Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.942

Мацерук
Евгений Анатольевич

Система обработки больших объемов информации для прогнозирования изменений цен на бирже

АВТОРЕФЕРАТ
на соискание степени магистра информатики и вычислительной техники
по специальности 1-40 81 04 «Обработка больших объемов информации»

_____

Научный руководитель
Теслюк Владимир Николаевич
доцент, кандидат физико-
математических наук

_____

Минск 2018

КРАТКОЕ ВВЕДЕНИЕ

В настоящее время сложно представить процесс торговли на бирже без анализа текущей ситуации. Существуют разнообразные наборы методов для прогнозирования изменений цен, которые широко используются различными автоматизированными системами.

Использование автоматизированных систем, для прогнозирования изменений цен позволяет участнику торгов учитывать огромное количество параметров при принятии решение о покупке или продаже актива. Кроме этого, использование различных методов для анализа в автоматизированных системах позволяет лишить процесс анализа человеческого фактора, что значительно снижает вероятность ошибки при прогнозировании.

Прогнозирование изменений цен на бирже является сложной задачей и на данный момент никто не может формализовать алгоритм для получения точного результата. Как показала практика, для того, чтобы повысить качество прогнозирования, необходимо учитывать информацию не только о предыдущих ценах актива (технический анализ), но и принимать во внимание другие факторы, например, информацию из СМИ и социальных сетей. Поэтому проектирование автоматизированной системы такого рода связано с рядом проблем:

1) Необходимо обеспечить сбор информации из различных источников.

2) Хранилище системы должно быть гибко масштабируемым, т.к. его объем будет постоянно увеличиваться с большой скоростью.

3) Вычисление результатов прогнозирования должно быть начаты после изменения цены на актив или же после получения новой информации о активе из других источников.

4) Выполнение операций анализа и прогнозирования изменений цен не должно превышать временного интервала, на котором производится торговля. Например, для интервала равного одной минуте, выполнение операции прогнозирования должно занимать меньше одной минуты.

5. Доступ к результатам прогнозирования должны быть обеспечен с насколько это возможно малой задержкой.

Для того, чтобы решить все описанные проблемы и реализовать все требования предъявляемые к системе прогнозирования изменений цен необходимо применять последние достижения в сфере обработки данных: новые архитектурные подходы, облачные технологии, нереляционные хранилища информации.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель и задача исследования

Проблема: проектирование и реализация системы обработки больших объемов информации для прогнозирования изменений цен на бирже.

Целью диссертационного исследования является проектирование и реализации системы обработки больших объемов информации для прогнозирования изменений цен на бирже на основе данных о цене актива, а также данных из социальных сетей, связанных непосредственно с биржевым активом.

Для достижения поставленной цели необходимо выполнить следующие действия, которые и будут являться задачами исследования:

– сделать обзор источников информации, необходимых для прогнозирования изменений цен;

– рассмотреть методы прогнозирования изменений цен;

– спроектировать систему обработки больших объемов информации для прогнозирования изменений цен;

– реализовать и протестировать спроектированную систему.

Объектом исследования является система обработки больших объемов информации.

Предметом исследования является проектирование и реализации системы обработки больших объемов информации.

Гипотеза исследования заключается в том, что возможно повысить качество прогнозирования изменений цен на бирже за счет анализа не только цены актива, а и данных из разнообразных источников информации связанной с активами, например, социальными сетями, и получить результат мы можем только если сможем обработать большие объемы неоднородной информации.

Связь работы с приоритетными направлениями научных исследований

В век цифровых технологий все больше и больше задач возлагается на системы обработки информации, которые позволяют совершать невероятные научные открытия, находить лекарства от ранее неизлечимых болезней, покорять новые и новые пространства космоса. Всё это стало возможным с появлением современных технологий обработки информации, которые позволяют анализировать огромные массивы данных и находить в них закономерности. В работе рассматривается проектирование и реализация

системы, которая позволит с более высокой вероятностью спрогнозировать изменения цен на бирже.

Личный вклад соискателя

Результаты, приведенные в диссертации, получены соискателем лично. Вклад научного руководителя, В.Н. Теслюка, заключается в формулировке целей и задач исследования.

Структура и объем диссертации

Пояснительная записка по диссертационной работе включает в себя оглавление, общую характеристику работы, введение, основную часть, состоящую из 5 глав, заключения и списка литературы.

Первая глава содержит обзор и анализ предметной области, определение методов прогнозирования изменений цен, обзор источников информации.

Вторая глава содержит теоретические описание методов, шаблонов и технологий, которые были применены для проектирования и реализация системы.

Третья глава содержит подробное описание спроектированной архитектуры автоматизированной системы.

В четвертой главе описываются детали реализации некоторых компонентов системы с примерами кода.

Пятая глава рассматривает процесс тестирования системы.

В заключение подводятся итоги и делаются выводы по работе, а также описывается дальнейший план развития проекта.

Общий объем работы составляет 51 с., 15 рис., 0 табл., 28 источников.

ОСНОВНОЕ СОДЕРЖАНИЕ

Во **введении** определена область и указаны основные направления исследования, показана актуальность темы диссертационной работы, дана краткая характеристика исследуемых вопросов, обозначена практическая ценность работы.

В **первой главе** сделан обзор методов для прогнозирования изменений цен на бирже с помощью технического анализа. Также были сформулированы принципы и описаны формулы по которым будут вычисляться значения для последующего их использования в прогнозировании изменений цен.

Также в этой главе сделан краткий обзор источников первичных данных, с которыми будет взаимодействовать автоматизированная система. В качестве основных были выделены CryptoCompare.com Web API, Twitter Web API, Информационно-статистический сервер Московской биржи. Были рассмотрены принципы взаимодействия с ними, а также способы получения новой информации из этих источников.

Вторая глава содержит обзор теоретических основ, которые необходимы для проектирования и реализация системы.

В первую очередь были рассмотрена шаблон *Лямбда*-архитектура, основными компонентами которого являются: *Batch Layer, Speed Layer и Serve Layer*, которые представлены на рисунке 1.

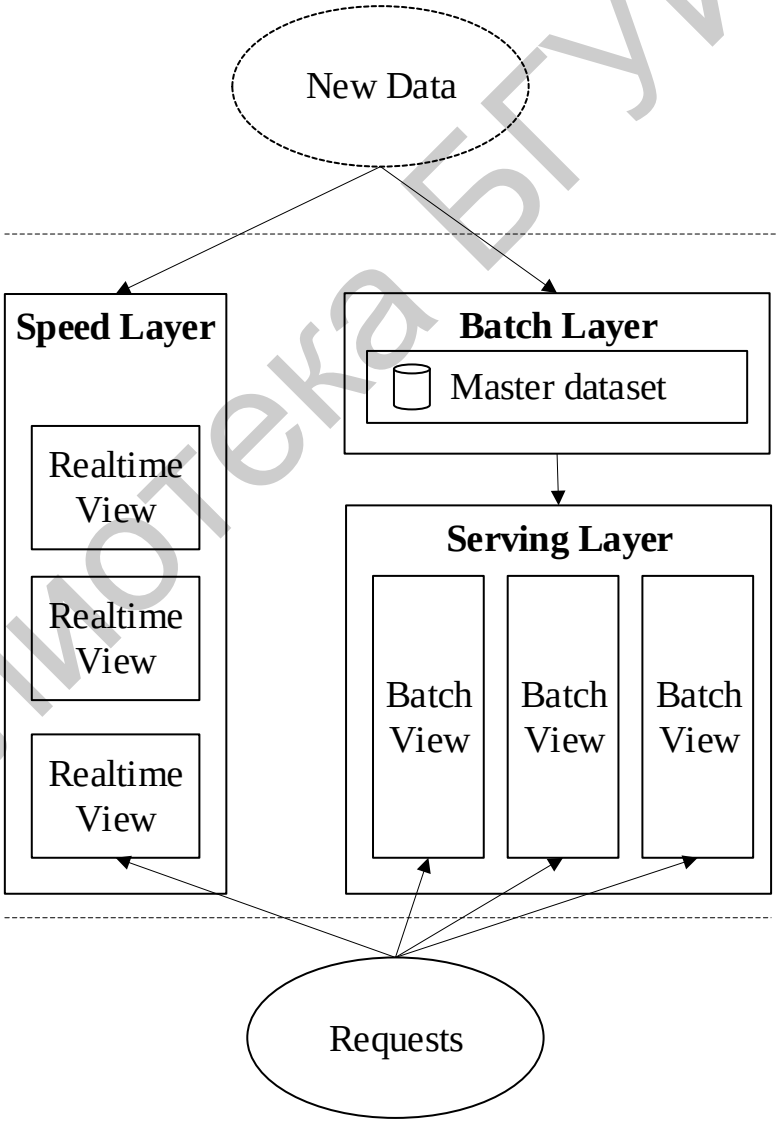

Рисунок 1 – Схема Лямбда-архитектуры

Были выделены преимущества и недостатки данного шаблона и рассмотрен другой шаблон для проектирования систем обработки больших объёмов информации *Kappa*-архитектура. Главный отличием от Лямбда-архитектуры является то, что в *Kappa*-архитектуре отсутствует *Batch Layer* и вся обработка данных производится только в компоненте *Speed Layer*. Концептуальная схема *Kappa*-архитектуры представлена на рисунке 2.

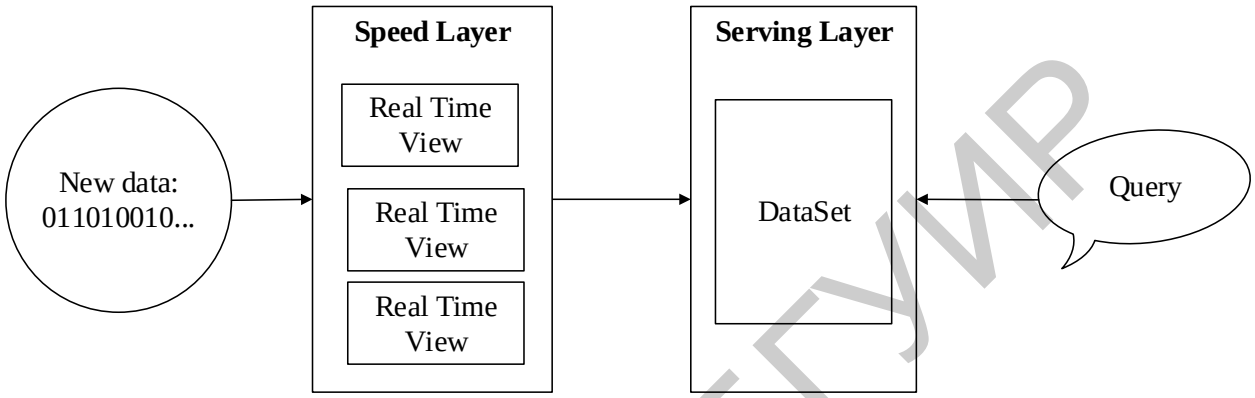

Рисунок 2 – Схема Kappa-архитектуры

В качестве основы для реализации обработки больших объемов информации в реальном времени была выбрана система *Apache Storm*. Были рассмотрены и проанализированы основные практики, применяемые при выполнении решений на основе этой системы, базовые элементы которыми оперирует система, а также сделан краткий обзор архитектуры системы. На рисунке 3 приведен пример топологии *Apache Storm* с основными элементами:

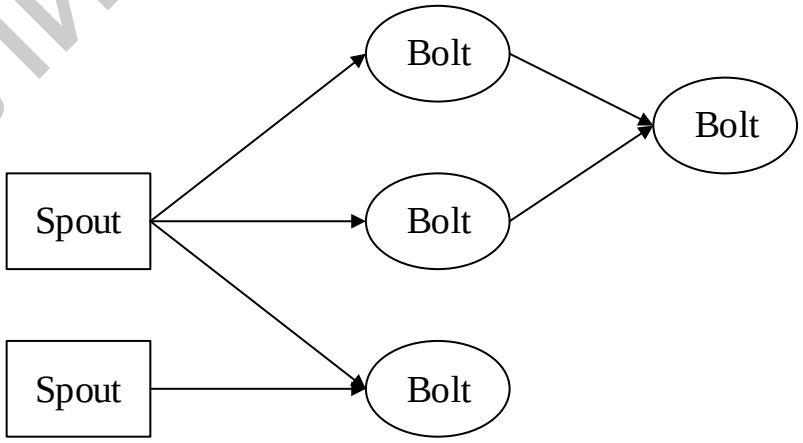

Рисунок 3 – Пример топология Apache Storm

Для задачи проектирования и реализации хранилища системы обработки больших объемов информации были рассмотрены нереляционные базы данных.

Среди них были особо выделены документ-ориентированная база *MongoDB*, а также база данных *Redis*. *MongoDB* была использована в качестве основного хранилища, а *Redis* в свою очередь в качестве кэша данных, за счет того, что он обладает более высокой скоростью обработки данных по сравнению с другими NoSQL базами.

После проектирования и реализации системы, её необходимо развернуть, и для этих целей была рассмотрена облачная платформа Microsoft Azure для разработки, тестирования и управлениями приложениями. Данная платформа предоставляет множество типов решений и поддерживает большое количество фреймворков и языков программирования, и именно поэтому она была выбрана в качестве основы для развертывания приложения.

Значимая часть в главе была уделена анализу возможностей сервисов, которые предоставляет данная облачная платформа: *HDInsight, CosmosDB, Text Analytics*.

Одним из важнейших вопросов в области разработки программного обеспечения является его тестирования и именно для этих целей были рассмотрены лучшие практики применяемы для контроля качества разрабатываемого программного обеспечения.

С целью организации модульного тестирования, была рассмотрена разница между «одиночными» и «общительными» (рисунок 4) тестами, и ситуации, когда стоит применять тот или иной подход.

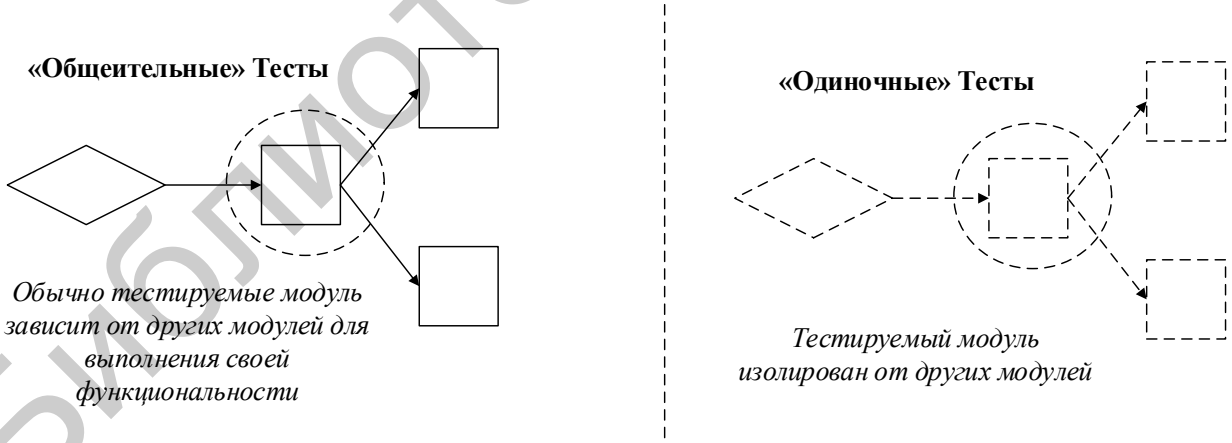

Рисунок 4 – «общительные» и «одиночные» тесты

Для «одиночных» тестов характерно отсутствие прямого вызова кода других модулей, для того чтобы избежать непредвиденного неудачного завершения теста из-за ошибки в другом модуле. Как правило, для реализации «одиночных» тестов используются подход с применением *TestDoubles.*

Кроме модульных тестов для организации автоматизированного тестирования программного обеспечения используют и другие типы тестов: компонентные и так называемые *BroadStack* тесты. Совокупность всех типов тестов образуют пирамиду тестов (рисунок 4) – это практика реализации различных уровней тестирования программного обеспечения, которая ставит перед собой целью найти баланс между различными видами автоматических тестов.

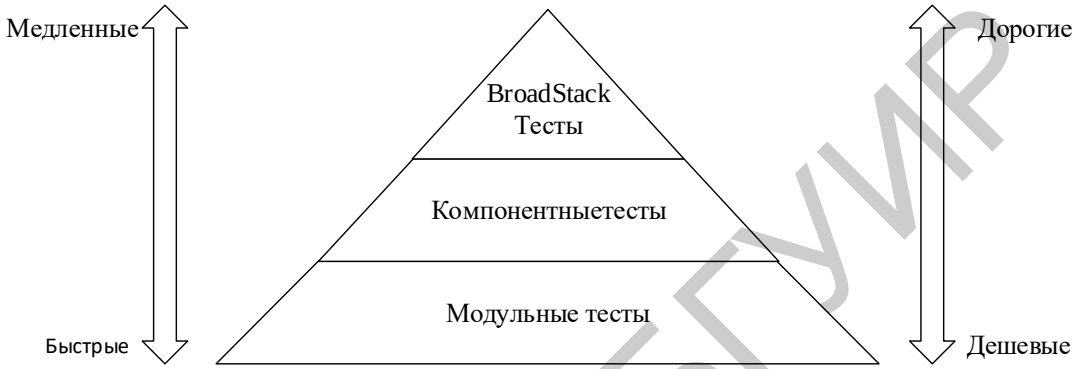

Рисунок 4 – Пирамида тестов

В настоящее время, набор автоматизированных тестов широко используется для организации процесса непрерывной интеграции и доставки приложений.

Непрерывная интеграция (CI) - это практика, используемая командами разработчиков для автоматизации слияния и тестирования кода. Внедрение CI помогает найти ошибки на этапе разработки программного обеспечения, что позволяет сэкономить ресурсы на их исправлении.

Непрерывная доставка (CD) - это процесс, посредством которого код создается, тестируется и развертывается в одной или нескольких тестовых средах.

Для выполнения этих практик при разработке программного обеспечения было решено использовать Visual Studio Team Services. Это связано в первую очередь с тем, что эти сервисы легко могут быть интегрированы с облачной платформой Microsoft Azure.

Третья глава описывает архитектуру автоматизированной системы обработки больших объемов информации.

Перед тем как приступить к реализации любой информационной системы, в первую очередь необходимо спроектировать её архитектуру. Архитектура данной системы была разделена на несколько модулей, каждый из которых должен отвечает за определенный набор задач. Это сделано для достижения

расширяемость, надежности и высокой отказоустойчивости системы.

Основными модулями архитектуру являются (рисунок 5):

1. *Модуль обработки информации и прогнозирования изменения цен.*
2. *Модуль отображения информации и доступа к системе.*
3. *Модуль доступа к информации*
4. *Хранилище информации*

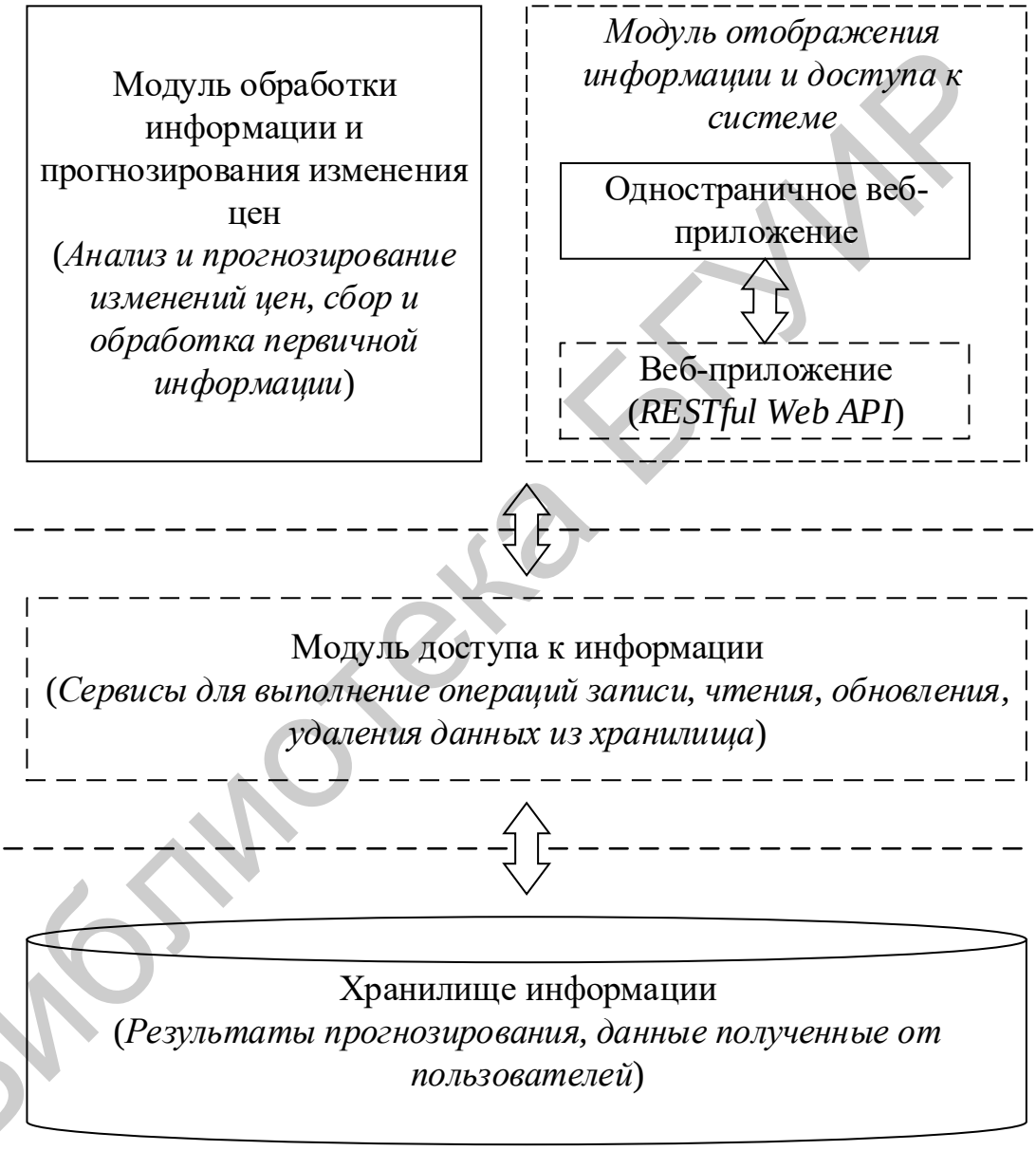

Рисунок 5 – Концептуальная архитектура автоматизированной системы

После того, как была спроектирована концептуальная схема архитектуры автоматизированной системы, была разработана архитектура каждого модуля в отдельности.

Для проектирования модуля обработки информации и прогнозирования изменений цен, была взята Карра-архитектура, и на её основе была построена

концептуальная архитектура системы. Т.к. решение разрабатывается на базе такой распределенной системы обработки больших объемов информации в реальном времени как Apache Storm, модуль обработки информации и прогнозирования изменения цен необходимо спроектировать как топологию Apache Storm, и в соответствии с этим представить все архитектурные компоненты в качестве Spout и Bolt, которыми оперирует топология данной вычислительной системы (рисунок 6).

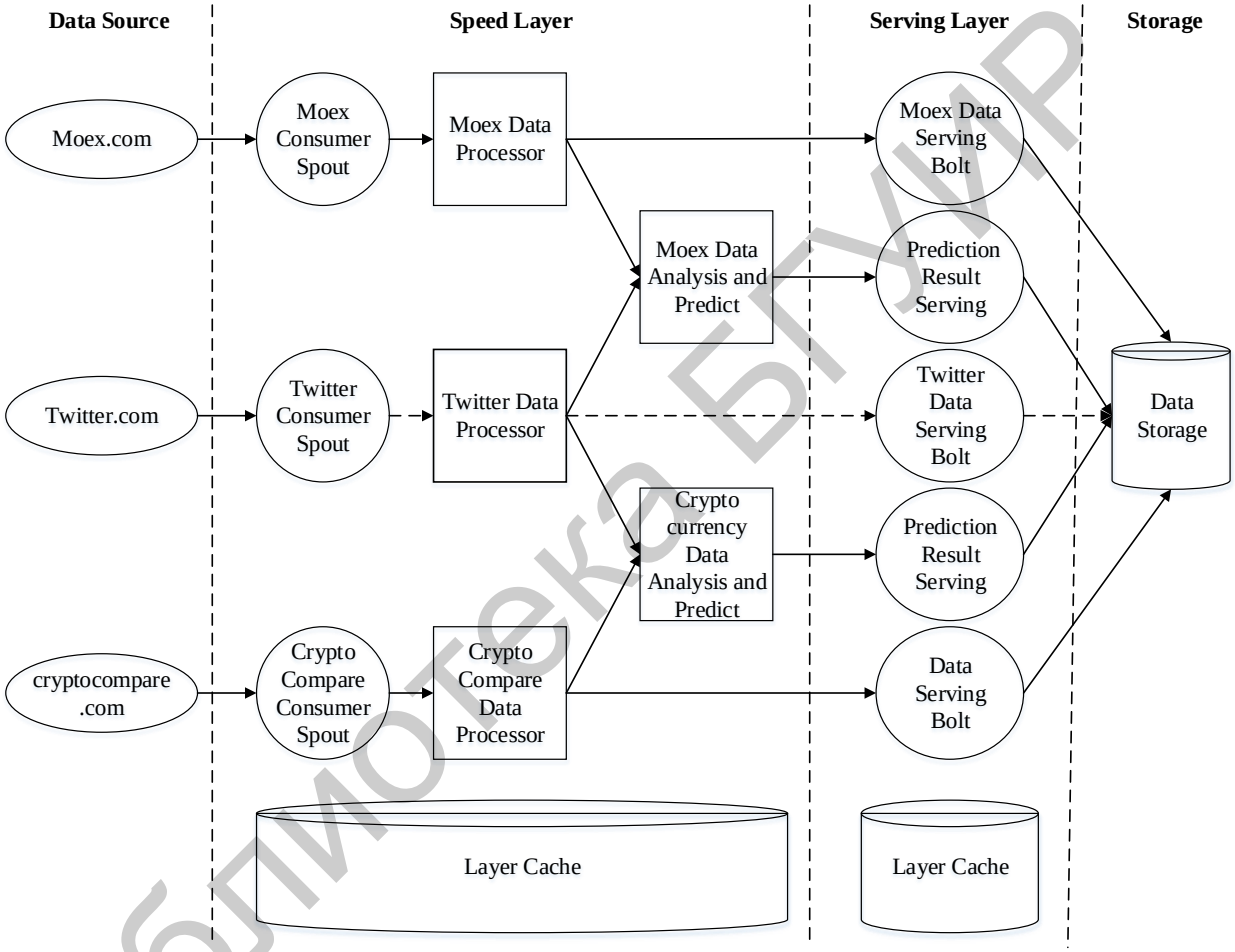

Рисунок 6 – Схема Apache Storm топологии

Для проектирования модуля отображения информации и доступа к системе были взяты принципы многоуровневой архитектуры, и были выделены следующие основные урони:

– уровень доступа к данным;

– уровень бизнес логики;

– уровень представления.

При проектировании хранилища было решено отказаться от реляционной базы данных в пользу нереляционной по следующим причинам:

– нереляционные базы данных позволяют быстро и при малых затратах менять структуру хранимых данных и нет необходимости поддерживать сложную схему базы и проще писать код для миграции данных;

– необходимо обеспечить более высокую скорость доступа к данным для записи, и здесь у нереляционных данных есть ощутимое преимущество

– мы имеем дело с большими объемами информации, и рано или поздно возникнет проблема масштабирования хранилища. Здесь нереляционные базы данных хорошо себя зарекомендовали, т.к. их масштабирование требует меньших усилий со стороны разработчиков по сравнению с реляционными.

Для доступа к хранилищу был спроектирован модуль доступа к информации, на который были возложены обязанности по выполнению всех необходимых операций: чтение, обновление, добавление и удаление информации.

В **четвертой главе** описаны некоторые детали реализации системы с примерами кода.

В первую очередь мы разобрали детали реализации хранилища и принципы, по которым он был разделен на различные базы данных.

После того, как хранилище спроектировано, необходимо предоставить к нему доступ, поэтому были рассмотрены базовые принципы на которых основывается модуль доступа к информации, а также представлены детали реализации, для подключения к хранилищу и получению информации.

Важнейшей частью системы является модуль анализа информации и прогнозирования изменений цен на бирже. Для его реализации используются система обработки информации *Apache Storm* для правильного функционирования которой необходимо реализовать топологию, а после отправить её на выполнение.

Исходным узлом любой топологии является *Apache Storm Spout*. Для того, чтобы его реализовать на платформе *.NET Framework,* необходимо использовать механизмы платформы *SCP.NET,* которая позволяет определить всю необходимую логику для функционирования топологии.

После того, как Apache Storm Spout получает информации из внешних систем, он передает её другому элемент *Apache Storm Bolt,* который можно реализовать использую всю ту же платформы SCP.NET.

Пятая глава содержит информацию о процессе организации тестирования системы.

В первую очередь были реализованы модульные тесты и процесс. После чего были реализованы более сложные тесты: компонентные, количество которых значительно меньше модульных. В самую последнюю очередь были выполнены нагрузочные тесты, которые позволили протестировать отказоустойчивость и надежность системы.

ЗАКЛЮЧЕНИЕ

Основные результаты диссертации

Данная работа демонстрирует процесс проектирования и реализации автоматизированной системы обработки больших объемов информации для прогнозирования изменений цен на бирже.

1) Была проработана предметная область, описаны подходы, применяемы для прогнозирования изменений цен. Также были рассмотрено несколько источников информации, которые используются системой для вычислений результата.

2) Для выполнения требований, предъявляемых к системе, был сделан обзор современных методов, практик и технологий, применяемых для проектирования и реализации систем обработки больших объемов информации. Было рассмотрено два архитектурных шаблона для проектирования систем обработки больших объемов информации и сделан осознанный выбор в пользу *Kappa*-архитектуры, из-за того, что она больше подходит для реализации данной конкретной системы.

3) Для реализации модуля обработки информации и прогнозирования изменений цен на базе системы *Apache Storm*, была разработана топология для обработки информации. После чего разработанная топология была успешна развернута на баз облачной платформы *Microsoft Azure*.

4) В целях хранения данных, было спроектировано хранилище информации и организован доступ к нему через отдельный модуль доступа к данным.

5) Для отображения результатов прогнозирования изменений цен был спроектировано и разработано веб-приложение, которое было также развернуто в облаке *Microsoft Azure*.

Все используемые принципы, шаблоны и технологии, были детально изучены и наиболее важные их аспекты отражены в работе, и стоит сказать, что выбор в пользу каждой из них был сделан на основе их преимуществ и недостатков.

Рекомендации по практическому применению результатов

1) Полученные результаты формируют теоретическую и практическую базу для разработки программного обеспечения, которое оперирует с большими объемами информации. Они могут быть использованы для модернизации и дальнейшего развития как существующих систем, так и создания новых, которые ориентированы в основном на обработку больших объемов информации.

2) Разработанная архитектура может применятся для реализации систем связанных с обработкой больших объемов информации в реальном времени.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

1-А. Мацерук, Е.А. Apache Storm – распределенная вычислительная система реального времени / Е.А. Мацерук // Технические науки: проблемы и решения: сб. ст. по материалам V Международной научно-практической конференции «Технические науки: проблемы и решения». – № 5(4). – М., «Интернаука», 2017.

2-А. Мацерук, Е.А. Применение Lambda-архитектуры на практике / Е.А. Мацерук // Технические науки: проблемы и решения: сб. ст. по материалам VI Международной научно-практической конференции «Технические науки: проблемы и решения». – № 6(5). – М., Изд. «Интернаука», 2017