MapReduce + External Sort. Гистограмма.

Грибчук Даниил

24 мая 2020 г.

Map

На данном этапе, я разбивал входной файл на файлы с максимальным размером 1000 строк, файлы создавались во временной директории (создавалась с помощью *boost::filesystem::unique_path*), для каждого такого файла запускался *mapper* с помощью *boost::process::child*, одновременно таких процессов может быть не более 50.

Задача *mapper* состояла в следующем — это разбить строку по \t и выделить из нее key и записать в свой файл вывода key'\t'1'\n'

После удачного завершения работы всех *mapper* объединяем все файлы и удаляем временную директорию со всеми временными файлами, если какой-то *mapper* завершился некорректным образом будет выброшена ошибка.

Reduce

На данном этапе, опять создается временная директория в которой будут создавться все временные файлы. Процессы запускаются также с помощью *boost::process::child*, .

- ExternalSort — разбиваем наш входной файл на остортированные файлы, максимум по 1000 строк каждый, и мерджим их все в один отсортированный файл с помощью *priority_queue*. В данном случае сортить можем как обычные строки, ибо у нас нет отрицательных значений и все ключи одинаковые.
- Reduce — разбиваем наш отсортированный входной файл, на файлы следующим образом:
  1. Находим длинну шага — $\text{step} = \frac{1}{\text{count\_of\_range}}$
  2. Находим правую границу текущего диапазона — prev_right_border + step, перовоначально prev_right_border = 0
  3. Записываем во временный файл все строки у которых key $\leq$ right_border. Если встречаем key $>$ right_border, то если временный файл был открыт, отдаем его *reducers*, увеличиваем right_border до тех пор пока она на станет больше key, тогда откываем новый файл и записываем туда key.
  4. *reducers* принимает файл и находит сумму всех value и записывает в свой временный файл last_key'\t'value_sum'\n'. Одновременно таких процессов может быть не более 50.
  5. После удачного завершения работы всех *reducers* объединяем все файлы и удаляем временную директорию со всеми временными файлами, если какой-то *reducers* завершился некорректным образом будет выброшена ошибка.

Create Histogram

На данном этапе происходит построение гистограммы. Скрипт принимает вывод *map_ reduce* и по этим значениям строит гистограмму. Если кол-во столбцов больше 12, то шкала по оси x разбивается на 10 интервалов, иначе разбивается на заданное кол-во инетрвалов, также все значение по оси x округлены до 2 знаков (На данные это никак не влияет).

На вход подается файл, и кол-во столбцов, далее я нахожу опять правые границы и смотрю:

- Если текущий key $\leq$ right_border, то значение для интревала [prev_right_border; right_border] это value и переходим к следующему *key* и интервалу.
- Если key $>$ right_border, то ставлю этому инервалу 0, и беру следующий интервал и опять сравниваю с этим key.

Example of work