Структура программы

Для хранения двух объектов типа Text в программе реализован WritableComparable класс TextPair.

Для удобной работы с датами и конвертации строк в даты реализован класс DateTimeStr. В нем реализован метод, вычисляющий начало свечи для произвольной отметки времени.

CandleMapper

- читает параметры конфигурации;
- получает строку csv-файла;
- проверяет, что эта строка не заголовок;
- проверяет, что дата расположена между *candle.date.from* и *candle.date.to*, а час расположен между *candle.time.from* и *candle.time.to*;
- проверяет, что название финансового инструмента удовлетворяет регулярному выражению *candle.securities*;
- для строки, удовлетворяющей перечисленным условиям, возвращает пару ключ TextPair(Symbol, Candle_start) и значение TextPair(ID, Price).

CandlePartitioner

- извлекает из ключа типа TextPair начало свечи;
- вычисляет количество миллисекунд, прошедшее с начала построения свеч;
- делит найденное количество миллисекунд на ширину свечи, тем самым вычисляя номер свечи;
- возвращает остаток от деления номера свечи на количество редьюсеров.

Такой Partitioner равномерно распределяет нагрузку на редьюсеры, так как для одного инструмента соседние свечи попадут на соседние редьюсеры.

CandleReducer

- итерируется по значениям TextPair(ID, Price) и находит цену, соответствующую минимальному ID, максимальную цену, минимальную цену и цену, соответствующую максимальному ID, то есть open, high, low, close соответственно;
- используя ключ TextPair(Symbol, Candle_start) и вычисленные цены, возвращает ключ Text - строку, в которой через запятую перечислены

Symbol, Candle_start, open, high, low, close и значение Text - пустую строку.

Эксперимент на больших данных

На приведенном графике по оси абсцисс - количество редьюсеров, по оси ординат - время в секундах, затраченное на обработку данных на кластере. Оптимальное число редьюсеров - в районе 32.