

MAP/REDUCE: ОТ ГРЕЗ К РЕАЛЬНОСТИ

С.Д. КУЗНЕЦОВ

ИНСТИТУТ СИСТЕМНОГО ПРОГРАММИРОВАНИЯ РАН

Первая встреча (1)

- Сразу после появления в конце 2004 г. статьи Jeffrey Dean и Sanjay Ghemawat
MAPREDUCE: SIMPLIFIED DATA PROCESSING ON LARGE CLUSTERS
- Как здорово, что теперь мы можем не зависеть от СУБД
- Соединения можно выполнять как угодно
- Объемы данных не ограничены
- Распределенная и дешевая среда
- Горизонтальное масштабирование

Первая встреча (2)

- Полный восторг
- Что есть грезы, а что – реальность?
- Да, распределенная и дешевая среда
- Да, горизонтальная масштабируемость
- Да, возможность обработки больших данных
- Нет возможности отказаться от СУБД
- Нет счастья с соединениями

Первая встреча (3)

- Ручное программирование
- Ручная оптимизация
- Похожие подходы существовали до появления SQL
 - Умышленно не говорю о реляционном подходе
- Навигационные СУБД
- Аналог программирования на языке ассемблера
- Мастер может добиться отличных результатов

Первая встреча (4)

- СУБД предназначены для массового использования
- Мастерами данных, а не программирования
- Основная мудрость в оптимизаторе запросов
 - Как в компиляторах наиболее важны оптимизаторы
- Он знает, как лучше выполнять соединения
- Программисты приложений занимаются логикой приложений
- Данные обеспечиваются СУБД

Технология на пальцах (1)

- На всякий случай, чтобы все понималось однозначно
- Без технических деталей, хотя, возможно, в них кроется дьявол
- Программа `map/reduce` состоит из двух функций
 - `Map`
 - `Reduce`
- Интерфейсы этих функций (интерфейсы со средой) фиксированы
- Тела функций пишутся программистами

Технология на пальцах (2)

- Выполнение функции `map` на узле среды `map/reduce` приводит к разделению данных, доступных на этом узле, между узлами среды
- Можно сказать, что функция `map` производит *partitioning* данных
 - Возможно, после некоторой начальной обработки
- После завершения выполнения `map` на всех узлах, получивших раздел данных, выполняется функция `reduce`
 - Функция называется `reduce`, потому что, вообще говоря, она сокращает объем данных
- За `reduce` может следовать новое выполнение `map` и т.д.

Первая реакция сообщества баз данных (1)

- Строгая критики, почти что гнев и возмущение
- Наиболее яркий пример – заметка Девитта и Стоунбрейкера
MAPREDUCE: A MAJOR STEP BACKWARDS (январь 2008 г.)
- Интересно, что позже авторы перестали ссылаться на эту заметку
- Она доступна только на других сайтах, позже будет понятно, почему
- Звучит, как борьба с высокочкой, претендующим стать конкурентом
- Вот основные аспекты критики

Первая реакция сообщества баз данных (2)

- Огромный шаг назад по отношению к парадигме программирования крупномасштабных приложений больших данных
- Плохая реализация с опорой на методы грубой силы вместо использования индексации
- Отсутствие новых идей – реализация известного подхода, разработанного около 25 лет тому назад
- Отсутствие большинства возможностей, обычно поддерживаемых в современных СУБД
- Несовместимость с большинством инструментов СУБД

Почему map/reduce подходит для аналитиков (1)

- Субъективные наблюдения
- Сослаться не на кого
- К чему аналитики привыкли в мире SQL?
- Им нужны многомерные кубы
- Реально база данных – набор таблиц
- Построение куба – операторы `group by rollup` и `group by cube` (Джим Грей и др., 1996 *Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals*)

Почему map/reduce подходит для аналитиков (2)

- Итак, в основе лежит группировка
- Очень похоже на разделение
- Сгруппированные данные можно обрабатывать параллельно на разных узлах в режиме *shared nothing*
- Группировать не обязательно по значениям одного или нескольких столбцов
- Инвариантом группы может служить любая функция от значений столбцов, в том числе хэш-функция

Почему map/reduce подходит для аналитиков (3)

- Мне кажется очевидным, что в силу этого наблюдения
 - Функцию map можно считать обобщение оператора group by
 - Функцию reduce можно считать обобщение обработчика группы (вычисления агрегатных функций)
- Можно было бы сказать, что map/reduce применяется для анализа не табличных данных
 - Как это делала google при анализе журналов
- Но на самом деле, данные подгоняются под табличные, например, в форме ключ-значение

Технология и идеология (1)

- Реально технология map/reduce достаточно сложна
- Опирается на стек вспомогательных технологий, включая распределенные файловые системы и т.д.
- Основное преимущество, как мне кажется, в простоте идеологии
- Реально для разработчика приложений map/reduce не важна реализация файловой системы и даже ее наличие
- Важна идейная простота массивно-параллельного программирования с параллелизмом по данным

Технология и идеология (2)

- Когда-то и СУБД опирались на использование файловых систем
- Это позволяло упростить их за счет наличия инфраструктуры более высокого уровня
- Однако уже в 1990-е гг. в развитых СУБД стали использовать *row disks*
- Повышение эффективности (и предсказуемости)
- Меньше виртуализации – больше доверия
- И еще раз, при наличии гарантий СУБД (если им можно верить) подобные мелочи не должны беспокоить пользователей

Технология и идеология (2)

- Когда-то и СУБД опирались на использование файловых систем
- Это позволяло упростить их за счет наличия инфраструктуры более высокого уровня
- Однако уже в 1990-е гг. в развитых СУБД стали использовать *row disks*
- Повышение эффективности (и предсказуемости)
- Меньше виртуализации – больше доверия
- И еще раз, при наличии гарантий СУБД (если им можно верить) подобные мелочи не должны беспокоить пользователей

Технология и идеология (3)

- Конечно, наряду с привлекательной идеологией, популярность map/reduce обеспечила Apache Software Foundation
- Реально в мире известен hadoop, а не map/reduce
- Честь и хвала open source
- Но хорошо, что map/reduce придумали люди из google
- Иначе в hadoop все менялось бы настолько же часто, как в распределенных файловых системах

Маркетинг и реальность (1)

- Трудно жить в мире маркетинга
- Трудно отличить маркетинг от реальности
- Много слов, они затуманивают смысл
- На мой взгляд, наиболее интересная часть реальности (в контексте данного семинара) состоит в образовании связки
 - Массивно-параллельных СУБД и
 - Map/reduce (конечно, имеется в виду hadoop)
- Чего не хватает каждой из технологий?

Маркетинг и реальность (2)

- Map/reduce без СУБД имеет узкую область применения
- Может ли обойтись без map/reduce технология массивно-параллельных аналитических баз данных?
- Сравнительно хорошо обрабатываются запросы на SQL, включая вызовы агрегатных функций
- Разделение данных можно хорошо подогнать для эффективной поддержки заданной рабочей нагрузки
- Изменение рабочей нагрузки влечет потребность в тяжелой операции переразделения данных

Маркетинг и реальность (3)

- Совсем плохо дело с серверной аналитикой, обеспечиваемой пользователями
- Это нужно, поскольку при увеличении объемов данных вычисления должны перемещаться ближе к данным
 - Толстые клиенты и даже сервера приложений в мире массивно-параллельной аналитики становятся вредны
- Формально SQL необходимые возможности обеспечивает
- Но получаются последовательные программы
- Явное параллельное программирование серверных приложений с использованием MPI непригодно

Маркетинг и реальность (4)

- На сегодняшний день только `map/reduce` обеспечивает парадигму серверного аналитического программирования
- Она проста и по духу близка аналитикам
- Именно поэтому замолчал Стоунбрейкер, когда в `Vertica` вслед за `Greenplum` и `Asterdata` были реализованы соответствующие возможности
- И в этом отношении технология `map/reduce` не идеальна
- Может быть, следующий шаг – это что-то типа `spark`

Маркетинг и реальность (5)

- Еще одна реальность – это массивно-параллельные СУБД на основе map/reduce
- Map/reduce служит инфраструктурой для построения такой СУБД
- В узлах вместо reduce работают экземпляры обычных СУБД (почти без переделки)
- Пример – hadoopdb (теперь HADAPT)
- Казалось, что это быстрый прототип
- На самом деле, получилась работающая коммерческая система

Проблема больших данных

- Да, в каком-то смысле она решается
- Обеспечивается горизонтально масштабируемое решение для аналитики
- Проблема больших данных в целом не решается
- Да и можно ли ее решить в целом?

СПАСИБО ЗА ВНИМАНИЕ!