Коннектор - подключение к источнику данных

Фидер - поток данных отдельного символа

# О системе

Многие финансовые компании и частные инвесторы для алгоритмического и

визуального анализа рынка используют рыночные данные о сделках. Эти данные

группируются в виде специфических кортежей и сохраняются в виде временного ряда

для последующего анализа.

Пример:

1) На мамбе сегодня с утра было зарегистрировано 36 тысяч различных тикеров.

2) Объем данных по наиболее ликвидным инструментам мамбы может достигать

десятков миллионов сделок в сутки. На западных площадках в виду большей

ликвидности объем данных возрастает на порядки.

3) Объем потока данных неравномерно распределен внутри дня и может скачкообразно

увеличиваться в часы наибольшей активности участников рынка.

Проблемы:

1) Брокеры и биржи неохотно предоставляют данную информацию.

2) В виду большого объема хранимой информации надежно организовать процесс сбора

данных на пользовательской машине не представляется возможным.

3) Данные о сделках нельзя перезаказать и получить повторно. Любые потери

при дальнейшем анализе дадут неправильный результат, что критично для финансовых

приложений

4) Нет унифицированного протокола передачи данных. Разные биржи и брокеры

используют различные технологии и протоколы для подключения клиентов. Реализация

возможности подключения ко всем необходимым площадкам может стоить дорого.

5) Брокеры требуют открытия счета для предоставления доступа к данным.

Цели:

1) Обеспечить унифицированный доступ к базе рыночных данных по любому

инструменту за любой период времени с типовыми периодами группировки.

2) Потоковый доступ в реальном времени (в перспективе)

# Базовые функции

\*) Выдача списка инструментов

\*) Выдача обновлений параметров инструментов

\*) Выдача сделок за сутки

\*) Выдача M1,M5,M10,M15,M30,H1,H3,H6,H12,D1,W,MN свечек

# Фазы готовности

1) Сохранение потока сделок. Формат должен совпадать с форматом сделок FINAM.

2) Надежное сохранение (2 фидера?)

2) Выдача сделок через REST API - не так критично для задач симуляции торгов.

Можно в любое время добавить эти фронты. А для симуляции можно использовать

прямой коннект к базе. Этот же самый драйвер использовать для быстрой реализации

фронтов. Общая нетривиальная проблема, которую нужно решить при реализации

доступа к данным - это индексирование данных. Сделок по инструменту может быть

миллионы в день.

Цель: Обеспечить сбор хранение и выдачу рыночных данных L1 (только сделки) и

OHLCV в различных таймфреймах по инструментам из различных источников. MVP не

расчитана на подачу данных в реальном времени, хотя и может предоставлять такую

возможность. На выходе система по запросу с указанием периода через API

система выдает:

[ time, price, volume ] - для сделок

[ time, open, high, low, close, volume ] - для свечей

Типы узлов в системе:

1) Входные узлы подключаются к поставщику данных, парсят и конвертируют данные

в универсальную форму, сохраняют в кафку. Узлы в принципе одноранговые, хотя

могут содержать код оптимизированный под конкретный провайдер данных. Требования

к этим узлам диктуюся конкретным поставщиком данных. Например, для мамбы и

транзака это винда с большим объемом памяти и хорошим CPU для парсинга XML

с либой в виде dll. Минимальное количество узлов на коннектор - 2, что бы

обеспечить избыточность в случае сбоев, именно эти узлы более всего будут

подвержены сбоям в виду специфики работы. На мамбе суммарно около 40к символов

в списке. Высокая вероятность, что через один коннектор такой объем не прокачать

или провайдер забреет, сломается и тп.

2) Узлы входящие в кафка кластер. Нужны хорошие сетевухи, быстрые, надежные

харды и большой объем памяти под кэши. CPU не критично. Минимальное количество

узлов - 3 для обеспечения минимальной надежности и сохранения скорости. Такой

конфиг выдержит падение одного из серверов кластера кафки с несильной просадкой

производительности, но при сохранении надежности. Нужно минимум 2 постоянно

рабочих узла, что бы паралельно обработать как минимум пару фидеров.

3) Выходные узлы обрабатывают запросы конечных пользователей. Это может быть

REST API, по типу как в большинстве криптобирж, FIX API или кастомный протокол.

Минимальное количество узлов - 1. Требования к железу минимальны.

Мысли по поводу балансировки фидеров.

Для мамбы можно использовать следующий подход:

1) При отсутствии статистики просто рандомно разделить символы между

доступным количеством серверов.

2) После закрытия вечерки считаем статистику по какому символу сколько

данных набежало. Сортируем от наиболее ликвидных к наименее ликвидным, после

чего составляем списки, отбирая поочереди для каждого из доступных серверов.

Перед началом следующей сессии ноды запускают фидеры в зависимости от списка.

Минимальная конфигурация

Запустить все узлы на 1 хосте. Никакой надежности и скорости, но можно тестировать функционал.

Минимальная продакт-конфигурация

3 хоста под кафку, 2 хоста для фидеров, 1 хост под REST API

Идеи дальнейшего развития:

1) Так же можно сделать конверторы в разные форматы, например в csv.

Но это будут отдельные ноды и они будут жрать CPU.

2) Можно на морду прикрутить классные графики, через которые потом проспамить.

Но как это сделать я не знаю, это достаточно сложная тема и нужен фронтендщик.

Кстати, библиотека отрисовки графиков реалтайм может стать частью продукта.

Наработки есть, но этим надо заниматься отдельно.

3) Подключить крипту можно, но там есть тонкости, связанные с типами данных.

Но если это сделать, получится офигенный инструмент для анализа вообще всего.

4) Можно продавать аналитику. Просто отдельными сервисами, которые берут

исходные потоки, анализируют и сохраняют в удобном виде.

5) Можно продавать слепок базы или доступ непосредственно к кафке.