

БАЗЫ ДАННЫХ NoSQL

КУРЧАН Павел

Coordonator: САРАНЧУК Дориан
Технический Университет Молдовы

Аннотация: В данной статье рассматривается концепция баз данных NoSQL, раскрываются некоторые особенности NoSQL, преимущества использования неструктурированных баз данных. Описываются типы хранилищ данных NoSQL, их предназначение. Вкратце описывается NoSQL система управления базами данных MongoDB, ее особенности и функциональные возможности.

Ключевые слова: NoSQL, система управления базами данных, MongoDB, хранилище данных.

1. ВВЕДЕНИЕ

Система управления базами данных (СУБД) – это совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление, создание и использование баз данных. Наиболее популярными являются реляционные СУБД.

Реляционная модель ориентирована на представление данных в виде двумерных таблиц. Такое представление удобно для пользователей. Однако, реляционные базы данных не могут справляться со всеми нагрузками, актуальными в наше время. Тремя основными проблемами реляционных СУБД являются:

- горизонтальное масштабирование при больших объемах данных;
- производительность каждого отдельного сервера;
- не гибкий дизайн логической структуры.

Это заставляет разработчиков искать альтернативу реляционным базам данных, используемым вот уже более тридцати лет. В совокупности все эти технологии известны как «NoSQL базы данных».

2. ОСНОВНЫЕ КОНЦЕПЦИИ NOSQL

Важной вехой для высоконагруженных систем стало развитие Интернета, ряд сервисов которого (DNS-серверы, поисковые машины, социальные сети и т.д.) изначально должны обрабатывать большие массивы информации и отвечать на огромное число запросов. Это требует не только максимального учета любой специфики обрабатываемой информации, но и перехода на распределенные вычисления. Никакой сколь угодно мощный сервер в принципе не способен в одиночку обеспечить нужную производительность. В итоге появилась концепция NoSQL, основными принципами которой стали: отказ от реляционной модели для учета специфики обрабатываемых данных, открытый исходный код, а также хорошая горизонтальная масштабируемость до сотен и тысяч серверов для обеспечения скорости работы.

NoSQL (англ. Not only SQL, не только SQL) – термин, обозначающий ряд подходов, проектов, направленных на реализацию моделей баз данных, имеющих существенные отличия от используемых реляционных СУБД с доступом к данным средствами языка SQL. Основная цель подхода – расширить возможности БД там, где SQL недостаточно гибок, и не вытеснять его там, где он справляется со своими задачами.

Масштабируемость предполагает автоматическое распределение между несколькими серверами. Такие системы называются распределенными (распределение имеет место между компьютерами по локальной сети). Однако это выявило еще одну проблему, сформулированную в виде теоремы CAP (Consistence, Availability, Partition tolerance — «согласованность, доступность, устойчивость к разделению»): в распределенной вычислительной системе невозможно одновременно выполнить требования по согласованности данных, доступности системы и устойчивости к разделению. Под последним требованием понимается то, что система не распадается на несколько изолированных секций, внутри которых выполняются требования по согласованности и доступности.

Нестрогое доказательство теоремы CAP основано на простых рассуждениях. Пусть распределенная система состоит из N серверов, каждый из которых обрабатывает запросы некоторого числа клиентских приложений (рис.1). При обработке запроса сервер должен гарантировать актуальность информации, содержащейся в отсылаемом ответе на запрос, для чего предварительно нужно выполнить синхронизацию содержимого его собственной базы с другими серверами. Таким образом, серверу необходимо ждать полной синхронизации либо генерировать ответ на основе не синхронизированных данных. Возможен и третий вариант, когда по каким-либо

причинам синхронизация производится только с частью серверов системы. В первом случае оказывается не выполненным требование по доступности, во втором — по согласованности, в третьем — по устойчивости к разделению.

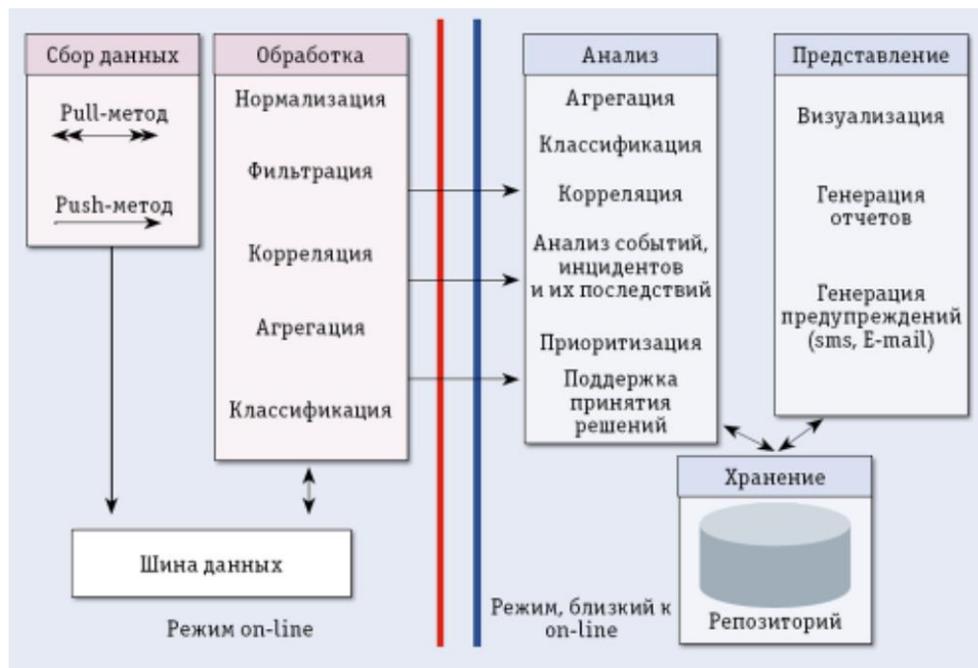


Рис. 1. Выполнение запроса к распределенной системе

К базам данных NoSQL предъявляются следующие требования :

– базовая доступность

(<http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9%D1%8F%D0%B7%D1%8B%D0%BA>*basic availability*)— каждый запрос гарантированно завершается (успешно или безуспешно);

– гибкое состояние

(<http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9%D1%8F%D0%B7%D1%8B%D0%BA>*soft state*)— состояние системы может изменяться со временем, даже без ввода новых данных, для достижения согласования данных;

– согласованность в конечном счёте

(<http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9%D1%8F%D0%B7%D1%8B%D0%BA>*eventual consistency*)— данные могут быть некоторое время рассогласованными, но приходят к согласованию через некоторое время.

3. СИСТЕМЫ NOSQL

NoSQL-системы могут успешно решать множество различных задач, среди которых можно выделить следующие: хранение XML-документов, хранение пар «ключ – значение», хранение графов, хранение кортежей произвольной длины, Triple Storages, многомерные данные и т. д.

Хранилища XML-документов (BaseX, eXist) представляют собой средства для работы с большим количеством XML-документов или с документами большого размера. Информация содержится не в текстовом виде, а в некотором внутреннем формате, позволяющем быстро выполнять операции поиска (запросы XPath и XQuery) и изменения документов (XSLT, eXtensible Stylesheet Language Transformations).

Формат XML — не единственный способ текстового представления структурированной информации: по аналогии с XML-хранилищами существуют СУБД (Apache Couch DB и MongoDB) для работы с данными, представленными в виде JSON (Java Script Object Notation) или BSON (Binary JSON).

Системы хранения графов и Triple Storages ориентированы на работы с семантическими данными, представленными в виде узлов и дуг. Отличительной чертой Triple Storages является то, что они ориентированы на поддержку стандартов SemanticWeb.

Графовые базы данных применяются для задач, в которых данные имеют большое количество связей, например, социальные сети, выявление мошенничества. Примерами могут служить Neo4j (англ.), Bigdata (RDF-хранилище), InfiniteGraph.

В хранилище семейств колонок данные хранятся в виде разреженной матрицы, строки и столбцы которой используются как ключи. Типичным применением этого вида СУБД является веб-индексирование, а также задачи, связанные с большими данными, с пониженными требованиями к согласованности данных.

Хранилища «ключ-значение» является простейшим хранилищем данных, использующим ключ для доступа к значению. Такие хранилища используются для хранения изображений, создания специализированных файловых систем, в качестве кэшей для объектов, а также в системах, спроектированных с прицелом на масштабируемость.

Хранение пар «ключ – значение» заключается в реализации двух операций: запись информации по ключу и чтение по ключу — при этом одному ключу может соответствовать сразу несколько значений. Востребованность данной функциональности при построении высоконагруженных систем привела к тому, что появилось целое множество соответствующих СУБД, которые, в зависимости от способа реализации, можно разделить на постоянные (CDB), редко изменяющиеся (Apache Cassandra, membase, MemcacheDB) и часто изменяющиеся (memcached). Постоянные СУБД не поддерживают изменения информации «на лету» — база данных создается один раз и затем используется продолжительное время в режиме чтения. Это, во-первых, позволяет провести глубокий анализ содержимого базы на этапе ее создания и обеспечить высокую степень компрессии данных, что впоследствии позволит минимизировать ввод/вывод и тем самым ускорить операции поиска. Во-вторых, если база не изменяется, то упрощается и ускоряется многопользовательская работа, поскольку не требуется выполнять никаких блокировок.

Системы для работы с часто изменяющимися данными всю информацию содержат только в оперативной памяти и вообще не используют ввода/вывода. Такой подход популярен при реализации механизмов кэширования, но принципиально не подходит при построении основного хранилища, поскольку любая авария питания приведет к потере данных. Наконец, системы с редко изменяющимися данными являются некоторым промежуточным звеном, обеспечивая возможность корректировки базы, достаточно высокую скорость поиска и сохранение информации на диске.

Часть хранилищ пар «ключ – значение» основаны на хэш-таблицах, а часть — на B-деревьях (BerkleyDB и MemcacheDB), что помимо выполнения операции поиска дает возможность упорядочивать ключи по возрастанию или убыванию.

В общем случае системы рассматривают ключ и значение как массив байтов, причем длина ключа может достигать нескольких килобайтов, а значения — нескольких мегабайтов. Однако часто некоторые СУБД предоставляют дополнительные возможности для структурирования ключей и значений. Так, Apache Cassandra позволяет разбивать значения на несколько колонок и работать с каждой из них отдельно, а BigTable, напротив, использует трехкомпонентные ключи (ключ столбца, ключ строки и временная метка), но значения рассматривает как массив байтов. Такое устройство ключа позволяет использовать BigTable как средство работы с двумерными таблицами большого размера. Примерно по такой же схеме работают хранилища кортежей произвольной длины.

Все эти классы систем NoSQL очень разнородны, и пока отсутствует единый стандарт требований к ним, однако такая стандартизация станет необходима по мере создания новых высоконагруженных систем и повышения требований к скорости обработки и стоимости разработки. Наличие единых стандартов облегчит использование нескольких NoSQL СУБД в одном проекте, упростит интеграцию СУБД между собой, ускорит миграцию на новые СУБД, сделает возможным создание универсальных инструментальных средств NoSQL, позволит заранее осуществлять подготовку специалистов по NoSQL и т. д.

В 2011 году был сделан первый шаг в направлении стандартизации — анонсирован язык запросов UnQL (Unstructured Query Language) для работы с неструктурированной информацией. Для удобства прикладных разработчиков синтаксис и семантика языка во многом схожи с SQL, что вполне естественно — благодаря поддержке UnQL каждая NoSQL-СУБД может стать гибким, удобным и легко используемым инструментом, и она по-прежнему будет основываться на собственных технических решениях, дающих преимущество в каких-либо условиях эксплуатации.

В качестве примера можно указать систему для хранения пар «ключ – значение». Если все СУБД поддерживают UnQL, то прикладной разработчик может подобрать оптимальную, практически не меняя кода приложения. Например, если необходима организация кэширования, то лучше использовать СУБД, ориентированную на работу в памяти (например, membase). Если необходима работа с редко изменяющимися данными, то имеет смысл выбрать СУБД, предназначенную для работы именно в таких условиях (например, CDB). Если помимо поиска необходима сортировка ключей, то подойдет система, основанная на B-деревьях (например, BerkleyDB).

4. NOSQL СУБД MONGODB

MongoDb – кроссплатформенная документо-ориентированная система управления базами данных с открытым исходным кодом, не требующая описания схемы таблиц. Написана на языке C++ и распространяется в рамках лицензии AGPLv3.

Основные возможности данной СУБД:

- документо-ориентированное хранилище;
- достаточно гибкий язык для формирования запросов;
- полная поддержка индексов (объектов базы данных, созданных с целью повышения производительности поиска данных);
- профилирование запросов;
- быстрые обновления «на месте»;
- эффективное хранение двоичных данных больших объемов, например, фото и видео;
- журналирование (запись событий) операций, модифицирующих данные в БД;
- может работать в соответствии с парадигмой Mapreduce.

СУБД управляет наборами JSON-подобных документов, хранимых в двоичном виде в формате BSON. Хранение и поиск файлов в MongoDB происходит благодаря вызовам протокола GridFS. Подобно другим документо-ориентированным СУБД (CouchDB, пр.), MongoDB не является реляционной СУБД.

ЗАКЛЮЧЕНИЕ

Движение NoSQL заметно возросло благодаря увлечению большого количества компаний связанных с использованием больших объемов данных. Появляется все больше систем, позволяющих организовывать и поддерживать огромные массивы данных, обрабатывать и контролировать эти данные. NoSQL - лучшее решение для документо-ориентированных систем, для объектно-ориентированных систем, для распределенных систем, требовательных к параметрам горизонтальной масштабируемости и небольшому времени отклика.

Два направления развития СУБД — SQL и NoSQL — во многом противопоставляются, хотя и подчиняются единым законам и движутся навстречу друг другу. Пока это движение почти незаметно, но оно активизируется по мере появления необходимости в средствах построения высоконагруженных систем, в результате чего произойдет слияние этих направлений.

ЛИТЕРАТУРА

1. Wikipedia, Wikimedia Foundation, 2003. <http://ru.wikipedia.org/wiki/NoSQL> [15.06.2014].
2. Wikipedia, Wikimedia Foundation, 2003. <http://ru.wikipedia.org/wiki/Реляционная СУБД> [15.06.2014].
3. Habranabr, 2006. <http://habrahabr.ru/blogs/nosql/77909/> Обзор NoSQL систем [15.06.2014].