

Курсовой проект
*Фреймворк и файловая система для
распределённой обработки больших данных в
рамках концепции map-reduce*

Выборнов А.И.

МГТУ им. Н. Э. Баумана

art-vybor@ya.com

10 декабря 2014 г.

Обзор

Постановка задачи

- ▶ Анализ требований и проектирование архитектуры системы. Реализация нераспределённого map-reduce. Решение проблемы RPC.
- ▶ Реализация распределённой файловой системы. Реализация фреймворка. Тестирование на примерах.

Зачем нужен map-reduce?

- ▶ Обработка больших данных (Big Data).
 - ▶ Вычисления превосходят возможности одной машины.
 - ▶ Данные не помещаются в памяти, необходимо обращаться к диску.
 - ▶ Можно хранить много данных, но задержки и пропускная способность оборудования растут пропорционально данным.
- ▶ Удобная абстракция для построения алгоритмов обработки больших данных.
- ▶ Устойчивость к отказам.

Что такое map-reduce?

- ▶ Структура $(key, value)$ - пара (ключ, значение).
- ▶ Программирование представляет собой определение двух функций:
 - ▶ $map : (key, value) \rightarrow [(key, value)]$
 - ▶ $reduce : (key, [value]) \rightarrow [(key, value)]$
- ▶ Между стадиями *map* и *reduce* происходит группировка и сортировка данных.

Как работает распределённая файловая система

- ▶ Файл разбивается на блоки фиксированного размера (64мб).
- ▶ Разбиение файла на блоки порождает генератор.
- ▶ Получившиеся блоки распределяются по машинам.

Как работает распределённый map-reduce

- ▶ На вход получает адреса файлов в DFS для ввода и вывода, а также файл с функциями map и reduce.
- ▶ Стадии выполнения:
 - ▶ Получение по входному файлу списка индексов в DFS.
 - ▶ Разбиение списка индексов по узлам.
 - ▶ Выполнение функций map на узлах.
 - ▶ Слияние результатов функций map.
 - ▶ Разбиение полученного результата по узлам.
 - ▶ Выполнение функций reduce на узлах.
 - ▶ Добавление результатов функций reduce в виде файла в DFS.
 - ▶ Переименование результатов.
 - ▶ Запись информации о новом файле в DFS.

Как работает map на узлах

- ▶ По списку индексов получаются данные путём обращения к узлу dfs, находящемуся на этой машине.
- ▶ Для каждого индекса выполняется переданная функция map.
- ▶ Результаты всех map агрегируются в ассоциативный массив (python dictionary).
- ▶ Полученный результат передаётся на главный узел.

Как работает reduce на узлах

- ▶ Для каждой полученной пары выполняется переданная функция reduce.
- ▶ Полученный результат разбивается на блоки равного размера (64мб).
- ▶ Блоки записываются в dfs на узел, где выполнялся reduce под псевдоименами.

Что сделано

- ▶ Разработаны архитектуры распределённых map-reduce и файловой системы.
- ▶ Реализован распределённый map-reduce.
- ▶ Для реализации RPC была выбрана связка: ZeroMQ + Marshal.
- ▶ Реализована распределённая файловая система для хранения больших данных.

Что предстоит сделать

- ▶ Развить map-reduce для работы с большими данными.
- ▶ Протестировать полученную систему на больших данных и проанализировать результаты.

Курсовой проект
*Фреймворк и файловая система для
распределённой обработки больших данных в
рамках концепции map-reduce*

Выборнов А.И.

МГТУ им. Н. Э. Баумана

art-vybor@ya.com

10 декабря 2014 г.