УПРАВЛЕНИЕ И АНАЛИЗ ИНФОМАРЦИИ В РАСПРЕДЕЛЁННЫХ CRDT БАЗАХ ДАННЫХ

В. В. Бернацкий
Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: bernatskiv@gmail.com

Рассмотрены ключевые аспекты CRDT баз данных, описаны некоторые особенности их использования, разработан механизм для управления и анализа информации в распределённых CRDT базах данных, предложено два способа реализации данного механизма

Введение

В настоящее время получают широкое применение распределённые базы данных типа «ключ-значение». Ими пользуются компании предоставляющие сервисы в глобальном масштабе, такие как Facebook [1], Yahoo [2], Amazon [3]. Ради повышения качества оказываемых услуг датацентры располагают в различных точках мира [3], [2], [4], чтобы уменьшить время доступа к данным за счет маршрутизации запросов к датацентру расположенному ближе к потребителю. Однако для случаев доступа в одним и тем же данным из различных точек мира это вызывает их репликацию, что ведет к необходимости решения проблем, связанных с обеспечением согласованности данных, их быстрой доступности и устойчивости сети к разделению. Как правило в большинстве баз данных реализованы модели с меньшей согласованностью, ради обеспечения большей доступности. Но при одновременном изменении одних и тех же данных на различных репликах возникают проблемы при синхронизации состояний этих реплик. Необходим механизм для разрешения подобных конфликтов. Одним из наиболее популярных решений является принцип «последняя запись побеждает» [5], [6], [1], при котором сохраняется версия данных записанная позже. Но данный способ не является универсальным. Существуют варианты использования, для которых этот сценарий является неприемлемым, к ним относятся различные счетчики. Одним из вариантов решения данной проблемы является использование баз данных с реализованными conflict-free replicated data type (CRDT).

I. Описание CRDT и её особенностей

Существует вид согласованности данных называемый eventual consistency. Он означает, что если в системе реализующий такой подход прекратить изменение данных на всех репликах, то в конечном счете данные на всех репликах придут в эквивалентное состояние. Strong eventual consistency накладывает еще одно ограничение: реплики получившие одинаковые изменения сразу приходят в эквивалентное состояние.

В CRDT предполагается, что система обеспечивает SEC и её состояния монотонно прогрессируют, не приводя к конфликтам. Монотонность в этом смысле означает отсутствие откатов: операции нельзя отменить, вернув систему в раннее состояние. Состояния такой системы связаны отношением частичного порядка, в математике такая система с определённой на ней операцией объединения называется полурешёткой [7].

CRDT системы предъявляют следующие требования к операциям разрешенным над хранимыми типам данных:

- Они должны быть идемпотенты, т.е. повторное применение одной и той же операции должно приводить данные в одно и то же состояние;
- Они должны быть коммутативны;
- Они должны быть ассоциативны.

Выполнение данных требований приводит к тому, что последовательность состояний данных представляют собой полурешётку и гарантируют сходимость к одному результату.

Как видно из требований CRDT подход накладывает серьезные ограничения на используемые типы данных. Эти структуры данных не всегда применимы к реальным задачам в чистом виде. В некоторых случаях необходима возможность создания временной блокировки части данных для гарантии какого-либо инварианта необходимого для выполнения определённого набора операций. Также CRDT концепция не предоставляет механизмов для анализа данных на отдельных репликах.

II. Распределённое выполнение функций

Для решения данной проблемы можно ввести новый тип функций, обладающий следующими свойствами:

- Каждая функция выполняется на каждой реплике один раз;
- Функция распространяется от реплики к реплике по тому же протоколу, что и другие данные CRDT;

– Результаты функции записываются в коллекцию ключ-значение, где в качестве ключа выступает ID реплики, а в качестве значения – результат выполнения функции;
– Коллекция результатов удовлетворяет требованиям CRDT, поэтому она распространяется по репликам как обычная CRDT структура данных.

В общем случае данную функциональность можно заключить в структуру данных содержащую следующие поля: ID, тело функции (скрипт или исполняемый файл), входные данные, коллекция результатов.

Применимо к практике такой тип функций можно использовать в различных сценариях:

– Получить информацию о каждой отдельной реплике или её окружении (например, подсчитав хэщь каждой реплики удостовериться, что они к пришли к единому состоянию);
– Воздействовать на данные в репликах (например, включить блокировки на обновление каких-либо данных для обеспечения сохранности инварианта);
– Выполнять любые вычисления и обработку информации, не связанные с работой самой БД.

III. Способы реализации

Для реализации данной функциональности в данной работе предлагается два варианта.

Написание «обертки» над существующей CRDT базой данных. В данном способе разрабатываемое ПО будет представлять собой отдельное приложение, которое общается с отдельными узлами CRDT базы данных. Каждому узлу БД будет соответствовать один экземпляр «обертки» и он будет находится на той же машине, что и узел БД. Для поддерживания работы этого решения в самой БД будет коллекция, содержащая структуру данных описанную выше. «Обертка» будет наблюдать за добавлением новых элементов (но она не будет отлавливать изменения происходящие во вложенной коллекции результатов) и при появлении новых функций будет исполнять её, а результат записывать в коллекцию результатов. Таким образом при минимальной реализации данному ПО необходим лишь доступ к узлу CRDT БД находящимся на той же машине, всю нагрузку на обмен информацией и сбором результатов берет на себя структура сама база данных.

Вторым вариантом является реализации данной функциональности в рамках самой CRDT БД. Структура данных для хранения информации о функции и её результатах та же, что и в предыдущем варианте. Механизм работы аналогичен.

При сравнении этих способов реализации можно выделить следующие достоинства первого способа: более быстрая реализация, возможность изменения или замены данного ПО независимо от самой БД, возможность использовать с существующими и проверенными CRDT БД, остановка работы приложения, вызванная ошибкой в функции, не повлияет на работу БД.

К достоинствам второго способа можно отнести: более высокая скорость обмена данными между базой и функцией, возможность реализации событий добавления нового элемента в коллекцию функций вместо регулярной проверки с заданным интервалом, отсутствие необходимости запускать дополнительного ПО на каждом узле, особенно это важно при переменном количестве узлов.

Таким образом был разработан механизм для управления и анализа информации находящейся в CRDT базах данных, а также. Данное решение позволяет расширить круг использования баз данных подобного рода, анализировать состояние отдельных реплик и информации находящейся в ней, а также использовать данное решение для вычислений не связанных с работой самой БД.

Были разработаны два способа для реализации данного механизма, каждый из которых обладает своим набором достоинств и недостатков и выбор конкретного варианта зависит требований предъявляемыми конкретной задачей.

Список литературы

1. LAKSHMAN, A. Cassandra: A decentralized structured storage system / A. LAKSHMAN, P. MALIK// SIGOPS Oper. Syst. — 2010, P. 35 – 40.
2. PNUTS: Yahoo!'s Hosted Data Serving Platform / B. F. COOPER, R. RAMAKRISHNAN, U. SRIVASTAVA, A. SILBERSTEIN, P. BOHANNON, H.-A. JACOBSEN, N. PUZ, D. WEAVER, R. YERNENI // VLDB Endow. 1 - 2008
3. Dynamo: Amazon's highly available key-value store / G. DECANDIA, D. HASTORUN, M. JAMPANI, G. KAKULAPATI, A. LAKSHMAN, A. PILCHIN, S. SIVASUBRAMANIAN, P. VOSSHALL, W. VOGELS // SOSP '07 - 2007.
4. Transactional storage for geo-replicated systems / Y. SOVRAN, R.POWER, M. K. AGUILERA, J. LI // SOSP '11 - 2011.
5. Don't settle for eventual: Scalable causal consistency for widearea storage with cops / W. LLOYD, M. J. FREEDMAN, M. KAMINSKY, D. G. ANDERSEN // SOSP '11 - 2011.
6. Stronger semantics for low-latency geo-replicated storage / W. LLOYD, M. J. FREEDMAN, M. KAMINSKY, D. G. ANDERSEN // nsdi'13 - 2013.
7. Репликация без конфликтов: CRDT в теории и на практике [Электронный ресурс]. - Режим доступа: https://habrahabr.ru/post/272987/ - Дата доступа: 01.06.2016