MapReduce

Доклад на семинаре «высокопроизводительные вычисления»

Чуканов Вячеслав, 6057/2

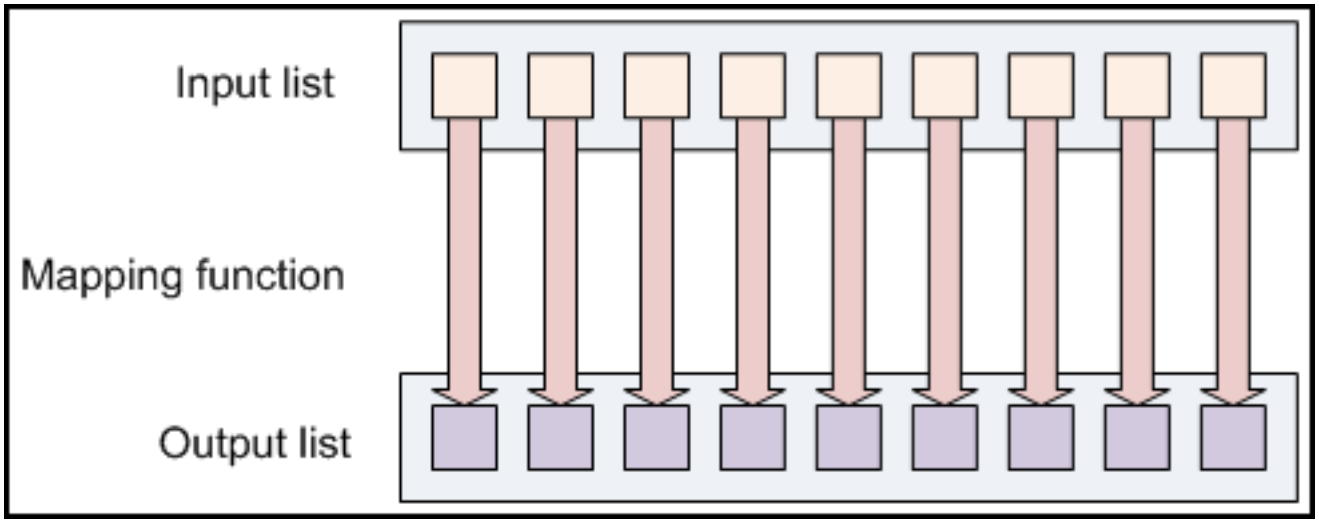
Содержание

Введение в MapReduce

- MapReduce – программная модель для решения задач обработки больших массивов данных на кластерах
- Входные/выходные данные – множество пар {ключ, значение}
- MapReduce состоит из двух основных стадий
  - Map – преобразование множества входных пар {ключ, значение} в некоторое другое множество пар {ключ, значение}.
  - Будем называть эти данные «intermediate-данными»
  - Reduce – принимает на вход пары {ключ, список значений} сформированные из пар {ключ, значение} с одинаковым ключом и выдает несколько пар {ключ, значение}
    - Число выходных пар должно быть меньше числа входных

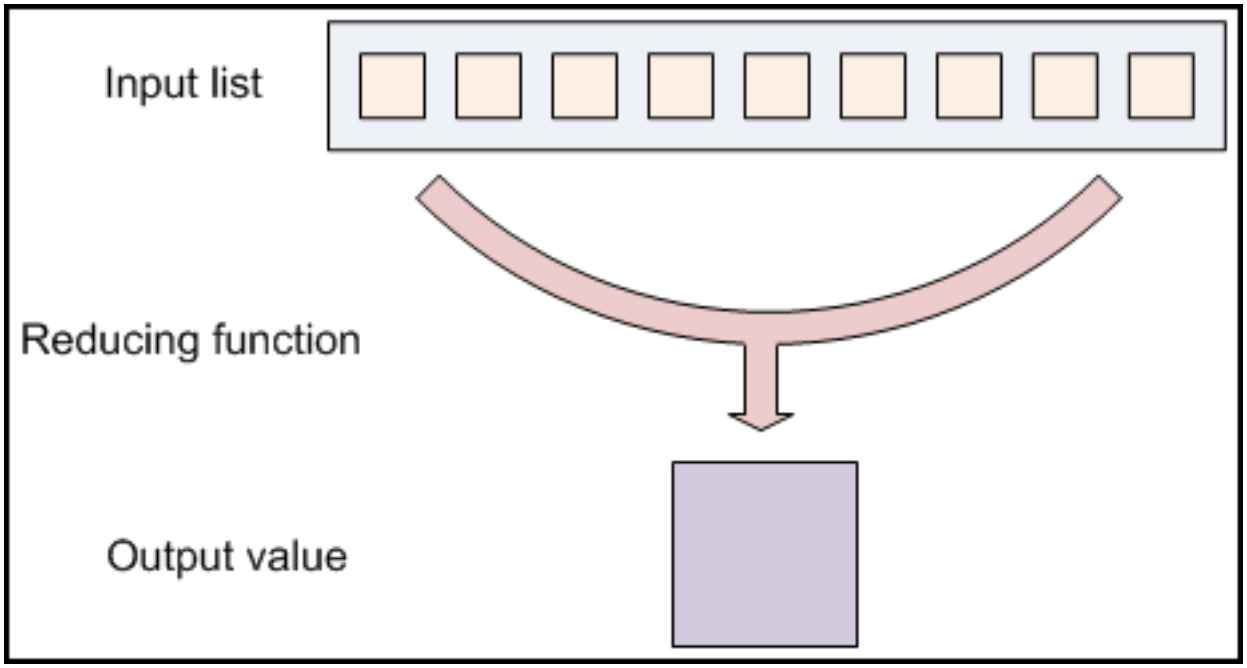
Распараллеливание Map

- Основная идея – обработка пар {ключ, значение} производится независимо



Распараллеливание Reduce

- Распараллеливание по ключам – пары {ключ, список значений} обрабатываются (по возможности) параллельно



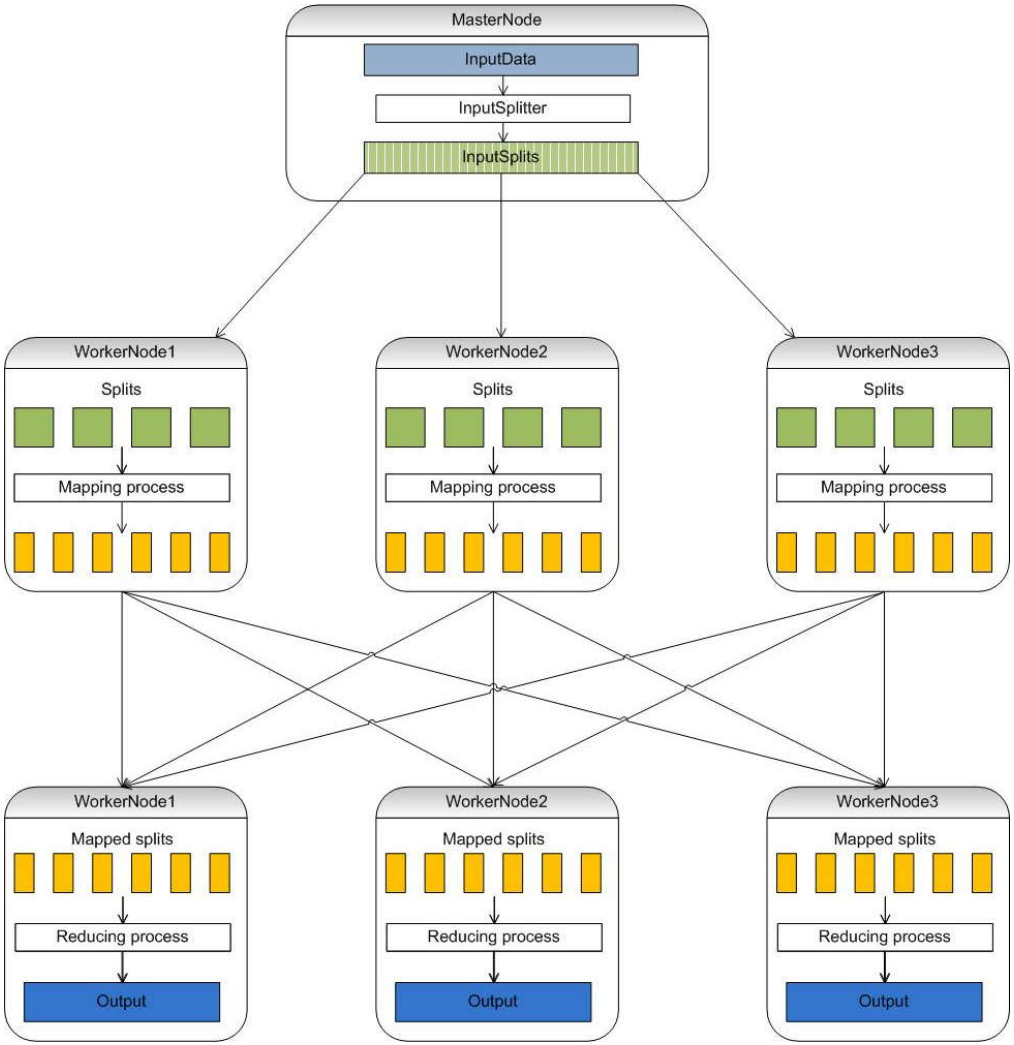
Пример MapReduce задачи

- Задача: посчитать количество различных слов
- Вход: множество пар {число, строка}
  - Число (ключ) – номер пары, от 1 до количества пар
  - Строка – слово, составленное из букв англ. Алфавита
- Map: преобразует пару {номер пары, слово} в пару {слово, 1}
- Reduce: считает сумму значений в списке для данного ключа
- Выход: пары {слово, количество}

Конвейер MapReduce

- Конвейер MapReduce состоит из множества стадий, определяющих пересылку и обработку данных
  - Составление входных данных – формирование пар {ключ, значение}
  - Распределение входных данных
  - Выполнение задач «map»
  - Сортировка полученных данных
  - Распределение intermediate-данных
  - Выполнение задач «reduce»
  - Сброс данных на один узел для выполнения задачи «reduce»
- Нет четкого регламента о том, как задается каждая стадия

Конвейер MapReduce (2)



Стадии MapReduce конвейера

- InputSplitter
- Map
- Partition
- Shuffle
- Sort
- Reduce
- Global Reduce

InputSplitter

- InputSplitter - стадия выделения пар {ключ, значение} из пользовательского потока данных
  - Пример: ключ – номер строки в файле, значение – содержимое строки
- InputSplitter нужен для выдачи массивов пар, сформированных из входного потока
- Реализация зависит от постановки задачи
  - InputSplitter может быть программно реализован пользователем (наследник класса)
  - InputSplitter может быть сконфигурирован автоматически на основе некоторых пользовательских данных

Стадии MapReduce конвейера

- InputSplitter
- Map
- Partition
- Shuffle
- Sort
- Reduce
- Global Reduce

Map

- Выполнение пользовательской функции map для каждой пары из входного массива данных
- Функция map задается на некотором языке
  - Hadoop: Java
  - GPU: OpenCL/CUDA
- Local reduce
  - Для уменьшения объема полученного массива данных запускается reduce
  - Накладывает ограничение на reduce – функция reduce должна быть замкнута на множестве входных ключей
  - Необходимо предварительно отсортировать данные для представления их в виде пар {ключ, список значений}

Стадии MapReduce конвейера

- InputSplitter
- Map
- Partition
- Shuffle
- Sort
- Reduce
- Global Reduce

Partition

- Распределение intermediate-данных по узлам
- Автоматизированное – в зависимости от производительности узлов сети (кластера)
- Пользовательское – выполнение некой функции, указывающей, какому узлу (по номеру) пары с какими ключами соответствуют
- Гибридное
  - Например, система нумерует узлы в порядке возрастания их вычислительной мощности и запускает пользовательскую функцию

Стадии MapReduce конвейера

- InputSplitter
- Map
- Partition
- Shuffle
- Sort
- Reduce
- Global Reduce

Shuffle

- Рассылка данных в соответствии с результатами стадии Partition
- Непосредственная рассылка данных по исполняющим узлам
  - Узлы сами координируют свои действия
- Перемещение данных на некоторое сетевое хранилище и выдача узлам метаданных для загрузки
  - Координация действий производится при помощи некоторого master-процесса

Стадии MapReduce конвейера

- InputSplitter
- Map
- Partition
- Shuffle
- Sort
- Reduce
- Global Reduce

Sort & Reduce

- Reduce стадия – выполнение пользовательской функции reduce
- За удовлетворение функцией определенным требованиям системы отвечает пользователь
  - Мощность выходного множества меньше множества входного
  - Замкнутость на множестве входных ключей
- Вход – пары {ключ, список значений}
  - Для составления таких пар требуется сортировка данных
- Сортировка
  - CPU – qsort
  - GPU – bitonic sort. GPU-сортировка часто применяется в случае, когда reduce выполняется на GPU

Стадии MapReduce конвейера

- InputSplitter
- Map
- Partition
- Shuffle
- Sort
- Reduce
- Global Reduce

Global Reduce

- Перемещение данных, полученных после стадии Reduce на некоторый узел для выполнения функции reduce
- Необходимо в случае, когда на разных узлах были обработаны пары с одинаковыми ключами
  - Пример: все intermediate-пары имеют один и тот же ключ
- Требует замкнутость функции reduce

Примеры задач, решаемых при помощи MapReduce

- String Match – поиск строки в тексте
  - Map: {номер, строка текста} → {номер, позиция подстроки}
  - Reduce – не требуется
- Word Count – подсчет числа слов
  - Map: {номер, слово} → {слово, 1}
  - Reduce: {слово, (1, 1, 1,…1)} → {слово, число вхождений в текст}

Примеры задач, решаемых при помощи MapReduce (2)

- PageViewCount – число просмотров веб-страницы разными пользователями
  - Вход – лог просмотров (URL, IP, Cookie)
- Решается в два запуска MapReduce конвейера
- Первый – удаление дубликатов
  - Map: {номер, строка лога} → {строка лога, размер строки лога}
  - Reduce выдает ровно одну пару из списка пар с одинаковыми ключами
- Второй – подсчет числа просмотров
  - Map: {строка лога, ее размер} → {URL, IP}
  - Reduce: {URL, (IP, IP, IP,…)} → {URL, count}

Примеры задач, решаемых при помощи MapReduce (3)

- PageViewRank – топ-N самых посещаемых
- Использует данные, полученные по завершению PageViewCount
- Один запуск MapReduce
  - Map: {URL, Count} → {Count, URL}
  - Sort
  - Reduce не требуется

Литература

- Bingsheng He, Wenbin Fang, Qiong Luo, Naga K. Govindaraju, Tuyong Wang «Mars: A MapReduce Framework on Graphics Processors» PACT 2008
- Yahoo Hadoop MapReduce Tutorial http://developer.yahoo.com/hadoop/tutorial/module4.html