

Big Data и горизонтально масштабируемая аналитика

Сергей Кузнецов
ИСП РАН
kuzloc@ispras.ru

Введение (2)

- Проблема больших данных вечна и призрачна
- Всегда имелись важные данные, которые хотелось бы уметь эффективно хранить и обрабатывать, но объемы которых делали эту задачу непосильной для существующих систем управления данными
- За то время, пока исследователи и разработчики умудряются справиться с вчерашними большими данными, появляются новые большие данные, с которыми совладать по-прежнему невозможно
- Вечность и призрачность проблемы связана не только с постоянным ростом объема данных, но и с тем, что возникают потребности в хранении и обработке новых видов данных, для которых существующие системы плохо приспособлены (или не годятся вовсе)

Введение (3)

- ❑ Вечность и призрачность проблемы вряд ли позволяет рассчитывать на ее полное и окончательное решение
- ❑ Это гарантирует постоянную занятость в будущем исследователей и разработчиков систем управления данными
- ❑ Их деятельность напоминает попытки моряков доплыть до заманчивого миража, навеянного Фата-Морганой, но в данном случае сами эти попытки весьма увлекательны и полезны, поскольку худо-бедно поддерживают общее развитие человечества
- ❑ Иногда удается найти решение частных, но чрезвычайно важных случаев проблемы больших данных
- ❑ Речь идет о тех категориях данных, для управления которыми традиционно предназначены СУБД

Как с большими данными справляются перспективные СУБД (1)

- ❑ Технология SQL-ориентированных («реляционных») СУБД) позволяет эффективно управлять *транзакционными и аналитическими* базами данных
- ❑ Транзакционные базы данных предназначены для поддержки оперативных транзакционных приложений
- ❑ В транзакционных базах данных, главным образом, содержатся оперативные данные, отражающие текущее состояние бизнеса или другой области деятельности, быстро и часто обновляемые и поддерживающие разные аспекты этой деятельности
- ❑ Аналитические базы данных содержат исторические данные, относящиеся к деятельности конкретного предприятия, некоторой области бизнеса или некоторому научному направлению
- ❑ Эти данные поступают в базу данных из разных источников, одним из которых являются соответствующие транзакционные базы данных

Как с большими данными справляются перспективные СУБД (2)

- Проболе больших данных подвержены обе эти категории баз данных
- Объемы транзакционных баз данных возрастают из-за развития оперативных потребностей пользователей, бизнеса или науки
 - например, транзакционные базы данных Internet-магазинов значительно увеличиваются в объеме при внедрении технологии персонализации услуг
- Объемы аналитических баз данных увеличиваются, прежде всего, из-за своей природы: данные в них всегда только накапливаются и никогда не уничтожаются
- Другой серьезной причиной роста объема аналитических баз данных является потребность бизнес-аналитиков в привлечении новых источников данных (например, данных, черпаемых из открытых Internet-источников)

Как с большими данными справляются перспективные СУБД (3)

- Для транзакционных баз данных частный случай проблемы больших данных можно сформулировать следующим образом:
 - нужно обеспечить технологию относительно недорогого масштабирования СУБД и транзакционных приложений, позволяющую поддерживать требуемую скорость обработки транзакций
 - при росте объема данных и
 - увеличении числа одновременно выполняемых транзакций
- Для аналитических баз данных частный случай проблемы звучит примерно так:
 - требуется обеспечить технологию относительно недорогого масштабирования СУБД и аналитических приложений, позволяющую аналитикам
 - расширять возможности СУБД по части выполнения аналитических запросов и
 - обеспечивать эффективную оперативную аналитическую обработку данных при росте их объема

Как с большими данными справляются перспективные СУБД (4)

- В первом десятилетии нового века исследователям во главе с одним из пионеров технологии баз данных Майклом Стоунбрейкером (Michael Stonebraker) удалось нащупать пути решений обоих частных случаев проблемы
- В основе обоих решений лежат следующие общие принципы:
 - перенос вычислений как можно ближе к данным;
 - использование архитектуры без совместно используемых ресурсов (sharing nothing);
 - эффективное разделение данных по узлам вычислительной системы с возможностью их репликации в нескольких узлах

Как с большими данными справляются перспективные СУБД (5)

- ❑ **Перенос вычислений как можно ближе к данным**
- ❑ Первый принцип означает, что сама СУБД и приложения баз данных организуются таким образом,
 - чтобы минимизировались пересылки данных по сети, связывающей узлы соответствующей вычислительной системы
- ❑ Важность этого принципа возрастает при росте объема данных
- ❑ Следствием первого принципа является потребность в переносе приложений баз данных (частями или полностью) на сторону сервера

Как с большими данными справляются перспективные СУБД (6)

- **Использование архитектуры без совместно используемых ресурсов (sharing nothing)**

- Второй принцип обеспечивает возможность реального распараллеливания работы СУБД и приложений, поскольку

- при отсутствии общих ресурсов между узлами вычислительной системы

- фактически, при использовании кластерной архитектуры

уменьшается вероятность конфликтов между частями системы и приложений, выполняемыми в разных узлах сети

Как с большими данными справляются перспективные СУБД (7)

- **Эффективное разделение данных по узлам вычислительной системы с возможностью их репликации в нескольких узлах**
- Третий принцип обеспечивает
 - эффективную параллельную обработку транзакций или
 - эффективную поддержку оперативной аналитической обработки данных

Как с большими данными справляются перспективные СУБД (8)

- Все три принципа далеко не новы
 - их появление можно датировать 80-ми годами прошлого века
- Например, на принципах sharing nothing основывалась популярная и в высшей степени эффективная параллельная СУБД Teradata, которая успешно используется на протяжении нескольких десятков лет
- Однако именно в наше время отмеченные три принципа удалось успешно применить при создании реально масштабируемых параллельных транзакционных и аналитических СУБД

Как с большими данными справляются перспективные СУБД (9)

- Применение этих принципов является необходимым, но не достаточным для реализации обеих категорий систем
- В каждом случае приходится применять некоторые дополнительные идеи
- Транзакционные параллельные СУБД оказывается выгодно основывать на давно известных идеях
 - управления базами данных в основной памяти,
 - а для обеспечения надежности данных применять
 - развитую репликацию
- В аналитических же системах более выгодно применять
 - технологию хранения табличных данных во внешней памяти по столбцамв совокупности
 - с поддержкой разнообразных избыточных структур данных.
- Идеи хранения данных по столбцам тоже совсем не новы и ранее применялись, например, в аналитической СУБД Sybase IQ

Как с большими данными справляются перспективные СУБД (10)

- Можно считать, что пути решения проблемы больших транзакционных и аналитических данных намечены
- Это не означает, что даже эти частные виды проблем решены
- Например, транзакционные параллельные СУБД эффективно работают только при таком разделении данных, которое
 - минимизирует число распределенных транзакций в имеющейся рабочей нагрузке
 - при изменении рабочей нагрузки требуется перераспределять данные
- Аналитические параллельные СУБД справляются со сложными аналитическими запросами только в тех случаях, когда
 - разделение данных соответствует специфике этих запросов.
- Оставшаяся часть проблемы больших данных в этих случаях состоит в том, что
 - время от времени придется перераспределять данные громадного объема
 - в том числе, и при горизонтальном масштабировании систем

Как с большими данными справляются перспективные СУБД (11)

- Понятие больших данных является относительным
 - в частности, большие транзакционные данные по объему на несколько порядков уступают большим аналитическим данным
- Из уважения к житейскому понятию слова «большой» сосредоточимся на больших аналитических данных
 - хотя с технической и практической точек зрения транзакционные большие данные заслуживают не меньшего внимания

Масштабируемая параллельная серверная бизнес-аналитика (1)

- Сообщество баз данных сравнительно хорошо научилось строить горизонтально масштабируемые параллельные аналитические СУБД, способные поддерживать эффективное выполнение стандартных аналитических запросов
 - пренебрежем отмеченной ранее проблемой потребности в перераспределении данных
- Но вернемся к первому основному принципу – приближению вычислений к данным
- Если на сервере СУБД обеспечиваются лишь базовые средства аналитики, в серьезном аналитическом приложении придется перетягивать крупные объемы аналитических данных
 - на рабочие станции или в лучшем случае на промежуточные аналитические сервера

Масштабируемая параллельная серверная бизнес-аналитика (2)

- Единственным способом устранения этого дефекта является обеспечение возможности расширения серверных аналитических средств
 - новыми аналитическими функциями, поставляемыми бизнес-аналитиками
- На первый взгляд, соответствующие возможности обеспечиваются средствами языка SQL, которые позволяют пользователям
 - определять собственные функции, процедуры и даже типы данных
- Но SQL никоим образом не обеспечивает *распараллеливания* этих программ
- Другими словами, аналитик вынужден писать параллельные программы, которые должны будут выполняться на стороне сервера при выполнении будущих аналитических запросов
- Кусочки этих программ должны будут выполняться поблизости от соответствующих кусочков данных

Масштабируемая параллельная серверная бизнес-аналитика (3)

- Как это сделать?
- Единственным распространенным методом параллельного программирования в кластерной среде является использование
 - интерфейса MPI — интерфейса передачи сообщений
- В этом случае программист сам решает,
 - как расположить отдельные параллельно выполняемые части своей программы в узлах кластеров и
 - как обеспечить их взаимодействие для формирования окончательного результата
- Программирование с использованием интерфейса MPI представляет собой большую сложность даже для профессионально подготовленных программистов

Масштабируемая параллельная серверная бизнес-аналитика (4)

- Можем ли мы сделать профессиональными программистами бизнес-аналитиков, которым гораздо ближе математическая статистика, а не методы параллельного программирования?
- Скорее всего, типичный аналитик, которому будет предложено приступить к решению такой задачи, предпочтет пользоваться старыми добрыми пакетами аналитических программ
 - на своей рабочей станции, что в корне загубит всю идею горизонтальной масштабируемости систем
- На первый взгляд, эта проблема кажется неразрешимой, равно как неразрешимой кажется и более общая
 - проблема обеспечения удобных и эффективных средств параллельного программирования для программистов широкого профиля

Масштабируемая параллельная серверная бизнес-аналитика (5)

- Однако не так давно удалось найти подход и к решению этой проблемы, по крайней мере, первой ее части
 - здесь следует отметить заслуги разработчиков параллельных СУБД Greenplum и Asterdata
- Это решение основывается на применении технологии *MapReduce*

Масштабируемая параллельная серверная бизнес-аналитика (6)

- Технология MapReduce появилась в недрах компании Google в качестве замены аналитическим параллельным СУБД для решения собственных аналитических задач
- Технология быстро обрела популярность среди многих практиков, особенно молодежи, и поначалу вызвала
 - глубокое возмущение в сообществе баз данных
- Авторитетные специалисты из этой области утверждали, что MapReduce — это
 - возврат к доисторическому времени, когда для решения проблем управления данными требовалось *явное программирование*,
- и упрекали сторонников MapReduce
 - в невежестве и неразумном отрицании серьезных результатов прошлых лет

Масштабируемая параллельная серверная бизнес-аналитика (7)

- Скорее всего, эти доводы были и остаются правильными
- Технология MapReduce не может и не должна заменить технологию баз данных
- Но оказалось, что эта технология может быть чрезвычайно полезной,
 - если ее применить внутри самой параллельной аналитической СУБД для
 - поддержки параллельного программирования и
 - выполнения аналитических функций, предоставляемых пользователями

Масштабируемая параллельная серверная бизнес-аналитика (8)

- MapReduce концептуально несравненно проще, чем MPI
- Программисту нужно понимать только одну идею:
 - данные надо сначала распределить по узлам кластера, а потом обработать
 - результат обработки можно снова распределить по узлам кластера и снова обработать и т.д.
- От программистов приложений требуется всего лишь обеспечить программный код двух функций —
 - функции map, обеспечивающей разделение данных по узлам кластера, и
 - функции reduce, обеспечивающей обработку данных в полученных разделах
- Конечно, такая парадигма программирования гораздо проще для непрофессиональных программистов, чем MPI, но, что еще более важно, она
 - должна быть понятийно близка аналитикам

Масштабируемая параллельная серверная бизнес-аналитика (9)

- Традиционная аналитика или то, что мы привыкли называть аналитикой, имеет дело с данными, явно или неявно представляемыми в виде
 - многомерного куба
- Первый и, по видимому, революционный шаг на пути к обеспечению аналитической обработки табличных SQL-ориентированных данных, в свое время обеспечивала работа Джима Грея, в которой он предложил
 - оператор roll up, позволяющий динамически построить многомерный куб в реляционной базе данных

Масштабируемая параллельная серверная бизнес-аналитика (10)

- Именно Джим Грей понял, что
 - традиционный оператор group by позволяет получить
 - грань многомерного куба,
 - а общая процедура построения многомерного куба сводится
 - к повторному выполнению оператора group by до тех пор, пока не будет получено нужное число измерений
 - для выполнения операции roll up требуется
 - столько же усилий, сколько и для выполнения простого group by
- Что же такое тогда аналитика в контексте SQL-ориентированной СУБД?
- Грубо говоря, она сводится к применению различных агрегатных аналитических функций к граням куба

Масштабируемая параллельная серверная бизнес-аналитика (11)

- Операцию `group by` можно было бы с успехом считать операцией «партиционирования» таблицы в соответствии со значениями заданного столбца группировки
 - после этого можно было бы легко обрабатывать полученные группы параллельно
- Можно считать, что функция `map` в программе MapReduce — это
 - обобщенный оператор `group by`, обеспечивающий разделение исходных данных по некоторым явно запрограммированным пользователем правилам
- Функцию `reduce` можно считать
 - обобщением агрегатной функции в SQL
- Аналитик при написании новой аналитической функции не обязан выходить за пределы того набора понятий, к которому он привык, работая с традиционной аналитической СУБД

Масштабируемая параллельная серверная бизнес-аналитика (12)

- Наличие поддержки технологии MapReduce внутри параллельной аналитической СУБД должно
 - полностью удовлетворить потребности аналитиков, и будущие аналитические приложения станут
 - серверными приложениями, выполняемыми параллельно поблизости от тех данных, которым они адресованы
- Все это означает, что горизонтальная масштабируемость будущих аналитических систем может быть обеспечена
 - тем самым для них может быть решена и проблема больших данных
 - если, конечно, по-прежнему, не принимать во внимание потребность в регулярном перераспределении огромных объемов аналитических данных

Масштабируемая параллельная серверная бизнес-аналитика (13)

- Фактически, в новом поколении аналитических параллельных СУБД обеспечивается возможность параллельного программирования аналитических приложений, которое
 - проще и понятней для неподготовленных специалистов, чем традиционный интерфейс MPI.
- Т.е. фактически частично решается более общая проблема — проблема параллельного программирования для суперкомпьютеров
- Встает вопрос, а
 - не заслуживает ли этот подход более широкого использования, чем расширение аналитического параллельного сервера баз данных?

Масштабируемая параллельная серверная бизнес-аналитика (14)

- Не стоит ли попытаться применять технологию MapReduce для программирования параллельных задач, требующих обработки больших объемов данных?
- Не оставляет мысль, что как
 - большие данные почти всегда бессмысленны без больших вычислений,
 - так и большие вычисления
 - чаще всего невозможны без поддержки эффективного доступа к большим данным.
- Не стоит ли об этом задуматься?

Спасибо за внимание!

Бонус (1)

- ❑ Можно ли считать аналитикой действия над данными, не представляемыми в виде многомерного куба?
- ❑ Что такое анализ графовых данных и можно ли его распараллелить?

Бонус (2)

- Не конфликтуют ли виртуализация «облачной» среды с потребностями СУБД в знаниях о реальной аппаратуре?
- Не стоит ли расширить требования к Service Level Agreement?