

## Проектирование и реализация высоконадежной, отказоустойчивой и распределенной облачной базы данных как надстройки для базы данных PostgreSQL

Бакин А.О. (Университет ИТМО)

Научный руководитель – кандидат технических наук, доцент Перл И.А.  
(Университет ИТМО)

**Введение.** Задача автоматизации развертывания и эксплуатации высоконадежной и отказоустойчивой СУБД сопровождается рядом проблем. Для обеспечения доступности СУБД в рамках одного дата-центра применяются техники избыточности, например, primary-standby. При этом, важно не допустить отказ СУБД ввиду повышенной нагрузки, для чего следует обеспечить её масштабирование: горизонтальное или вертикальное. При использовании горизонтального масштабирования требуется решить проблему автоматизации процесса оркестрации (управления) для множества узлов СУБД и обеспечить ACID свойства транзакций среди всех узлов. Также немаловажно обеспечить прозрачность взаимодействия клиентов с узлами СУБД.

Однако, при выходе из строя самого дата-центра СУБД будет недоступна. Для решения этой проблемы СУБД может быть развернута в пределах нескольких дата-центров или же, зон доступности. При этом, необходимо решить проблему синхронизации данных между зонами доступности и обеспечить ACID свойства транзакций.

В связи с особенностями реализации каждой СУБД, вышеупомянутые проблемы следует решать индивидуально для каждой из них. В рамках работы в качестве целевой СУБД была выбрана PostgreSQL, как самая популярная open-source СУБД [1]. Целевой средой развертывания стала среда Docker, как одна из самых распространенных облачных сред исполнения.

В ходе исследования было выяснено, что для Docker задача автоматизации процесса оркестрации при использовании горизонтального масштабирования полностью не решается ни одним из существующих решений "из коробки". Задача репликации данных СУБД между дата-центрами также не может быть полноценно решена и автоматизирована, с использованием существующих решений. Таким образом, целью работы является автоматизация эксплуатации и масштабирования высоконадежного, отказоустойчивого и распределённого кластера СУБД на базе PostgreSQL в среде Docker.

**Основная часть.** Для решения обозначенных проблем было спроектировано и реализовано новое решение, получившее название PgFacade. Для обеспечения горизонтального масштабирования используется техника шардирования совместно с протоколом двухфазного коммита (2PC), чтобы обеспечить ACID свойства транзакций при работе с данными в нескольких шардах. Для репликации данных между дата-центрами также используется протокол 2PC. Решение состоит из двух частей: SQL посредник и оркестратор.

SQL посредник перехватывает все SQL запросы от клиентов и выполняет их обработку. Для запросов выборки данных (SELECT) решение строит план исполнения, с учетом распределенности данных между шардами, в том числе для запросов с JOIN [3]. Для запросов изменения данных (INSERT/DELETE/UPDATE), решение автоматически определяет шарды для изменения, используя значения ключей партиционирования из запроса. Также, для запросов на изменение данных, используется протокол 2PC для синхронизации данных как между шардами, так и между дата-центрами. Дополнительно, решение перехватывает все клиентские транзакции и фиксирует их, используя протокол 2PC. Протокол 2PC использует таймауты для предотвращения блокировок [4]. Для повышения производительности в посредник встроен пул соединений. Благодаря всему вышеперечисленному, решение работает прозрачно для клиентов, используя стандартный протокол взаимодействия.

Оркестратор следит за доступностью каждого из узлов и за количеством узлов типа primary и standby в каждом из шардов. Решение автоматически производит аварийное переключение standby в случае отказа primary. Дополнительно, обеспечивается непрерывное резервное копирование и возможность восстановления для каждого шарда.

Для повышения надежности решения, PgFacade также запускается с помощью нескольких узлов. Для обеспечения консенсуса при принятии решений используется протокол Raft [5], а оркестрацией занимается только Raft лидер. Благодаря этому, все решение целиком не содержит единой точки отказа. Обработкой SQL запросов занимаются все узлы PgFacade, а для равномерной нагрузки используется балансировщик.

Для подтверждения работоспособности решения было проведено тестирование. В ходе тестирования проверялась способность решения обеспечивать доступность всей СУБД в случае выхода узлов СУБД и/или дата-центров из строя. Также проверялась общая производительность SQL посредника и корректность исполнения им SQL запросов. По результатам тестирования был сделан вывод, что решение работоспособно и решает существующие проблемы и поставленные задачи.

**Выводы.** Было спроектировано и разработано решение для автоматизации эксплуатации и масштабирования высоконадежного, отказоустойчивого и распределённого кластера СУБД на базе PostgreSQL в среде Docker. Было проведено тестирование решения, показавшее его работоспособность. Решение отличается от существующих простотой использования и полной автоматизацией всех процессов.

#### **Список использованных источников:**

1. Aleem Akhtar, Popularity Ranking of Database Management Systems [Электронный ресурс]. 2023. URL: <https://arxiv.org/pdf/2301.00847.pdf> (дата обращения 03.02.2025).
2. Официальная документация PostgreSQL – URL: <https://www.postgresql.org/docs/> (дата обращения: 03.02.2025).
3. Liang F. et al. RelJoin: Relative-cost-based selection of distributed join methods for query plan optimization //Information Sciences. – 2024. – Т. 658. – С. 120022.
4. Babaylov V., Timofeeva G. Calculation of the optimal timeout of executing a SQL query for multi-master replication system //AIP Conference Proceedings. – AIP Publishing, 2022. – Т. 2522. – №. 1.
5. Diego Ongaro , John Ousterhout, In search of an understandable consensus algorithm // USENIX ATC'14: Proceedings of the 2014 USENIX conference on USENIX Annual Technical Conference, June 2014, Pages 305–320