

Реализация агрегатора книг заявок валютных рынков для торговой платформы Tbricks

Абрамов Иван Александрович

научный руководитель: Константин Николаевич Романов

СПб АУ НОЦНТ РАН

13 июня 2017 г.

- Валютный рынок не имеет единой централизованной биржи
- Торговые площадки предоставляют информацию в виде книг заявок

Таблица: Пример книги заявок

Покупка	Продажа	Объем
1.3505	1.3506	1000000
1.3504	1.3507	3000000
1.3503	1.3508	5000000
1.3502	1.3509	10000000

- Агрегация книг заявок
- Вычисление значения технического индикатора
 - Volume Weighted Average Price

$$VWAP = \frac{\sum_{i=1}^n price_i \cdot volume_i}{\sum_{i=1}^n volume_i} \quad (1)$$

- Автоматическая отправка новых заявок на рынок

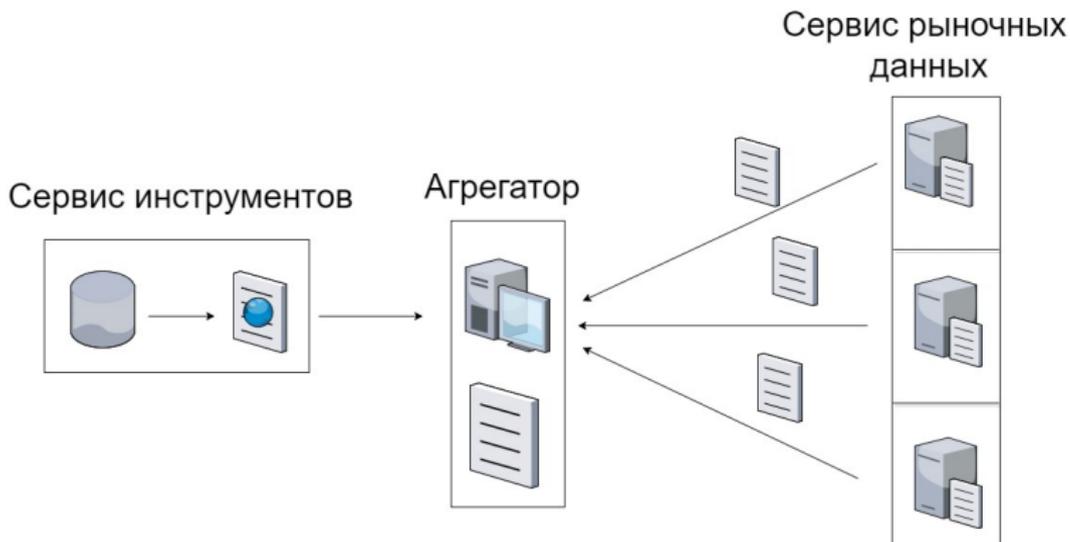
- Большинство существующих решений являются частью программного комплекса и не расширяются под нужды клиента
- Tbricks — это платформа для автоматической торговли на электронных биржах
- Tbricks приложение — это плагин, который загружается в сервис и обрабатывает внешние события
- Требуется создать модуль для агрегации книг заявок, который можно использовать при разработке Tbricks приложений

Цель: реализовать модуль для агрегации книг заявок валютных рынков для приложений торговой платформы Tbricks

Задачи:

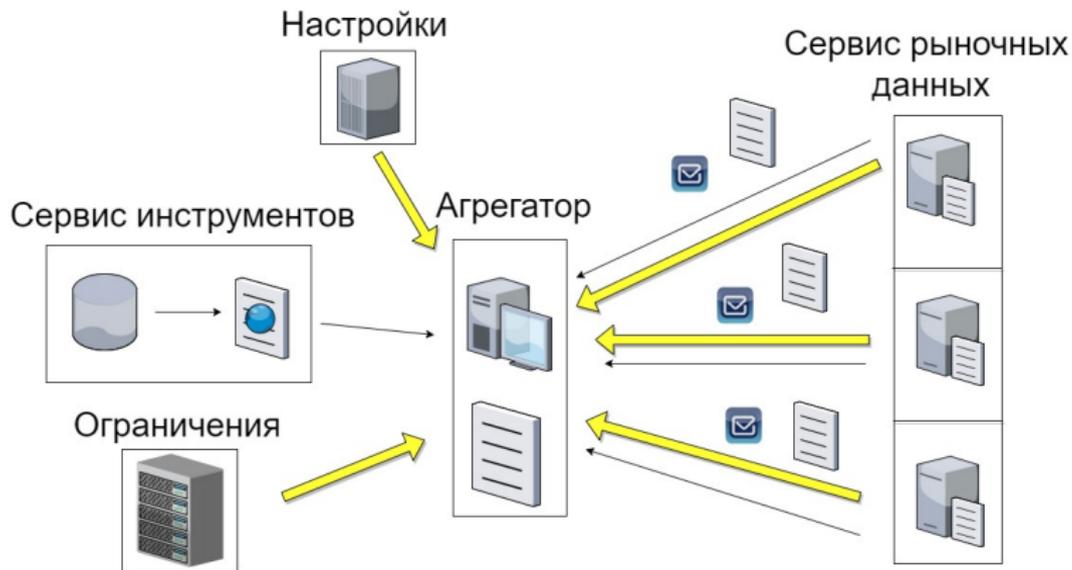
- реализовать модуль для обработки и агрегации книг заявок:
 - обрабатывать книги заявок разных типов
 - учитывать ограничения торговых площадок
- разработать приложение для автоматической отправки заявок на рынок:
 - реализовать сценарии исполнения заявок
- протестировать реализованные решения

Реализация модуля для агрегации



- Получить валютные инструменты (40 площадок)
- Обработать потоки рыночных данных (обновления до 100 раз в секунду)
- Сформировать агрегированную книгу заявок

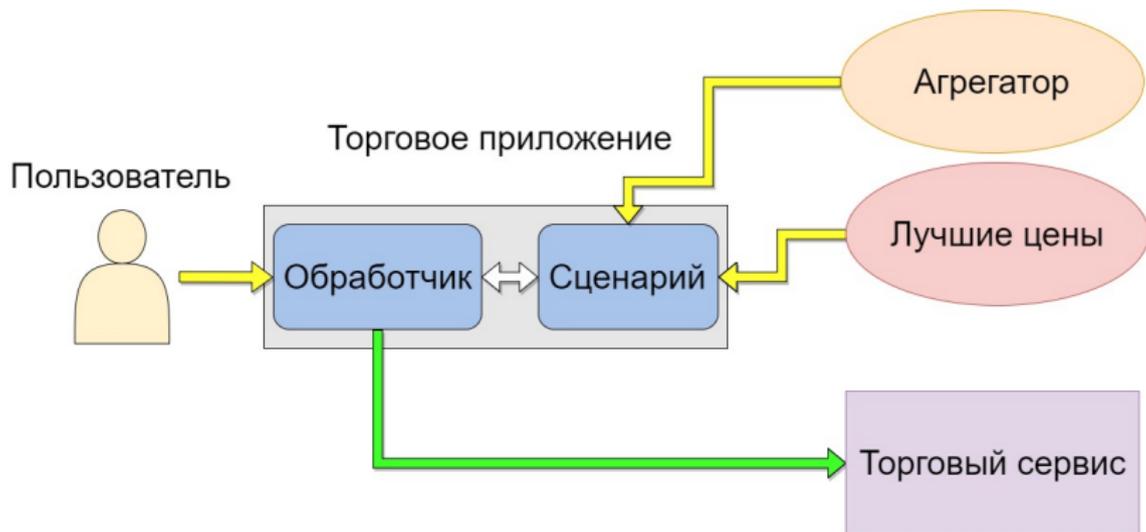
Реализация модуля для агрегации



- Использовать данные о статусе потоков
- Учитывать ограничения торговых площадок
- Обрабатывать пользовательские настройки

- Разработанное приложение состоит из:
 - обработчика
 - торгового сценария
- Задачи обработчика:
 - проверять корректность входных данных
 - создавать и отслеживать заявки
 - блокировать торговые площадки
- Задачи сценария:
 - проводить мониторинг рыночных данных
 - определять момент для отправки заявок
 - формировать параметры заявок
 - принимать решение о завершении стратегии

Реализация торгового приложения



- Реализованы торговые сценарии:
 - торгующие на одном рынке (Join, Iceberg)
 - торгующие на нескольких рынках (Take, VWAP)
 - анализирующие и торгующие на разных рынках (Peg, Stop Loss, Aggressive Watch)

- Тестирование сценариев при помощи фреймворка TestEngine:
 - настройка виртуальных сервисов платформы
 - проверка параметров исходящих заявок
 - проверка реакции на изменение рыночных данных
 - проверка корректного завершения
- Создано 33 TestEngine теста для различных сценариев
- Тесты интегрированы в CI тест-план

- Существующие ограничения:
 - 40 торговых площадок
 - максимальная глубина книги заявок 50 уровней
 - обновление до 100 раз в секунду

Таблица: Результаты измерений

Перцентиль	Время (мкс)
99%	65
95%	62
80%	54
50%	47

- Стандартное отклонение 1.5 - 2 мкс

- Реализован модуль для обработки и агрегации книг заявок для Tbricks приложений
- Разработано Tbricks приложение для автоматической отправки заявок на рынок:
 - реализовано 7 различных торговых сценариев
- Проведено тестирование:
 - реализовано 33 теста на фреймворке TestEngine
 - проведено тестирование производительности системы
- Агрегатор и торговое приложение внедрены в репозиторий Tbricks приложений

- Реализован модуль для обработки и агрегации книг заявок
 - 17 классов, 5500 строк
- Разработано приложения для автоматической отправки заявок на рынок:
 - 13 классов, 4600 строк
- Проведено тестирование:
 - реализовано 33 теста на фреймворке TestEngine, 2400 строк
 - реализовано 18 модульных тестов на фреймворке Google Test, 1100 строк

Торговый сценарий Stop Loss

