РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ПО АГРЕГАЦИИ И ХРАНЕНИЮ БИРЖЕВЫХ КОТИРОВОК

Бобер Е.Л.

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Образцова О.Н. – и.о. зав. кафедрой ИСиТ, к.т.н., доцент,
Малиновская Т.И. – ст. преподаватель

В работе представлена разработка программного средства, обеспечивающего агрегацию и хранение биржевых данных в режиме реального времени, и предоставляющее программный интерфейс для чтения исторических данных, а также передачу агрегированных данных в режиме реального времени.

Биржевые котировки - значение цены на некоторый товар которым торгуют сейчас на бирже. Существуют несколько способов представления биржами торговых данных. В данной статье рассматривается система агрегации сырых данных в формате BBO в формат OHLC с различным интервалами.

Структура формата **OHLC** (Open-High-Low-Close):

O — open (цена открытия интервала);

H — high (максимум цены интервала);

L — low (минимум цены интервала);

C — close (цена закрытия интервала).

Данный формат удобен для построения графика баров и японских свечей.

Минимальное сообщение в формате **BBO** (BestBidandOffer) содержит поля bid и ask. Получая поток сообщений в формате BBO система должны возвращать множество потоков данных в формате OHLC с различной агрегацией по времени. Наиболее часто используемыми агрегациями являются секундные (1, 5, 10, 30), минутные (1, 5, 15, 30), часовые (1, 2, 4, 8), дневные (1, 2, 7) и месячные (1, 3, 6). Также система должна предоставлять программный интерфейс для получения данных в любой из перечисленных агрегаций за заданый промежуток времени без задержки. Продолжительность операции выборки данных должны быть константной и не зависеть от количества исторических данных в системе.

Количество обрабатываемых сообщений в формате BBO для системы работающей с 10 биржами на каждой из которых будет 100 активов, по которым будет приходить 10 сообщений в секунду: 10 000 (100 * 10 * 10) сообщенией в секунду.

Приблизительный размер хранилища после года работы, если исходить из того что одна запись будет состоять из 5 полей по 128 бит, будет равен 3205 гигабайтам.

Данный расчёт не учитывает репликацию в кластере и не учитывает расходов на хранение метаданных СУБД, сырых данных бирж, и других дополнительных значений, необходимых для записи. Система хранение должна иметь линейный рост количества запросов к количеству нод в кластере.

Исходя из вышеизложенных требований и примерных расчетов нагрузки и объёмов данных, для построения системы хранения будем использовать СУБД «Cassandra».

Минимальный набор полей для хранения данных в формате OHLC:

- asset_idint;
- close_timetimestamp;
- openvarint;
- closevarint;
- highvarint;

– lowvarint.

Для обеспечения заданным параметрам масштабируемости необходимо произвести кластеризацию хранимых данных. Для этого создадим составной основной ключ состоящий из asset_id и искусственно добавленного поля update_date. Поле update_date будет содержать дату закрытия свечи в формате yyyy-mm-dd. Данный ключ позволяет создавать достаточно небольшие партиции данных, которые могут быть распределены между узлами кластера, и при этом позволяют строить относительно удобные запросы на выборку данных за необходимый период времени. Запрос создания таблицы будет выглядеть следующим образом:

```
CREATE TABLE candle_1s IF NOT EXISTS (
asset_id int,
  close_time timestamp,
  update_date text,
  open decimal,
  close decimal,
  high decimal,
  low decimal,
  PRIMARY KEY(
    (asset_id, update_date),
    close_time
  )
) WITH CLUSTERING ORDER BY (update_time DESC)
```

Алгоритм «агрегация» представляет собой каскад из буферов различного интервала. Первым является односекундный буфер, который накапливает BBO данные в течение 1 секунды и формирует из них свечу временным интервалом в одну секунду. После этого полученная свеча записывается в базу и передаётся в буфер большей размерности (в данной системе пятисекундный). Пятисекундный буфер аналогично односекундному буферу производит агрегацию данных в течени 5 секунд. После чего передаёт агрегированные данные вышестоящему буферу. При старте системы происходит инициализация агрегационного каскада путем считывания агрегированных данных из базы данных и прогона агрегации исходный данных. Данный процесс достаточно ресурсоемкий и при большом простое системы агрегация сырых данных может занимать значительное время.

Разработанное программное средство выполняет следующие функции:

хранение данных в формате BBO;

riагрегация данных в различные интервалы OHLC;

предоставление api для получение данных в формате OHLC;

стриминг данных в формате OHLC или BBO.

Список использованных источников:

1. Nishant Neeraj, Mastering Apache Cassandra - Second Edition. – Packt Publishing - ebooks Account; 2 edition (March 26, 2015) – 322 c.