

МАСШТАБИРУЕМОСТЬ PostgreSQL В РАСПРЕДЕЛЕННЫХ СРЕДАХ

Кабаков В.П.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Кузмин И.А. – ассистент кафедры ПИКС

Аннотация. Анализ масштабирования PostgreSQL в распределенных средах. Охватывает производительность в разных конфигурациях. Предоставляются теоретические основы и рекомендации для улучшения производительности и отказоустойчивости в условиях растущих объемов данных, с учетом требований разработчиков и администраторов.

Ключевые слова: база данных, оптимизация, масштабирование

Введение. В контексте современных высоконагруженных информационных систем, эффективное масштабирование баз данных становится ключевым фактором обеспечения производительности и доступности данных. Настоящее исследование фокусируется на аспектах масштабируемости PostgreSQL в распределенных средах. Анализируются методы горизонтального и вертикального масштабирования, а также применение репликации. Эксперименты включают анализ производительности и сравнение баз данных в различных конфигурациях. Полученные результаты предоставляют теоретические основы и практические рекомендации для оптимизации PostgreSQL с целью обеспечения эффективной работы в условиях постоянно растущей сложности и объемов данных [1].

Основная часть. На текущий момент, использование облачных технологий стало широко распространенным явлением в информационной инфраструктуре. Однако, несмотря на это, не все компоненты легко и незамедлительно адаптируются к облачным средам. Одним из наиболее сложных в интеграции компонентов является база данных, поскольку она постоянно требует значительных ресурсов и высокой производительности.

Особое внимание привлекают вопросы масштабируемости и надежности в условиях сбоев. Эти аспекты становятся особенно критичными в контексте баз данных, и именно по этой причине за последние годы наблюдается активное развитие альтернативных систем управления базами данных (СУБД) [2]. Этот тренд объясняется постоянной потребностью в эффективных решениях, которые обеспечивают высокую производительность и надежность в современных динамичных облачных средах.

Обычно акцент на масштабирование делается, когда происходит столкновение с острой нехваткой ресурсов, таких как процессорные ядра, память или место на диске. Эти системные ресурсы, предоставляемые компьютерами или облачными платформами, становятся критическими для эффективной работы.

Второй момент, когда масштабирование становится важным, связан с бизнес-потребностями и стратегическими задачами. Руководство или бизнес могут поставить задачу расширения на несколько дата-центров для обеспечения высокой доступности и отказоустойчивости. Это требует конфигурирования базы данных таким образом, чтобы она функционировала в нескольких локациях, готовых быстро реагировать на сбой в одном из дата-центров. Эта стратегия обеспечивает надежность и устойчивость в условиях выхода из строя отдельных центров обработки данных [3].

Процесс масштабирования можно отложить, например, на стадии разработки стартапа, особенно при создании минимального жизнеспособного продукта (MVP), фокус должен быть на быстром выводе продукта на рынок и получении обратной связи. Масштабирование, особенно сложные механизмы, такие как шардинг, могут ввести

дополнительную сложность и затраты, которые не оправданы, когда объем данных еще невелик и нет необходимости в высокой производительности.

Решение о внедрении масштабирования, в том числе и шардинга, следует откладывать до тех пор, пока реальные проблемы с производительностью не возникнут. Такой подход позволяет сосредоточить усилия на реальных вызовах, с которыми сталкивается продукт в процессе масштабирования и роста пользовательской базы.

Один из наиболее распространенных и простых способов - использование потоковой репликации.

Потоковая репликация представляет собой процесс, при котором существует мастер-узел, который передает журналы транзакций своим репликам. Этот подход удобен тем, что реплики могут действовать как горячие резервные узлы, готовые к включению в случае отказа мастера. Кроме того, на реплики можно направлять трафик только для чтения, что предоставляет опцию масштабирования.

Однако, при использовании потоковой репликации существуют проблемы, такие как возможный лаг репликации, что может привести к различным результатам при запросах, отправленных на мастер и реплику. Решение этой проблемы часто требует вмешательства на уровне приложения, где разработчики должны активно управлять тем, какие данные получать с мастера, а какие с реплик [4].

Важно учитывать, что при использовании автоматизированных систем вроде auto-failover, где роль мастера может меняться, приложение должно быть готово к таким изменениям, переустанавливая соединение и обрабатывая сбои.

В случае, если производительность оказывается на пределе, одним из наиболее простых и легких вариантов является scale-up, то есть, увеличение вычислительных ресурсов, таких как количество ядер и объем памяти. Этот подход, хоть и более затратный, может оказаться проще, чем внедрение сложных распределенных систем.

В итоге, PostgreSQL имеет ограниченные возможности масштабирования, особенно в контексте записи. Возможности масштабирования чтения более широки, включая потоковую репликацию, но каждый метод имеет свои ограничения и недостатки.

Список литературы

1. Масштабирование базы данных через шардирование и партиционирование [Электронный ресурс]. – 2022. – Режим доступа <https://highload.guidelog/scaling-database>. – Дата доступа :12.02.2024.
2. Что такое PostgreSQL? [Электронный ресурс]. – 2024. – Режим <https://azure.microsoft.com/ru-ru/resources/cloud-computing-dictionary/what-is-postgresql>. – Дата доступа :10.02.2024.
3. Масштабирование PostgreSQL: как мы укротили 10 миллиардов ежедневных записей и более 350 ТБ данных [Электронный ресурс]. – 2023. – Режим доступа <https://hackernoon.com/ru/масштабирование-postgresql%2C-как-мы-укротили-10-миллиардов-ежедневных-записей-и-350-ТБ-данных>. – Дата доступа :05.02.2024.
4. Масштабирование баз данных — партиционирование, репликация и шардинг [Электронный ресурс]. – 2023. – Режим доступа https://web-creator.ru/articles/partitioning_replication_sharding. – Дата доступа :20.01.2024.

UDC 004.65:004.75

POSTGRESQL SCALABILITY IN DISTRIBUTED ENVIRONMENTS

Kabakov V.P.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Kuzmin I.A. – assistant of the department of ICSD

Annotation. Research on scaling PostgreSQL in distributed environments, including analysis of horizontal and vertical scaling. The experiments cover performance and comparison of databases in different configurations. Provides theoretical foundations and recommendations for improving performance and resiliency in the face of growing data volumes, taking into account the requirements of developers and administrators.

Keywords: database, optimization, scaling