Лекция 9. Распределенные системы для BigData

Если оглянуться назад, до появления электронных компьютеров, термин "компьютер" означал человека, выполнявшего численные расчеты. Поскольку на протяжении большей части истории человечества использовались целые комнаты таких компьютеров, возможно, распределенные (т.е. многоместные) вычисления появились гораздо раньше, чем принято считать.

Современные распределенные вычисления, вероятно, берут свое начало от попыток создания кластеров компьютеров "Беовульф"[1] в 1990-х годах и от ранних вычислений ad hoc. Программное обеспечение на компьютерах затем тесно взаимодействовало друг с другом по сети, чтобы разделить работу, которую нужно было выполнить на всех компьютерах. Ранние специальные вычисления, включая distributed.net и SETI@Home, использовали незанятые домашние компьютеры для поиска криптографических ключей и инопланетных радиосигналов в радиоастрономических данных. Ни у одного из этих проектов не было ни вычислительных мощностей для решения проблемы, ни достаточного бюджета для этого. Однако проблемы были достаточно просты, чтобы большое количество компьютеров могло добиться значительного прогресса при минимальной координации между ними. Домашние компьютеры под управлением программ distributed.net или SETI@Home связывались с главным сервером, чтобы получить часть работы, которую необходимо выполнить (ключи для проверки или данные радиосигналов для изучения).

Классические суперкомпьютеры были очень большими, очень дорогими машинами, которые содержали специализированное быстродействующее оборудование и процессоры. Эти суперкомпьютеры содержали в своем специализированном оборудовании такие функции, как несколько центральных процессоров и векторное оборудование. Векторное оборудование используется для выполнения одинаковых операций над несколькими частями данных. Например, двумерный вектор имеет компоненты X и Y. Сложение набора векторов выполняется по компонентам, поэтому выполнение одной инструкции, которая добавляет одновременно X и Y, может привести к удвоению скорости обработки.

Стоимость аппаратного обеспечения значительно снизилась. Системы с большим количеством ядер/большой памятью (массивно-параллельная обработка [MPP]) и кластеры умеренных систем позволяют реализовать гораздо более крупные проекты по добыче данных. Для аналитики больших данных единственным решением является перенос аналитики в данные; перенос данных в аналитику непрактичен из-за времени, необходимого для передачи данных.

Теперь могут быть рассмотрены для анализа более сложные проблемы - проблемы, которые потребляют гораздо большие объемы данных, с гораздо большим количеством переменных. Кластерные вычисления можно разделить на две основные группы распределенных вычислительных систем. Первая - это база данных, которая уже несколько десятилетий является неотъемлемой частью центров обработки данных. Вторая - распределенные вычислительные системы, среди которых в настоящее время доминирует Hadoop.

Вычисления в базах данных

Реляционная система управления базами данных (RDBMS) или просто база данных существует с 1970-х годов и до недавнего времени была наиболее

---

[1] Кластеры компьютеров Beowulf представляли собой стандартные серверные или даже настольные компьютеры, объединенные в сеть.

распространенным местом хранения данных, создаваемых организациями. Ряд крупных поставщиков, а также проекты с открытым исходным кодом предоставляют системы баз данных. Традиционные РСУБД предназначены для поддержки баз данных, размер которых значительно превышает объем памяти или хранилища, доступного на одном компьютере. С тех ранних времен сегодня существует множество различных баз данных, которые служат специальным целям для организаций, связанных с высокопроизводительным поиском данных и анализом больших данных.

Базы данных в памяти (IMDB) были разработаны в 1990-х годах. В настоящее время IMDB являются популярным решением, используемым для ускорения критически важных операций с данными в финансовой сфере, электронной коммерции, социальных сетях, информационных технологиях и других отраслях. Идея технологии IMDB проста – хранение данных в памяти, а не на диске, повышает производительность. Однако у IMDB есть заметные недостатки, а именно повышенная стоимость и нестабильность памяти.

Базы данных MPP (massively parallel processing) начали развиваться из традиционных технологий СУБД в 1980-х годах. Базы данных MPP призваны служить многим из тех же операционных, транзакционных и аналитических целей, что и предыдущее поколение коммерческих баз данных, но предлагают функции производительности, доступности и масштабируемости, предназначенные для обработки больших объемов данных при использовании стандартных пользовательских интерфейсов. Базы данных MPP позиционируются как наиболее прямое обновление для организационных корпоративных хранилищ данных (EDW). Технология, лежащая в основе баз данных MPP, обычно включает кластеры товарных или специализированных серверов, которые хранят данные на нескольких жестких дисках.

Большим преимуществом вычислений на базе данных является экономия времени на перемещение данных в аналитику. В случае действительно больших данных время на перемещение данных и аппаратные ресурсы, необходимые для их эффективной обработки после перемещения, делают стратегию неэффективной. Использование программного обеспечения, которое может перемещать аналитику к данным и обрабатывать их на месте, используя большие вычислительные ресурсы, которые предоставляет распределенная база данных, приведет к созданию более быстрых моделей и сокращению времени выполнения по сравнению с нераспределенными вычислительными системами.

Вычисления в файловых системах

Существует множество вариантов выбора платформы для вычислений на файловых системах, но рынок быстро консолидируется на Hadoop с его многочисленными дистрибутивами и инструментами, совместимыми с его файловой системой, такими как Hive, MapReduce и HBase.

Hadoop был создан Дагом Каттингом и Майком Кафареллой. (см. рис. 1.) Название "Hadoop" не является аббревиатурой или ссылкой на что-то конкретное. Оно произошло от имени, которое сын Каттинга дал чучелу желтого слона. Изначально он был разработан в 2004 году на основе работы Каттинга и Кафареллы над Nutch[2] и статьи, опубликованной Google, в которой была представлена парадигма MapReduce для обработки данных на больших кластерах. В 2008 году Hadoop стал полноценным проектом Apache и использовался несколькими компаниями, занимающимися обработкой больших данных, такими как Yahoo!, Facebook и The New York Times.

[2] Nutch - это проект Apache с открытым исходным кодом для поиска информации в Интернете.

Современная тенденция заключается в хранении всех доступных данных в Hadoop. Hadoop привлекателен тем, что он может хранить и управлять очень большими объемами данных на обычном оборудовании и может легко расширяться за счет добавления аппаратных ресурсов с постепенным увеличением затрат. Традиционно размер систем определялся исходя из ожидаемых объемов данных, без расчета на то, что данные будут накапливаться бесконечно. Hadoop сделал возможным крупномасштабное накопление данных и потенциально является существенным отличительным фактором для конкурентного преимущества. Те, кто сможет успешно использовать ценность исторических данных, смогут получить огромное преимущество в будущем. Hadoop становится передовым решением для хранения больших объемов исторических данных. Однако эти данные редко (фактически, вероятно, никогда) находятся в форме, пригодной для анализа данных.

Кроме того, для решения многих задач используются другие хранилища данных, дополняющие данные в Hadoop. Например, транзакции по кредитным картам могут храниться в Hadoop, но информация о счетах владельцев карт может храниться и поддерживаться в традиционной базе данных. Этап исследования данных включает в себя определение данных и хранилищ данных, которые будут использоваться в процессе моделирования. Затем данные должны быть объединены, обобщены и сохранены для анализа данных. Эта работа обычно выполняется в Hadoop, поскольку вычислительные затраты на единицу продукции ниже, чем в MPP или RDBMS. Такой гибридный подход к хранению данных, вероятно, будет общепринятой практикой в течение многих лет, пока либо Hadoop не достигнет уровня баз данных, либо не появится другая технология, которая сделает базы данных и Hadoop менее желательными.

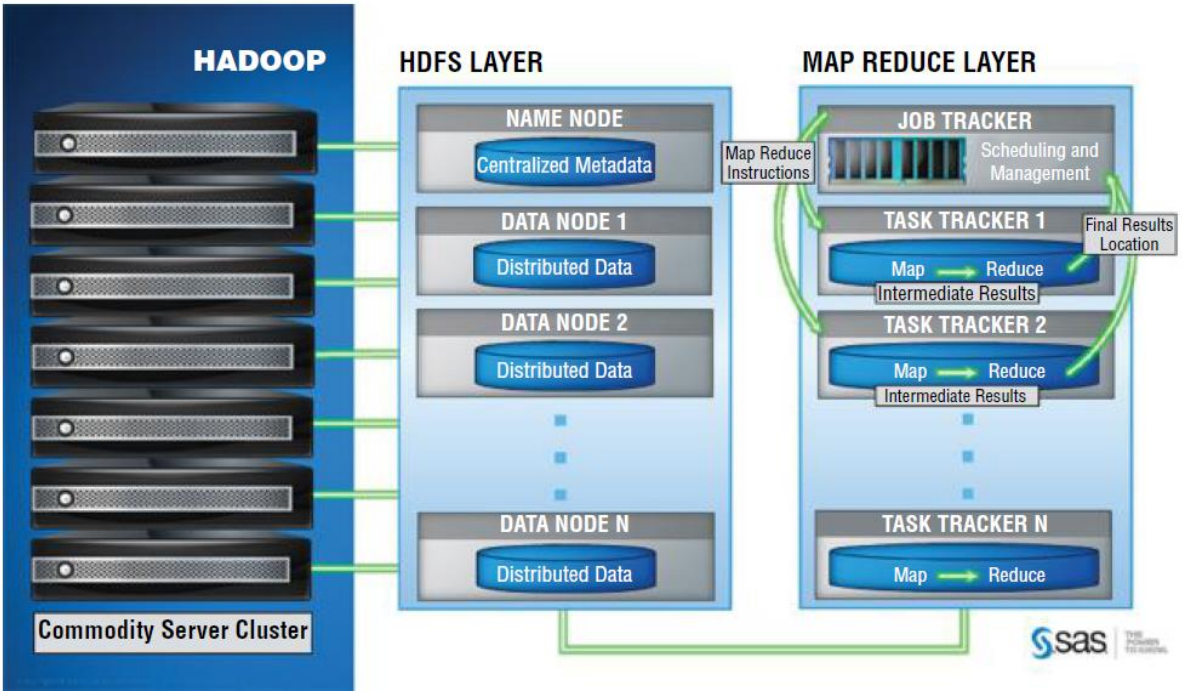


Рис 1 Графическая иллюстрация системы Hadoop

Облачные вычисления, поскольку они способны быстро предоставлять новые серверы или выводить из эксплуатации серверы, которые больше не нужны, могут использоваться в любом качестве. Однако не все высококлассные "фишки", такие как специализированное сетевое оборудование, доступны. Их гибкость позволяет быстро

переключаться между простыми вычислениями ad hoc и скоординированными вычислениями и даже смешивать эти модели на лету.

Соглашения

Вот несколько вопросов, которые следует рассмотреть при выборе высокопроизводительной платформы для добычи данных.

– Каков размер ваших данных в настоящее время?

– Каковы ожидаемые темпы роста объема ваших данных в ближайшие несколько лет?

– Хранятся ли у вас в основном структурированные или неструктурированные данные?

– Какое перемещение данных потребуется для завершения ваших проектов по поиску данных?

– Какой процент ваших проектов по поиску данных может быть решен на одной машине?

– Подходит ли ваше программное обеспечение для добычи данных для проектируемой вами вычислительной среды?

– Какие самые большие жалобы ваших пользователей на текущую систему?

На рисунке 2 сравнивается ряд технологий больших данных. На рисунке показаны различные типы систем и их сравнительные достоинства и недостатки. Решение о покупке вычислительной платформы для добычи данных, скорее всего, будет принимать ИТ-организация, но сначала она должна понять, какие проблемы решаются сейчас и какие проблемы необходимо решить в ближайшем будущем. Рассмотрение компромиссов между платформой и потребностями организации при прозрачном обмене информацией приведет к наилучшему результату для всей организации и отдельных заинтересованных сторон.

	In-Memory Database	MPP Database	Big Data Appliance	Hadoop	NoSQL Database
Consistent	●	●	●	▲	▲
Available	●	●	●	▲	▲
Fault tolerant	●	●	▲	●	●
Suitable for real-time transactions	●	●	●	◆	◆
Suitable for analytics	▲	▲	●	●	◆
Suitable for extremely big data	◆	▲	▲	●	●
Suitable for unstructured data	◆	◆	▲	●	●

Рис. 2 Сравнение технологий больших данных

На рисунке 2 символы имеют следующее значение:

■ Круг: Соответствует широко распространенным ожиданиям.

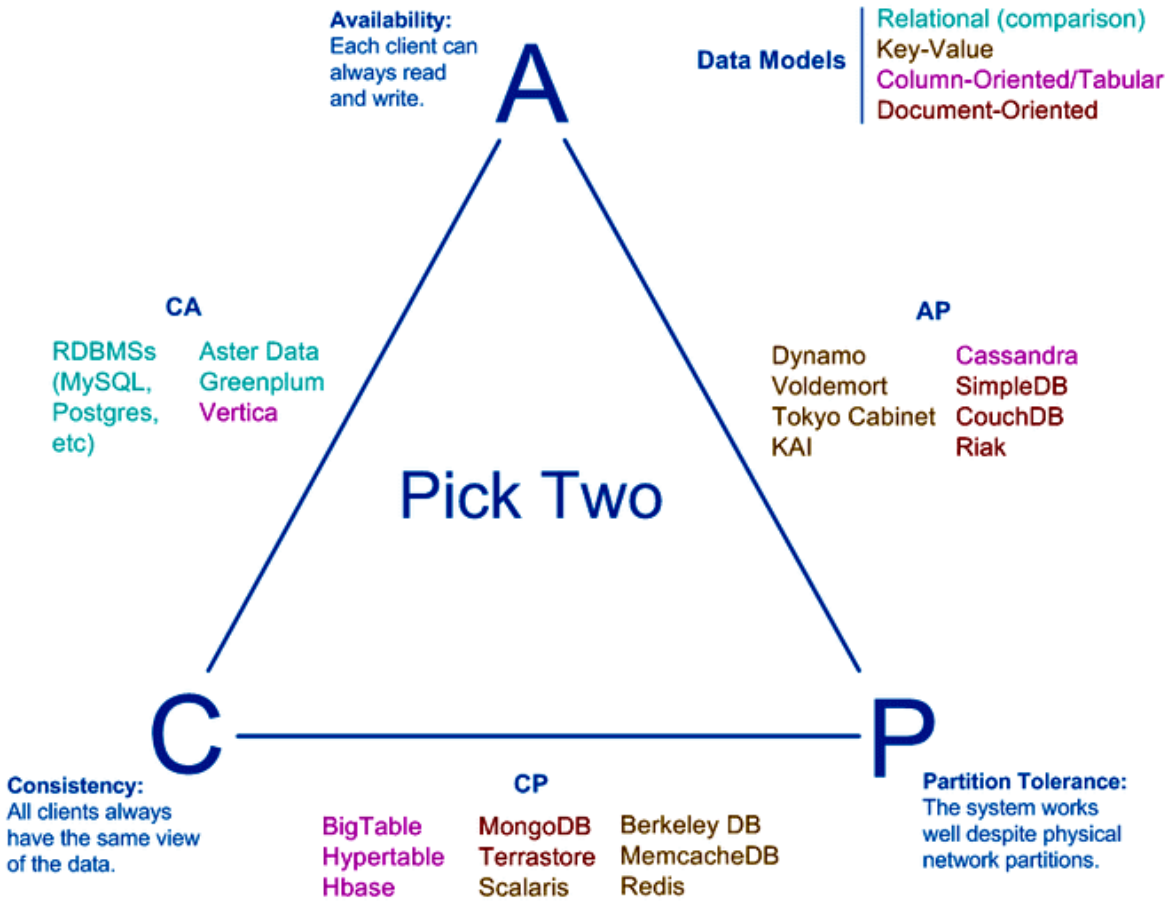
■ Треугольник: Потенциально соответствует широко распространенным ожиданиям.

■ Ромб: Не соответствует широко распространенным ожиданиям

Проекты по работе с большими данными/добыче данных должны учитывать, как данные перемещаются по всему сквозному процессу.

Вычислительная среда имеет решающее значение для успеха: Ваша вычислительная среда включает в себя четыре важных ресурса: сеть, диск, центральные процессоры (CPU) и память. Время достижения целей решения, ожидаемые объемы данных и бюджет будут определять ваши решения относительно вычислительных ресурсов. Подходящие вычислительные платформы для анализа данных зависят от многих параметров, в первую очередь от объема данных (начальный объем, рабочий набор и объем выходных данных), схемы доступа к данным и алгоритма анализа. Они будут варьироваться в зависимости от этапа анализа данных [1].

CAP



CAP – это акроним от англоязычных слов Consistency (Согласованность, Целостность), Availability (Доступность) и Partition tolerance (Устойчивость к разделению). Согласно утверждению профессора Калифорнийского университета в Беркли, Эрика Брюера, сделанному в 2000-м году, в распределенных системах осуществимы лишь 2 свойства из указанных 3-х. В частности, считается что нереляционные базы данных жертвуют согласованностью данных в пользу доступности и устойчивости к разделению, когда расщепление распределённой системы на несколько изолированных частей сохраняет корректный отклик от каждой из них **[1]**. В 2002 году Сет Гилберт и Нэнси Линч из MIT опубликовали формальное доказательство гипотезы Брюера, после чего она стала считаться теоремой **[2]**.

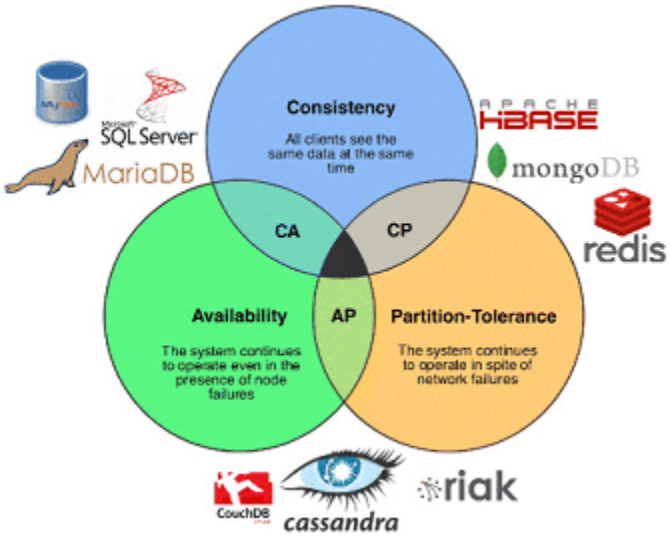
Классы NoSQL-СУБД с точки зрения CAP-теоремы и их значимость для Big Data

По аналогии с железным треугольником проектного менеджмента, когда требуется найти баланс между сроками, затратами и качеством выполненных работ **[3]**, тройственная ограниченность характерна и для распределенных систем. В этом смысле фактически, любое Big Data решение, не только NoSQL-СУБД, можно рассматривать с точки зрения ограничений CAP-теоремы, классифицируя ее по сочетанию 2-х свойств из 3-х возможных [4]**:**

- **CA (Availability + Consistency — Parition tolerance)**, когда данные во всех узлах кластера согласованы и доступны, но не устойчивы к разделению. Это означает, что реплики одной и той же информации, распределенные по разным серверам

друг другу, не противоречат друг другу и любой запрос к распределённой системе завершается корректным откликом. Такие системы возможны при поддержке ACID-требований к транзакциям (Атомарность, Согласованность, Изоляция, Долговечность) и абсолютной надежности сети. На практике таких решений на основе кластерных систем управления базами данных почти не существует. Классическим примером CA-системы называют распределённую службу каталогов LDAP, а также реляционные базы данных (PostgreSQL, MySQL, MariaDB, MS SQL Server).

- **CP-система (Consistency + Partition tolerance — Availability)** в каждый момент обеспечивает целостность данных и способна работать в условиях распада в ущерб доступности, не выдавая отклик на запрос. Устойчивость к разделению требует дублирования изменений во всех узлах системы, что реализуется с помощью распределённых пессимистических блокировок для сохранения целостности. По сути, CP – это система с несколькими синхронно обновляемыми мастер-базами. Она всегда корректна, отрабатывая транзакцию, только в том случае, если изменения удалось распространить по всем серверам. Она продолжает корректно читать данные даже при отказе одного из узлов кластера. Но в этом случае запись будет обрываться или сильно задерживаться, пока система не убедится в своей целостности и согласованности (консистентности). Из NoSQL-СУБД к CP-системам принято относить Apache HBase, MongoDB, Redis, MemcasheDB, Berkley DB, HyperTable и Google Big Table.
- **AP-система (Availability + Partition tolerance — Consistency)** не гарантирует целостность данных, обеспечивая их доступность и устойчивость к разделению, например, как в распределённых веб-кэшах и DNS. Считается, что большинство NoSQL-СУБД относятся к этому классу систем, обеспечивая лишь некоторой уровень согласованности данных в конечном счете (eventually consistent). Таким образом, AP-система может быть представлена кластером из нескольких узлов, каждый из которых может принимать данные, но не обязуется в тот же момент распространять их на другие сервера. Такая система отлично справляется с отказами нескольких узлов, но, когда они снова начинают работать, возможна выдача пользователям старых данных. К AP-системам относят CoucheDB, Cassandra, Riak, Amazon DynamoDB.



Тройственная ограниченность баз данных и распределенных Big Data систем с точки зрения CAP-теоремы по аналогии с железным треугольником проектного менеджмента

При всей понятной на первый взгляд концепции, CAP-теорему критикуют за **чрезмерное упрощение** важных понятий, что приводит к неверному пониманию первоначального смысла модели. В результате этого теорема из строгого, математически доказанного утверждения превращается в маркетинговый термин с расплывчатым смыслом [5].

Список использованных источников:

1. Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners (Wiley and SAS Business Series) 1st Edition.
2. CAP. URL: bigdataschool.ru/wiki/cap (Дата обращения: 21.09.2020)