

## ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ В ВЫСОКОНАГРУЖЕННЫХ СИСТЕМАХ

Кудрявцев М.Г.

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

Шамына А.Ю. – старший преподаватель

В данной работе произведён анализ основных техник масштабирования данных, получивших широкое распространение при разработке высоконагруженных систем.

С ростом объёмов данных возрастает и потребность в масштабировании систем для их обработки. Существует несколько техник масштабирования данных, основными среди которых являются шардинг, репликация и комбинированный подход.

Шардинг (от англ. *shard* – осколок) – это техника масштабирования работы с данными, которая заключается в разбиении (партиционировании) базы данных на отдельные разделы (узлы), что позволяет располагать каждый из них на отдельном сервере. Главным вопросом, который необходимо решить для эффективного применения шардинга, является ассоциирование записей с узлами [1]. Основная задача – равномерно распределить данные и нагрузку между ними. В таком случае, теоретически, 10 узлов базы данных могут одновременно обрабатывать 10 запросов. Самый простой способ распределения данных это закрепление непрерывного промежутка ключей за одним узлом. Тогда, зная границы промежутков, можно всегда с лёгкостью определить раздел, в котором храниться запись. Однако такой подход обладает большим недостатком – закономерности в обращении к данным могут приводить к так называемым «горячим точкам» (*hot spots*), что сводит к нулю эффективность от применения шардинга. Другой способ, лишённый данного недостатка, – вычисление хэша ключа и закрепление промежутков значений хэш-функции за разделами базы данных. Это позволяет равномерно распределить данные между разделами, но обладает другим недостатком – существенно снижается эффективность запросов по промежуткам ключей, так как теперь нет никакой гарантии, что записи со смежными ключами будут находиться в одном разделе. Поэтому в современных СУБД, таких как Cassandra DB, используется гибридный подход с комбинированными ключами.

Другой популярный подход к масштабированию работы с данными – репликация. Суть подхода заключается в создании нескольких полных дубликатов базы данных, что позволяет снизить нагрузку на основной сервер и распределить её между несколькими копиями. Основная проблема, которую необходимо решить при применении репликации, это сохранение согласованности данных. Не всегда после добавления записи в БД можно сразу же извлечь её, так как возможна ситуация, когда запись была добавлена в одну копию БД, а запрос на чтение был отправлен к другой копии. Проблема согласованности данных усугубляется при большом количестве одновременных запросов на изменение данных, и её решению посвящено множество научных исследований. Самым простым (но далеко не самым эффективным) способом сохранения согласованности данных является деление всех копий БД на две группы – *master* и *slave*. При этом запросы на изменение данных обрабатываются только *master*-копиями. Если гарантировать, что последовательность запросов на изменение и на чтение будет обработана одной *master*-копией, то проблема будет решена.

Шардинг позволяет повысить быстродействие системы, но не оказывает никакого эффекта на её доступность. Репликация в свою очередь повышает доступность системы за счёт резервирования данных – если один сервер выходит из строя, то все запросы могут быть обработаны другими серверами. Чтобы сохранить преимущества обоих методов используется гибридный подход, в основе которого стоит шардинг, однако разделы реплицируются на несколько серверов. Применение гибридного подхода позволяет повысить как быстродействие системы обработки данных, так и её доступность.

В процессе выполнения дипломного проекта на тему «Программное средство организации работы франшизы тренажёрных залов» был сделан вывод, что применение шардинга позволит значительно улучшить пользовательский опыт, если в качестве критерия разбиения на разделы использовать географический регион, в котором расположен зал. Размещение разделов базы данных на серверах, расположенных в соответствующих регионах, существенно сократит время обработки запросов.

Использование вышеописанных подходов является важной частью проектирования высоконагруженных систем, позволяет повысить уровень удовлетворённости пользователей и достичь большего коммерческого успеха.

**Список использованных источников:**

1. Клеппман, М. *Designing Data-Intensive Applications* / М. Клеппман. — O'Reilly, 2017. — 613 с.