Практическое задание на Apache Spark

Срок сдачи: 30 декабря, 23:59:59. Решения, присланные позже данного срока, не принимаются.

Вариант на Ваш выбор!

Вариант 1. “Поиск наиболее скоррелированных пар инструментов + временных интервалов корреляции + интервалов сдвига между инструментами в паре.”
Задание является усложнением соответствующего задания на Hadoop.
Теперь требуется найти не просто корреляции между CLOSE (а точнее, между относительным приростом CLOSE как в аналогичном предыдущем задании) свечей заданной ширины для всевозможных пар инструментов, но и проверить всевозможные интервалы свечей и величины сдвига во времени между парами инструментов (пусть они кратны интервалу свечи).
Таким образом, число подсчетов корреляций увеличивается в O(M*K) раз, где M - число проверяемых значений ширины свечи, K - число проверяемых значений сдвига.
Параметр candle.width переименовываем в candle.widths - множество свечных интервалов (в сек.), которые необходимо проверить.
Добавляется параметр candle.shifts - множество интервалов сдвига во времени (выражается в числе свечей) между свечами инструментов, которые необходимо проверить.
В ОЗУ кластера кешируйте свечи (вернее то, что Вы из них используете).
Параметры можно передавать в конфиг задачи.
Примеры значений всех параметров:
candle.securities = ".*" #шаблон инструментов – задается в виде регулярного выражения;
candle.date.from = 20110111;
candle.date.to = 20110112;
candle.time.from = 1000;
candle.time.to = 1015.
candle.widths=5,10
candle.shifts=0,1,2
Формат выходных данных:
<инструмент1> <инструмент2> <ширина свечи> <сдвиг> <коэффициент корреляции>
Сортировка по убыванию модуля коэффициента корреляции.

Прислать результаты работы программы для запуска на
candle.securities = "SVH1|GDH1"
candle.widths = 5,10;
candle.date.from = 20110111;
candle.date.to = 20110112;
candle.time.from = 1000;
candle.time.to = 1015

candle.shifts=0,1,2
Данные: wasb://financedata@bigdatamsu2.blob.core.windows.net/ (пример данных небольшого объема у Вас есть)

Дополнительное задание (можно делать тем, кому интересно или кто хочет улучшить ситуацию, если не успел сделать вовремя другие ДЗ. Дополнительное задание, решенное с ошибкой, не оказывает негативного влияния на оценку основного задания)
Предложить и проверить доходность простейшей торговой стратегии, работающей для пары отрицательно скорреллированных инструментов. Т.е. предложить алгоритм совершения покупок и продаж соответствующих инструментов и подсчета итоговой прибыли.
Вы попадаете на биржу с нулевыми позициями (число купленных Вами инструментов) двух инструментов и некоторой суммой денег. Далее считаем, что если в исторических данных встретилась сделка с некоторой ценой, считаем, что Вы могли в этот момент купить/продать по этой цене (на какой объем - число купленных/проданных активов - была сделка - не учитываем). Обратите внимание, что с 0 позицией по инструменту Вы имеете возможность продать (как бы того, чего у Вас нет). Например, при продаже двух акций, Ваша позиция станет равной -2.

Вариант 2. “Для тех, кому надоели финансовые данные”.
Найти все часто встречаемые комбинации доменов в историях посещений Интернета пользователями с помощью алгоритма Apriori (http://rakesh.agrawal-family.com/papers/vldb94apriori.pdf).
Использовать кеширование необходимой информации из лог-файла в ОЗУ Spark.
Предложить эвристику для отфильтровывания наиболее интересных частых наборов из всего множества частых наборов (варианты эвристики - удаление самых частых, введение меры интересности типа: во сколько раз вероятность набора отличается от произведения вероятностей его элементов).

Входные параметры реализации:
<имя входного файла>
<имя выходного файла1>
<имя выходного файла2>
<min_support> - минимальное число пользователей, у которых должна встретиться данная комбинация доменов, чтобы считаться часто встречаемой.
Формат выходных данных:
Выходной файл1 - содержит частые наборы в формате:
<частота> <домен1> <домен2> … <доменN>
Сортировка по: размеру набора, набору (наборы можно сравнить как списки лексикографически сравнимых элементов).

Выходной файл2 - содержит потенциально интересные частые наборы в формате:
[<мера интересности>] <частота> <домен1> <домен2> … <доменN>

Сортировка по невозрастанию меры интересности (если Вы ее придумали и используете), либо сортировка как в выходном файле1 - если используете обрезку по каким-то простым фильтрам.

Прислать выходной файл 1 из результатов запуска для пользователей с int(идентификатор)<3000 и min_support=2

Данные: wasb://web@bigdatamsu2.blob.core.windows.net/ (в качестве примера данных небольшого объема можно взять отдельный файл)
Дополнительное задание (можно делать тем, кому интересно или кто хочет улучшить ситуацию, если не успел сделать вовремя другие ДЗ. Дополнительное задание, решенное с ошибкой, не оказывает негативного влияния на оценку основного задания)
Также реализовать поиск ассоциативных правил.
Формат выходного файла - часть задания (т.е. его нужно придумать).

Общие требования
Любое из заданий должно быть сделано на PySpark.
В экспериментах должны быть использованы узлы класса A3.

Предлагается обратить внимание на разумную оптимизацию производительности при реализации (не следует кешировать в ОЗУ то, что не имеет смысла кешировать, оптимизировать размер кешируемых данных - не следует кешировать атрибуты данных, которые не используются).
Прислать конфигурацию параметров запуска (параметры spark-submit и конфигурации контекста), которые обеспечивают максимальную скорость исполнения.
Содержание архива должно быть аналогично предыдущим задачам.
Параметры программ, на которых будет производиться тестовый запуск и замер времени нами, будут приведены позднее.
Обратите внимание, данных стало больше чем в предыдущем хранилище. В финансовых данных возможны битые строки - ловите эксепшены в map, когда строка не разбивается на нужное число столбцов и т.п.

В присланном архиве должны оказаться:

- исходные тексты и файл сборки для них
- время исполнения на всех данных
- конфигурация исполнения для запуска на всех данных (
  для финансовой задачи:
  candle.widths = 5,10;
  candle.shifts=0,1,2
  при этом без ограничений на инструменты, даты и времена
  для apriori:
  min_support=600
)
- результат исполнения на отдельных “маленьких данных”