

# ПРОИЗВОДИТЕЛЬНОСТЬ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ И ЕЁ ЭФФЕКТИВНОЕ ТЕСТИРОВАНИЕ

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

Краснослободцев М. Ю.

Писарчик А. Ю – ассистент кафедры ПИКС.

Рассмотрены способы проверки эффективности реляционных баз данных, а также способы и инструменты тестирования оптимизаций производительности.

С проблемами эффективности баз данных (БД) может столкнуться любой инженер или администратор баз данных. Причины медленно работающей БД могут скрываться в самых различных аспектах системы: от неправильной конфигурации БД до некорректно написанных SQL-запросов. Данный материал призван помочь инженерам в поиске узких мест в их базах данных.

Существуют два основных подхода к тестированию производительности: эталонное тестирование и профилирование. Вкратце, главное отличие между ними можно выразить в нескольких словах: эталонное тестирование отвечает на вопрос “Насколько хороша производительность?”, а профилирование - “Почему производительность именно такова?” [1].

В первую очередь, необходимо понять, что именно будет измеряться – то есть, определить цели будущей оптимизации. Так, например, следующие показатели могут быть выбраны в качестве целей:

- Пропускная способность (т.е. транзакции в секунду);
- Время отклика;
- Масштабируемость системы;
- Уровень конкуренции.

В первую очередь, перед разработкой тестов необходимо понять, где кроется узкое место в системе. Это, например, можно сделать при помощи журналов медленных запросов, кои присутствуют почти в любой реляционной БД. Такие журналы занимают дополнительное место на жёстком диске, но позволяют записывать медленно работающие запросы (понятие “медленные” определяется самим разработчиком – журнал легко настраивается, и можно внести, насколько медленные запросы должны записываться в журнал). Кроме того, множество современных веб-сервисов, такие как *Amazon AWS* или *DigitalOcean* предоставляют исчерпывающую инфографику по нагрузке БД (и не только) в любом промежутке времени. Как только будет понятно, на каком этапе времени происходит падение производительности, можно предпринимать определенные шаги. Так, например, если окажется, что производительность падает при большом наплыве пользователей, то, возможно, стоит подумать об уровне конкуренции, что ставит вопрос об грамотной реализации очередей на стороне веб-приложения [2] или об автоматической масштабируемости во время пиковых периодов активности [3].

Если же производительность не связана с описанными выше ситуациями, стоит начать анализировать SQL-запросы. Оптимизация запросов не входит в рамки данного материала, но даны шаги, которые помогут необходимо предпринять, чтобы удостовериться в эффективности оптимизации. В первую очередь необходимо создать тестовую базу данных и воспроизвести уже оптимизированные запросы. Однако, при выполнении подобного теста, следует быть внимательным к следующим моментам, которые могут привести к неточным результатам во время тестирования:

- Объем данных должен совпадать (желательно – превышать) с реалистичным набором данных;
- Использование однопользовательского сценария для многопользовательского приложения;
- Итеративное выполнение одинаковых запросов. Результаты одинаковых запросов сохраняются в кеше, что приводит к тому, что сам запрос не выполняется;
- Отсутствие контроля ошибок. После теста необходимо проверить наличие ошибок, чтобы удостовериться, что все запросы были выполнены;
- Наборы данных при старте теста должны быть одинаковыми.

Сами тесты необходимо запускать несколько раз и записывать результат их выполнения. Если необходима высокая точность результатов – запускать их много раз. Полученные результаты можно сравнить с исходными данными и понять, была ли эффективна оптимизация.

Есть множество инструментов, которые могут помочь в тестировании. Так, например, SQL-оператор *EXPLAIN* поможет проанализировать то, как строится сам запрос базой данных. Или, например, *Apache Jmeter* позволит симитировать высокую нагрузку на веб-сервер, что может помочь в поиске узкого места во всей системе. Стоит упомянуть также и *MySQL Benchmark* – весьма удобная функция для анализа скорости выполнения операций.

Таким образом, были рассмотрены основные способы тестирования эффективности реляционных баз данных, даны шаги по поиску узких мест в системах, которые используют реляционные БД и способы оценки эффективности оптимизаций.

## Список использованных источников

1. Шварц, Б. *MySQL – Оптимизация производительности* / Б. Шварц – O’Reilly Media, 2018 – 60 с.
2. Шварц, Б. *MySQL – Оптимизация производительности* / Б. Шварц – O’Reilly Media, 2018 – 564 с.
3. Thomas A. Limoncelli – *The Practice of Cloud System Administration: DevOps and SRE Practices for Web Services* / Thomas A. Limoncelli – Informit, 2018 – 95 с.