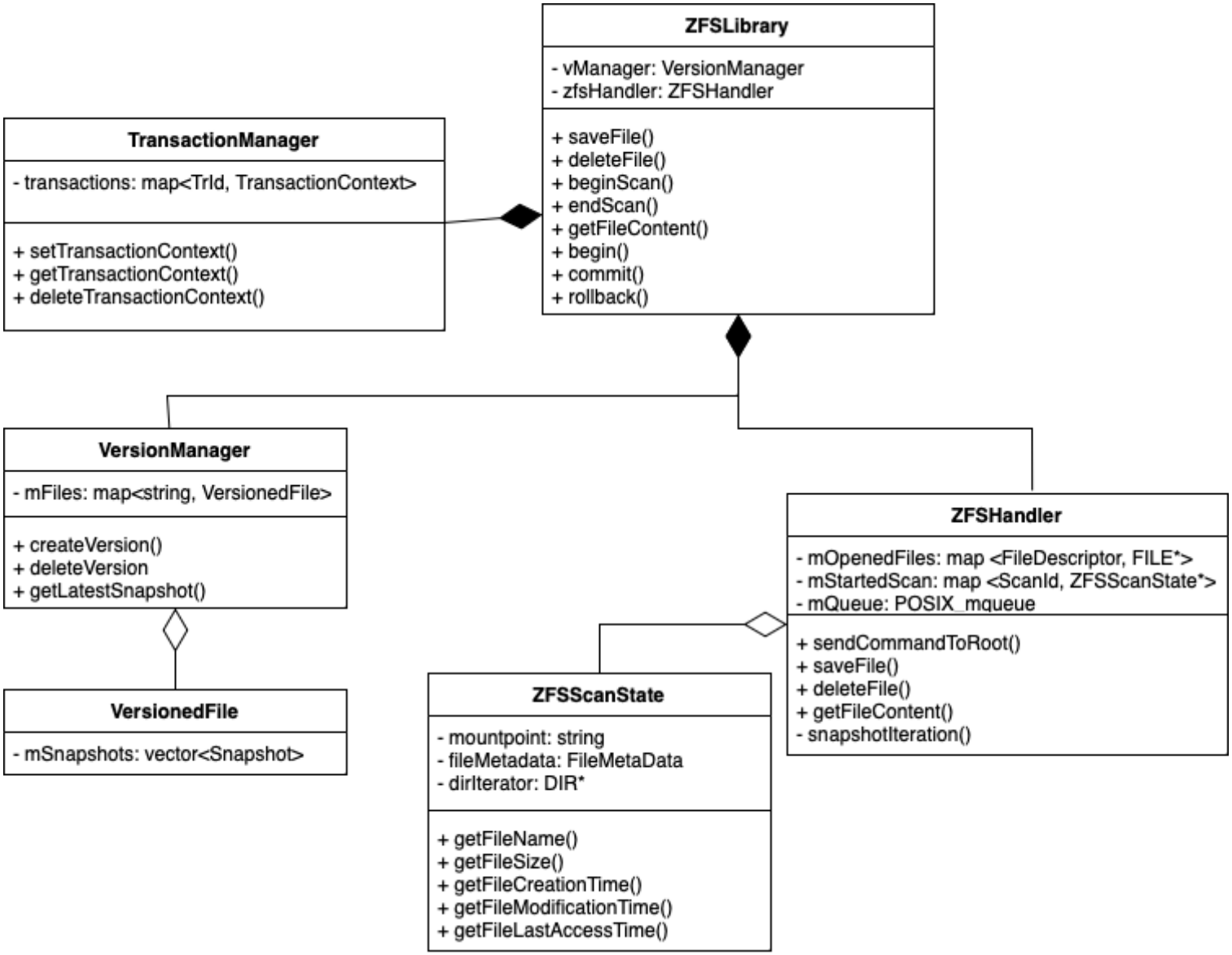
Выбор файловой системы (2)

- EXT4 не поддерживает механизм copy-on-write
- Btrfs показывает более низкую эффективность в сравнении с ZFS при работе с большими объёмами данных[4]
- Решено остановиться на ZFS и библиотеке OpenZFS[5]

[4] https://www.enterprisedb.com/blog/postgres-vs-file-systems-performance-comparison

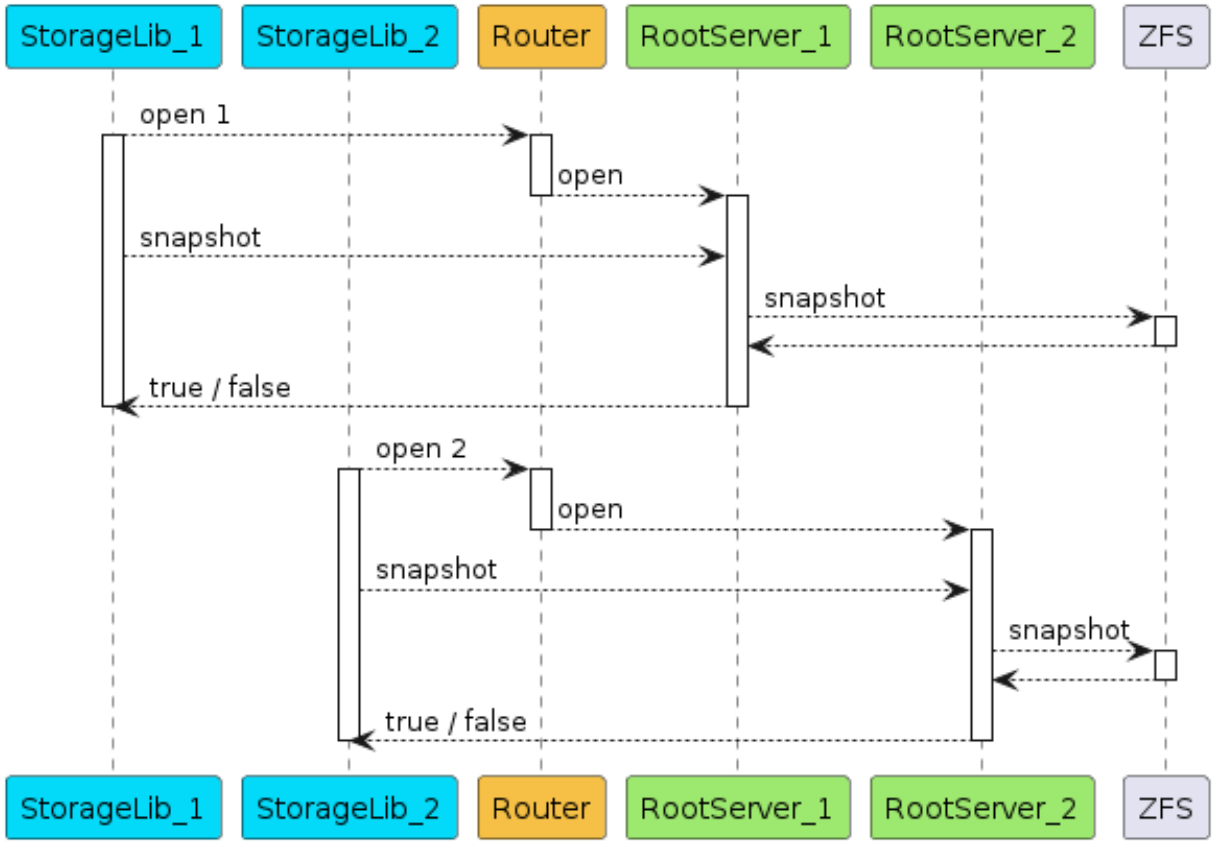
[5] https://github.com/openzfs/zfs

Диаграмма классов библиотеки



Разграничение адресного пространства

- Библиотека должна работать в пользовательском адресном пространстве
- Однако выполнение некоторых функций файловой системы ZFS происходит на уровне ядра



Функциональное тестирование

- Для тестирования функциональности библиотеки разработан отдельный фреймворк
- Он при помощи системного вызова fork() создаёт новые процессы и запускает в них тестовые сценарии
- Таким образом выполняются операции над файловым хранилищем в нескольких процессах одновременно, имитируя реальное использование библиотеки

Результаты

В результате были решены следующие задачи:

- Изучены существующие подходы к решению задачи хранения бинарных данных в PostgreSQL
- Определен подход к хранению бинарных данных
- Реализована библиотеку, предоставляющая API для транзакционного доступа к бинарным данным в PostgreSQL[6]
- Разработан фреймворк для составления тестовых сценариев

Планы для дальнейшей работы:

- Составить тестовые сценарии и подтвердить корректность работы библиотеки

[6] Код проекта закрыт и принадлежит компании ООО “Датаджайл”