

ИСПОЛЬЗОВАНИЕ РЕПЛИКАЦИЙ В БАЗАХ ДАННЫХ

Гагарин Вячеслав Юрьевич
студент

Арташкин Евгений Павлович
студент

Мордовский государственный университет им. Н.П. Огарева, Саранск

Аннотация. В данной статье рассматриваются основные подходы при использовании репликаций. Объясняется, для чего они нужны, и какие задачи можно с помощью них решить.

Ключевые слова: база данных, оптимизация, репликация.

USE OF REPLICATIONS IN DATABASES

Gagarin Vyacheslav Yuryevich
student

Artashkin Evgeniy Pavlovich
student

Ogarev Mordovia State University, Saransk

Abstract. This article examines the main approaches when using replicas. It explains why they are needed, and what tasks can be solved with the help of them.

Keywords: database, optimization, replication.

Каждый год все больше сфер человеческой деятельности переходят на электронный вариант хранения данных. Для этого используются разнообразные системы управления базами данными (СУБД), такие как Oracle Database, Firebird, Interbase, IBM DB2, Informix, MS SQL Server, PostgreSQL, MySQL. Скорость работы современных реляционных СУБД позволяет изменять и считывать десятки тысяч строк информации в секунду. Базы данных намного упростили процесс ведения учета продаж, поиск информации, предоставление услуг посредством сети Internet и т.д. Такие возможности доступны благодаря беспереывно работающим удаленным системам. И в данных ситуациях совершенно невозможно себе даже представить, если по техническим причинам нельзя получить необходимую информацию. Чаще всего в таких случаях дают сбой именно базы данных. Для этого следует сделать систему отказоустойчивой. Один из способов разработки отказоустойчивых баз данных – создание их реплик.

Репликация (англ. replication) – механизм синхронизации содержимого нескольких копий объекта (например, содержимого базы данных). Репликация – это процесс, под которым понимается копирование данных из одного источника на другой (или на множество других) и наоборот. При репликации изменения, сделанные в одной копии объекта, могут быть распространены в другие копии. Репликация в базах данных – это тиражирование изменений данных с главного сервера баз данных на одном или нескольких зависимых серверах. Главный сервер при этом будет называться мастером, а зависимые сервера – репликами. Репликация может быть синхронной и асинхронной. Синхронной репликация называется, при обновлении данных сервера и всех других реплик того же фрагмента в одной и той же транзакции. Логически это означает, что существует лишь одна версия данных. В большинстве продуктов синхронная репликация реализуется с помощью триггерных процедур. Но синхронная репликация имеет тот недостаток, что она создаёт дополни-

тельную нагрузку при выполнении всех транзакций, в которых обновляются какие-либо реплики (кроме того, могут возникать проблемы, связанные с доступностью данных). Асинхронная репликация называется, в случае обновления данных одной реплики, которые распространяется на другие спустя некоторое время, а не в той же транзакции. Таким образом, при асинхронной репликации вводится задержка, или время ожидания, в течение которого отдельные реплики могут быть фактически неидентичными. В большинстве продуктов асинхронная репликация реализуется посредством чтения журнала транзакций или постоянной очереди тех обновлений, которые подлежат распространению. Преимущество асинхронной репликации состоит в том, что дополнительные издержки репликации не связаны с транзакциями обновлений, которое имеет важное значение для функционирования, и предъявлять высокие требования к производительности.

К недостаткам этой схемы относится то, что данные на репликах не всегда идентичны.

Изменения данных, происходящие на мастере, повторяются на репликах (но не наоборот). Поэтому запросы на изменение данных (добавление, изменение, удаление и т.д.) выполняются только на мастере, а запросы на чтение данных, могут выполняться как на репликах, так и на мастере. Процесс зеркалирования на одной из реплик не влияет на работу других баз данных, и практически не влияет на работу мастера. Репликация происходит при изменении журнала, который создается или изменяется при изменении данных на основном сервере базы данных. В журнал сохраняются все запросы к базе данных, приводящие к изменениям данных в ней. При этом запросы сохраняются не в явном виде, поэтому разобрать их вручную достаточно сложно и, как правило, задача решается при помощи специальных программных утилит. Важной особенностью является то, что при репликации передаются не сами измененные данные, а только кодированные запросы, вызывающие измене-

ния. После изменения, журнал, передается базам данных реплик, начиная с позиции $n+1$, где n – последняя выполненная позиция из бинарного лога. После этого журнал декодируется, и выполняются описанные в нем действия. Таким образом, при репликации содержимое базы данных дублируется на одном или нескольких серверах как показано на рисунке. Создание реплик решает сразу несколько задач по повышению надежности и производительности системы.

- *Производительность и масштабируемость.* Если один сервер не справляется с нагрузкой, создаваемой одновременными операциями чтения и записи в базу данных. Запросы на чтение данных можно перенаправить на реплики, при этом выгода такой системы будет весомей, если на одну операцию записи происходит несколько операций чтения.
- *Отказоустойчивость.* При отказе одной из реплик, запросы чтения можно распределить по другим репликам или перевести все запросы на мастер, если была лишь одна реплика. В случае отказа мастера, запросы на запись можно перевести на одну из реплик, а после того как мастер будет восстановлен, его можно сделать репликой.
- *Резервирование данных.* При использовании реплики мы всегда уверены что есть, «страховочная» база данных, практически идентично повторяющая основную.
- *Сложные вычисления.* Сложные и медленные запросы к базе данных можно выполнять на отдельной реплике, не замедляя работу всей системы.
- *Модернизация оборудования.* При необходимости «остановить» один из серверов, для установки, например дополнительной оперативной памяти, не нарушится работа системы в целом.



Рис. 1. Репликация базы данных

- *Возможность использования различных структур таблиц на разных серверах.* К примеру, на реплике базы данных, для сложных вычислений, можно выставить альтернативный набор индексов или изменить тип таблицы. У рассматриваемого процесса репликации есть также ряд недостатков.
- *Не точная копия в единицу времени.* При асинхронной репликации существует расхождение данных мастера от реплик. Так как при добавлении записи в БД мастера производится обновление журнала, который впоследствии передается репликам, декодируется и выполняется. На практике расхождение в базах данных может достигать до двух секунд.
- *Проблемы с random значением.* Если в запросе использовалась функция random с неопределенным значением, то на реплике это значение может быть отличным от базы данных на мастере.

При создании инфраструктуры базы данных на основе метода репликации, не критичен отказ главного сервера. Работоспособность системы можно восстановить за несколько минут в ручную. Можно создать контроллер, который будет выполнять эти действия автоматически, сде-

лав одну из реплик главным сервером и заменив настройки остальных реплик таким образом, чтобы они забирали журнал с нового, главного сервера, баз данных.

Метод можно улучшить при помощи создания отдельного контролера, через который будут проходить все запросы к базам данных. И именно он будет следить за распределением нагрузки по доступности реплик. В итоге, не будет главного сервера базы данных (мастер), а запросы на выборку данных будут распределяться только по тем репликам, у которых данные для запроса актуальны.

Список использованных источников

1. Аббакумов А.А., Акимов В.Л., Егунова А.И., Лещанкин К.А., Таланов В.М. Базы данных (MS ACCESS, MYSQL). Саранск: Изд-во Мордов. ун-та, 2011. С. 5.
2. Александров Э.Э., Афонин В.В. Программирование на языке С в Microsoft Visual Studio 2010 [Электронный ресурс]. Режим доступа: <http://www.intuit.ru/department/pl/prcmsvs2010>
3. Таненбаум Э. Современные операционные системы. СПб., 2010. 972 с.
4. Прата С. Язык программирования С. Лекции и упражнения. М.: Вильямс, 2006. 256 с.