

Санкт-Петербургский Государственный Университет

Кафедра системного программирования
09.03.04 «Программная инженерия»

Бощенко Алина Вячеславовна

Отчёт по учебной практике

**“Транзакционная синхронизация данных в
распределенной базе данных Xodus”**

Руководители:

Луцив Дмитрий Вадимович
доцент Кафедры системного программирования, к.ф.-м.н

Науменко Евгений Владимирович
технический руководитель YouTrack, JetBrains s.r.o

Санкт-Петербург
2021

Saint Petersburg State University
Software Engineering

Boshchenko Alina

**“Transactional Multi-Node Xodus database state
synchronization”**

Supervisors:

Dmitry Lusiv

Associate Professor of the Department of System Programming, Ph.D.

Naumenko Evgeny

Technical Lead, YouTrack, JetBrains s.r.o

Saint Petersburg
2021

Содержание

Введение	3
Постановка задачи	3
Цель	4
Задачи	4
План работы	4
Обзор	5
Заключение	7
Список литературы	7

Введение

Потеря данных является одной из главных проблем современной IT-индустрии. Все мы хотя бы раз слышали о трагической потере данных или пережили ее. Для предотвращения подобных ситуаций существует резервное копирование баз данных. Основная причина резервного копирования – сохранить важные файлы в случае сбоя системы. Данная работа направлена на создание методики синхронизации данных в удаленной и локальной базах, что необходимо для поддержания актуального состояния данных пользователей в резервном хранилище.

Постановка задачи

JetBrains Xodus - это встроенная бессхемная транзакционная база данных, написанная на Java и Kotlin. Первоначально она была разработана для JetBrains YouTrack, инструмента отслеживания задач и управления проектами. Xodus также используется в JetBrains Hub – платформе управления пользователями для командных инструментов JetBrains, и в некоторых внутренних проектах JetBrains. В качестве объекта для выполнения работы взято приложение YouTrack.

В приложении существуют две копии распределенной базы данных Xodus – локальная и удаленная. В настоящее время синхронизация выполняется с помощью протокола Rsync. Rsync — это программа для UNIX-подобных систем, которая эффективно выполняет синхронизацию файлов и каталогов в двух местах, необязательно локальных, с минимизированием трафика, используя кодирование данных при необходимости.

Rsync является не транзакционным¹ протоколом, из-за чего у этого подхода есть существенный недостаток. Протокол направлен на отслеживание изменений в файлах, однако, из-за особенностей реализации, он не нацелен на разграничение транзакций, из-за чего

¹ **Транзакция** — группа последовательных операций с базой данных, которая представляет собой логическую единицу работы с данными. Транзакция может быть выполнена либо целиком и успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще, и тогда она не должна произвести никакого эффекта.

появляется необходимость вводить дополнительные правила, чтобы зафиксировать момент, когда база готова к синхронизации.

Цель

Создать механизм устойчивой и надежной транзакционной синхронизации данных между локальной и удаленной базами данных Xodus.

Задачи

1. Изучить существующие транзакционные протоколы и выбрать подход, который наиболее подходит цели работы.
2. Для консистентности подхода необходимо ввести ограничение – один узел данных должен перманентно находиться в состоянии только на чтение, в то время как другой - только на запись.
3. На основе собранной информации спроектировать и реализовать дополнительный слой в протоколе Rsync для придания ему свойства транзакционности.
4. Математически доказать корректность спроектированного алгоритма.
5. Протестировать реализованный алгоритм, главным образом с помощью нагрузочного тестирования.

План работы

февраль-март 2022:

- изучение архитектуры базы Xodus и структур хранения данных
- изучение транзакционных протоколов синхронизации данных
- анализ возможных подходов придания протоколу Rsync транзакционных свойств.

март-июль 2022:

- оценка совместимости различных подходов с приложением
- математическое доказательство корректности алгоритма

август-ноябрь 2022:

- реализация алгоритма в кодовой базе приложения YouTrack

ноябрь-декабрь 2022:

- тестирование

Обзор

Транзакция – это группа связанных действий, которые необходимо выполнить как одно действие. Другими словами, транзакция - это логическая единица работы, эффект которой виден вне транзакции либо полностью, либо не виден вовсе. Это необходимо для обеспечения целостности данных в приложении.

В архитектуре, основанной на событиях, типичным примером транзакции является обновление базы данных и создание события для использования другими сервисами. Цель состоит в том, чтобы эти две операции либо происходили вместе, либо не происходили вовсе, и это достигается при помощи объединения этих операций в одну транзакцию.

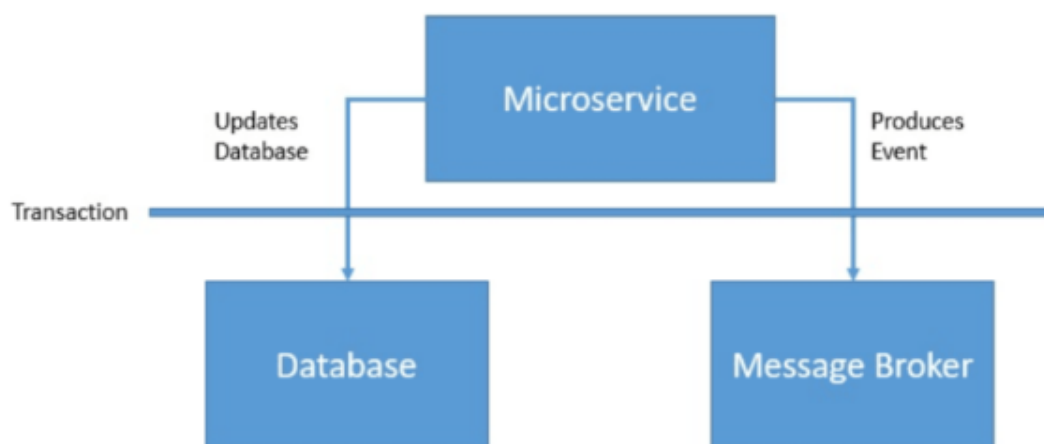


Рис.1: Пример транзакции в приложении

Для того, чтобы реализовать дополнительный слой в протоколе Rsync, обеспечивающий транзакционность, необходимо изучить строение существующих транзакционных протоколов, которые удовлетворяют требованиям работы.

В ходе написания обзора были рассмотрен протокол **Atomic transaction**, который основан на протоколе **Two-phase commit**.

Протокол работает следующим образом: один узел назначается координатором, а остальные узлы в сети – участниками. Протокол предполагает, что:

1. На каждом узле есть стабильное хранилище с журналом записи;
2. Ни один из узлов не дает фатального сбоя (сбоя навсегда);
3. Данные в журнале записи никогда не теряются или не повреждаются при сбое;
4. Любые два узла могут связываться с друг с другом.

Протокол инициируется координатором после достижения последнего шага транзакции. Затем участники отвечают сообщением о

соглашении или сообщением об отмене в зависимости от того, была ли транзакция успешно обработана на участнике.

Однако данный протокол предполагает применение транзакции либо на всех агентах, либо ни на одном из них, что не соответствует поставленной задаче. В рамках рассматриваемого приложения транзакция на узле, который находится в состоянии записи, применяется всегда, а значит протокол Atomic transaction не может быть взят в качестве ссылки.

Заключение

На данный момент изучена предметная область, сделан обзор предполагаемой работы и стека технологий, сформулирована цель и поставлены задачи. В настоящее время в процессе находится изучение архитектуры базы Xodus и выбор транзакционного протокола, который мог бы послужить ролевым примером.

Список литературы

1. Mukkamala R., and Hyuk Son S., **A Secure Concurrency Control Protocol for Real-Time Databases** – IFIP Advances in Information and Communication Technology book series, 1999
2. Abbott, R. K., and Garcia-Molina, H. **Scheduling Real-Time Transactions: A Performance Evaluation** – 14th VLDB Conference, 1992
3. Chandrakan, K., **Introduction to Transactions** – Baeldung, 2021
4. **JetBrains Xodus wiki** – <https://github.com/JetBrains/xodus/wiki>