Домашнее задание №1

Срок сдачи: 29 октября, 23:59:59. Решения, присланные позже данного срока, не принимаются.

Формулировка задания:

На вход подается .csv файл (разделители – запятые) с финансовыми данными. Данный файл лежит в директории /data/fin на общем кластере, доступном по адресу http://users.bigdata.local:8888

Столбцы данного файла имеют следующие названия:

#SYMBOL,SYSTEM,MOMENT,ID_DEAL,PRICE_DEAL,VOLUME,OPEN_POS,DIRECTION

где #SYMBOL – название финансового инструмента;

MOMENT – время (дата);

PRICE_DEAL – цена.

Пример строки в файле:

SVH1,F,20110111100000080,255223067,30.46000,1,8714,S

Обратите внимание, что колонки во входных файлах могут располагаться в разном порядке. Это необходимо учитывать в решении и корректно обрабатывать.

Данный файл отсортирован по дате и времени.

Дата и время указываются в формате YYYYMMDDhhmmssfff, где f – миллисекунды.

Внимание!

Так как финансовые данные большого размера, отладку программы следует производить на подвыборке меньшего размера:
https://m.cs.msu.ru/s/mA4xyL6ytqAEE8f,
и только убедившись, что все работает, запускать программу на данных большого размера.

Вам необходимо:

Привести данные к формату японских свечей, используя Hadoop MapReduce Java API. **При этом, необходимо оптимизировать обработку данных с учетом ликвидности инструментов.**

Каждая свеча – это:

- MOMENT – время начала свечи;
- OPEN – цена первой сделки за свечу;
- HIGH – максимальная цена за свечу;
- LOW – минимальная цена за свечу;
- CLOSE – цена последней сделки за свечу.

Формат командной строки для запуска программы:
hadoop jar candle.jar candle -conf config.xml <Название входной директории> <Название выходной директории>

Параметры программы должны содержаться в конфигурационном файле config.xml.
В программе должны использоваться следующие параметры со значениями по умолчанию:

- candle.width = 300000 #"ширина" свечи в числе миллисекунд;
- candle.securities = ".*" #шаблон инструментов – задается в виде регулярного выражения;
- candle.date.from = 19000101 #первый день периода времени (ГГГГММДД);
- candle.date.to = 20200101 #первый день после последнего дня периода (ГГГГММДД);
- candle.time.from = 1000 #время (ЧЧММ) начала первой свечи;
- candle.time.to = 1800 #время (ЧЧММ) после окончания последней свечи;
- candle.num.reducers = 1 #число редьюсеров

Обратите внимание, что на кластере отсутствует поддержка задания параметров программы через конфигурационный файл (опция -conf). Ваша программа должна поддерживать задание параметров через конфигурационный файл (опция -conf) на Cloudera и через опцию -D (прямое задание значений параметров) на кластере. Например: -D candle.width=2 и т.д.

Свечи "начинаются" в моменты времени, кратные "ширине". Отсчет времени для вычисления кратности начинается в 00:00 рассматриваемого дня.

На выходе необходимо получить директорию с файлами. Имена файлов должны содержать SYMBOL в качестве префикса. Файлы для каждого SYMBOL должны быть отсортированы по MOMENT с учетом номера Reducer в имени файла.

Формат выходных данных (каждого файла) без шапки:

SYMBOL,MOMENT,OPEN,HIGH,LOW,CLOSE

Пример строки в выходном файле (обратите внимание, что точность определяется одним знаком после запятой):

GDH1,20110111100000000,1407.0,1407.0,1379.0,1379.3

Далее Вам необходимо с Вашей **gse-почты** отправить на почту **bigdatamsu@gmail.com** архив в формате Task1-Фамилия**.zip**. Архив должен содержать:

1. Директорию “prog”, в которой должны быть:
    a. Файл pom.xml, в котором описывается вся структура Вашего проекта – .jar мы будем собирать на своей стороне;
    b. Директорию с исходными файлами, необходимыми для сборки .jar **(!)** Очень важно, чтобы имя Вашей программы было **candle.jar**;
    c. Файл config.xml с параметрами задачи (см. “Отчет”) и оптимальными параметрами запуска;
    d. Другие вспомогательные файлы, которые потребуются для запуска программы.
2. Файл readme.txt с описанием того, как Вы компилировали и запускали программу.

Обратите внимание, что Ваш архив должен соответствовать указанной выше структуре (**т.е. readme.txt и prog должны находиться в корне архива**).
В теме письма необходимо указать номер задания (в том же формате, что и архив: Task1-Фамилия.zip).
(!) Самое главное: задания будут взяты в обработку только если Вы отправляете своё решение с Вашей gse-почты, в противном случае задания приниматься не будут!
Если с этим имеются сложности – отпишитесь, пожалуйста, на bigdata@cs.msu.ru.

Отчет:
В тексте письма с Домашним заданием №1 необходимо указать ссылку на документ Google docs с Вашим отчетом по выполнению домашнего задания. В отчете необходимо:

1. Подробно описать принципы работы Вашей программы;

2. Оптимизировать равномерность распределения работ по редьюсерам в зависимости от количества используемых редьюсеров на “большом” датасете со следующими входными параметрами:

    candle.width = **1000**
    В качестве значений оставшихся параметров программы используются значения по умолчанию.

    Подобранные оптимальные параметры кластера (количество редьюсеров, использование комбинаторов и т.д.) необходимо отразить в конфигурационном файле config.xml.

3. Сформулировать выводы по результатам проведенных экспериментов, построить вспомогательные графики.

Пояснение:

1) В ситуациях, когда рассматриваются записи, в которых совпадают названия инструментов и моменты времени, но цены различны, для разрешения неоднозначности необходимо дополнительно рассматривать поле ID_DEAL. При одинаковых моментах времени для цены открытия (OPEN) выбирается цена с наименьшим ID_DEAL, для цены закрытия (CLOSE) – с наибольшим ID_DEAL;
2) Свечи необходимо строить от candle.time.from до candle.time.to каждого рассматриваемого дня;
3) Рассматривать переход через сутки не нужно (программа будет тестироваться на свечах, построенных в рамках одного рабочего дня, 10:00 – 18:00, либо меньшего периода времени);
4) Рассматривать случай, когда последняя свеча не залезает целиком в рассматриваемый промежуток времени, не нужно (считаем, что в рассматриваемый период времени укладывается целое число свеч, и в момент времени candle.time.to должна начаться новая свеча, которую мы не рассматриваем);
5) Рассматривать случай, когда candle.time.from не кратно размеру свечи, не нужно. Считаем, что candle.time.from всегда кратно размеру свечи.