Лабораторная работа 7.

Разработка простой топологии storm.

1. Задача.

Требуется разработать топологию для составления частотного словаря файлов.

Требуется разработать Spout который будет опрашивать директорию и в случае обнаружения в ней новых файлов читать ее построчно и генерировать tuple в исходящий поток words для каждой строки. После завершения чтения файла Spout должен выдать tuple в исходящий поток sync. Также после окончания файла требуется перенести файл в папку для обработанных файлов

Требуется разработать Bolt Splitter который будет принимать строку и разбивать ее на слова.

Требуется разработать Bolt Counter который будет принимать слова из входящего потока Splitter и вести частотный словарь. Также по команде от входного потока sync — печатать текущий словарь на экране и обнулять его.

2. Подсказки.

а. Запуск standalone приложения с помощью maven осуществляется следующим образом

mvn exec:java -Dexec.mainClass="<Main class>"

б. Базовый класс spout — org.apache.storm.topology.base.BaseRichSpout

в. Базовый класс bolt — org.apache.storm.topology.base.BaseRichBolt

г. Парметры в spout передаются с помощью создания экземпляра класса org.apache.storm.Config и передачи его в cluster.submitTopology

д. Параметры извлекаются в spout в методе open, в параметрах которого есть Config

е. Читать файл построчно можно с помощью класса BufferedReader

например

reader = new BufferedReader(new InputStreamReader(new FileInputStream(file), Charsets.UTF_8));

ё. Перенести прочитанный файл из директории можно с помощью класса

com.google.common.io.Files используя метод move (класс Files содержится в библиотеке Guava подключенной к примеру файла проекта)

ж. В основном цикле spout в методе nextTuple мы должны проверить, читаем ли мы в данный момент файл. Если читаем то отправить вызовом метода SpoutOutputCollector.emit следующую строку в поток words. Если строки в файле кончились, то отправить сигнал в поток sync и перенести файл в папку

обработанных файлов.

Если файл не читаем, то просканировать директорию в поисках нового файла для чтения.

В случае если файла нет то вызовом org.apache.storm.utils.Utils.sleep простановить работу потока на время (например 100мс)

з. В первом bolt принимаем tuple содержащий строку текста и разбиваем ее на слова. Каждое слово отправляем в поток данных word

и. Во втором bolt принимамем слова и строим частотный словарь.

й. При получении сигнала из потока sync печатаем словарь и обнуляем.

к. Идентифицировать поток входящего tuple можно вызвав метод tuple.getSourceStreamId()

л. <Необязательно>Для более надежной работы spout можно считать количество отправленных tuple или запоминать msgid. И далее обрабатывать ack или fail.

После получения ack на все отправленные tuple генерировать sync.