# Большие Данные: разделяй и властвуй

Сергей Кузнецов, ИСП РАН

# Три принципа массивно-параллельных СУБД

- Архитектура Shared-Nothing
- Приближение вычислений к данным
- Эффективное разделение данных

# Архитектура Shared-Nothing

- Отсутствие ресурсов, общих для нескольких узлов системы
- Общение только на основе передачи сообщений
- Возможность использования компьютеров массового спроса
- Горизонтальная масштабируемость

# Приближение вычислений к данным

- Чем сложнее обработка данных в приложении, тем большие объемы данных приходится передавать по сети
- При использовании параллельных СУБД выгоднее и вычисления выполнять параллельно
- Выгоднее всего обрабатывать данные в тех узлах, в которых они хранятся
- Подробнее в выступлении на **МСКФ-2014**

# Эффективное разделение данных

- Массивно-параллельные СУБД могут эффективно работать только при наличии эффективного разделения данных
- Для аналитических баз данных разделение должно минимизировать число обменов сообщениями между узлами при выполнении запросов для заданной рабочей нагрузки
- Для транзакционных баз данных разделение должно минимизировать число распределенных транзакций для заданной рабочей нагрузки
- Разделение критически важно для обеспечения эффективности системы
- Без возможности эффективного разделения нельзя говорить о горизонтальной масштабируемости
- Дальше говорим только о разделении

# Важность разделения данных для массивно-параллельных технологий обработки данных

- Разделение данных важно не только для массивно-параллельных СУБД
- Правильное разделение данных необходимо для эффективной работы систем, основанных на парадигме map/reduce
  - map должна сработать так, чтобы последующая reduce была осмысленна и эффективна
- Без эффективного разделения данных невозможна параллельная обработка графовых структур данных
  - Подграфы, на которые разделяется граф, должны связываться минимальным числом дуг
- Другими словами, если мы умеем эффективно разделять данные, мы можем обеспечить их горизонтально-масштабируемую обработку, т.е. (по крайней мере, частично, решим проблему Больших Данных)

# В чем состоит общая задача

- Для обеспечения эффективного разделения данных во всех случаях приходится решать одну и ту же графовую задачу
- Заданный граф нужно разбить на подграфы с заданными характеристиками (например, с одним и тем же числом узлов) таким образом, чтобы число узлов, связывающих подграфы, было минимально (не вполне точная формулировка)
- Известно, что задача эта NP-полная; для больших графов точное решение получить практически невозможно
- Применяются эвристики

# Что стоит делать?

- Задачу разделения графа приходится решать в разных областях
  - В области массивно-параллельных баз данных
  - В области САПР (видимо, потребность возникла раньше всего)
  - В области анализа социальных сетей
  - И т.д.
- В каждой области применяются свои эвристики
- Нужно выполнить развернутое исследование с анализом и сравнением всех известных эвристических алгоритмов
- Нужно изучить, в каких условиях одни эвристики работают лучше других
- Возможно, нужно придумать гибридные алгоритмы разделения, которые будут автоматически переходить с одной эвристики на другую

# Наступит ли счастье?

- Вряд ли
- Во-первых, при оптимизации разделения данных с учетом рабочей нагрузки каждая отдельная операция этой рабочей нагрузки может выполняться неоптимально (это неустранимый изъян)
- Во-вторых, при смене рабочей нагрузки приходится (вообще говоря) переразделять данные
  - Это массовая и дорогостоящая работа
- В-третьих, переразделения данных невозможно избежать при горизонтальном масштабировании системы
  - Могут возникать длительные простои системы

# Как приблизиться к счастью?

- Требуется научиться сохранять работоспособность системы при перераздeлении данных из-за смены рабочей нагрузки
  - Возможна деградация производительности
  - Но производительность должна повышаться по мере развития процесса перераздeления
- Требуется научиться сохранять работоспособность системы при перераздeлении данных из-за добавления узлов к системе
  - В идеале, система должна показывать производительность на той же раочей нагрузке, не меньшую той, которая была до начала процесса перераздeления узлов

# Возможно ли это?

- Наверное, сложнее всего найти действительно хороший и достаточно универсальный алгоритм разделения графа
- Проблемы переразделения не являются простыми, но здесь, на мой взгляд, могут сработать чисто «программистские» хитрости
- Так что, главное, научиться разделять данные, с остальным справиться можно

# Спасибо за внимание