

$$y_i = 1,176 - 0,043y_{t-1} - 0,023y_{t-2} + \xi_t,$$

в российских рублях (RUB/USD) – модель ARIMA (2,2,2):

$$y_t = 1,176 - 0,012y_{t-1} + 0,004y_{t-2} + \xi_t,$$

в евро (EUR/USD) – модель ARIMA (2,2,2):

$$y_t = 1,176 - 0,003y_{t-1} + 0,025y_{t-2} + \xi_t.$$

Все модели проверены на адекватность, которые могут использоваться для прогнозов курсов валют на фондовых рынках.

Литература

1 Финансовые новости: статьи, оценки, аналитика мирового финансового рынка [Электронный ресурс]. – Режим доступа : <http://www.finanz.ru/>. – Дата доступа : 01.10.2019.

2 Кобзарь, А. И. Прикладная математическая статистика. Для инженеров и научных работников / А. И. Кобзарь. – Москва : ФИЗМАТЛИТ, 2006. – 816 с.

УДК 004.67

М. Н. Гавриленко

ОДИН ИЗ ПОДХОДОВ К ОБРАБОТКЕ БОЛЬШИХ ОБЪЕМОВ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ APACHE SPARK, APACHE KAFKA И ELASTICSEARCH

Рассматриваются практические вопросы применения современных информационных технологий для обработки и хранения большого объема данных. Описывается один из подходов к созданию системы по обработке данных в режиме реального времени, хранения и отображения информации с использованием инструментов Elasticsearch и Kibana. Результатом работы приложения является многофункциональная система, обеспечивающая оперативный доступ, обработку и отображение больших объемов информации.

Введение. Увеличение информации в объемах, с одной стороны, а так же появление и быстрый рост новых информационных технологий, с другой стороны, требуют разработки и использования новых инструментов и методов, отвечающих современному уровню развития технических средств обработки большого количества данных. Огромные корпорации нуждаются в анализе своих данных и оперативном отображении их с использованием ресурсов Интернет. В связи с этим началась разработка и использование новых технологий, которые удовлетворяют основным требованиям пользователей: скорости обработки данных, качеству и простоте использования.

В настоящее время разработка современных приложений, как правило, не обходится без применения информационных технологий обработки и хранения больших данных. Поэтому актуально рассмотрение вопросов использования современных информационных технологий обработки больших объемов данных, о которых пойдет речь в данной статье.

Функциональные возможности использования подобных технологий и языков программирования рассматриваются на примере проекта. Суть приложения заключается

в том, чтобы получать данные в режиме реального времени, которые приходят с какого-либо устройства, попадают в Apache Kafka и обрабатываются с помощью Spark Streaming. Spark Streaming в свою очередь является подписчиком для брокера сообщений. Далее обработанные данные попадают в Elasticsearch и отображаются с помощью Kibana, тем самым пользователь может увидеть всю последнюю информацию своего устройства. Для простоты демонстрации подхода, работу по отправке данных с устройства имитирует Java-приложение. Подробно рассматриваются перспективы использования брокера-сообщений Kafka, а также Spark фреймворка для обработки и хранения информации.

Инструментарий для анализа и обработки Big Data. На сегодняшний день становится популярным изучение и применение технологий Big Data для анализа и обработки большого количества информации, которая может быть как структурируемой (данные JSON формата, CSV, и др.), так и не структурируемой (файлы логирования) [1].

Apache Spark (рисунок 1) позволяет читать файлы данных различных форматов и объемов, также позволяет читать Streaming данные (во время загрузки данных). Реализация Spark написана на Scala – языке программирования, который включает в себя как объектно-ориентированную парадигму, так и функциональную [1].

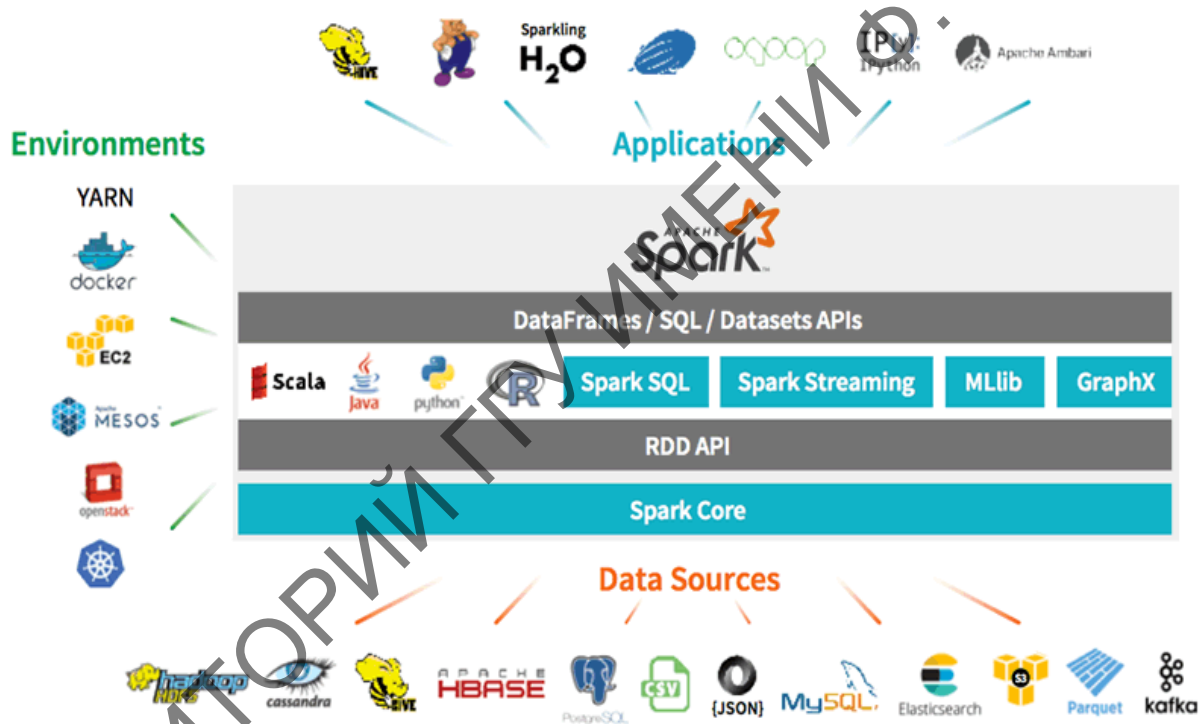


Рисунок 1 – Схема использования Apache Spark

Apache Spark имеет преимущества как инструмент для анализа и обработки данных из-за таких положительных качеств, как параллелизм, обеспечивающий распределение задач по разным узлам, тем самым увеличивая скорость работы приложения; возможности реализации на разных языках (Java, Scala, Python, R); возможности использования инструмента Spark Streaming для работы с данными в режиме реального времени. Последний прекрасно взаимодействует с брокером сообщений Apache Kafka и Elasticsearch и обеспечивает возможность получения данных при их изменении или дополнении в них. Очевидным плюсом фреймворка Spark является его простота в написании. Код Spark похож на работу с SQL, а именно: вызов определенных функций, вызываемых на нужном Dataframe.

В представленной системе обработки данных Apache Spark является основным инструментом анализа, фильтрации и группирования данных. Он принимает информацию в режиме реального времени из брокера сообщений Apache Kafka и выполняет на этих данных различные операции, такие как фильтрация, сортировка, выборка данных по критериям и другие. После всех преобразований Spark отправляет данные в Elasticsearch, где они отображаются с помощью инструмента визуализации данных Kibana.

Нативно Spark поддерживает Scala, Python и Java. Основным понятием в Spark'e является RDD (Resilient Distributed Dataset), который представляет собой Dataset. На нем можно делать преобразования двух типов: трансформация и действия.

Результатом применения операции трансформации RDD является новый RDD.

Достаточно популярным на сегодняшний день является еще один инструмент хранения и обработки данных Apache Kafka – распределённый программный брокер сообщений. Написан он на языках программирования Scala и Java. Спроектирован в виде распределённой, горизонтально масштабируемой системы, обеспечивающей наращивание пропускной способности как при росте нагрузки со стороны источников, так и при увеличении количества систем-подписчиков. Подписчики могут быть объединены в группы. Поддерживается возможность временного хранения данных для последующей пакетной обработки [2].

На рисунке 2 изображена схема работы инструмента Apache Kafka. Он состоит из поставщика сообщений или информации (Producer), которые записываются в определенный пакет (Topic), и потребителя, который читает или получает данную информацию из определенного пакета. Основным плюсом данного инструмента является то, что получатель, подписавшийся на определенный топик, получает информацию оперативно в режиме реального времени при изменении содержимого в топике [2].

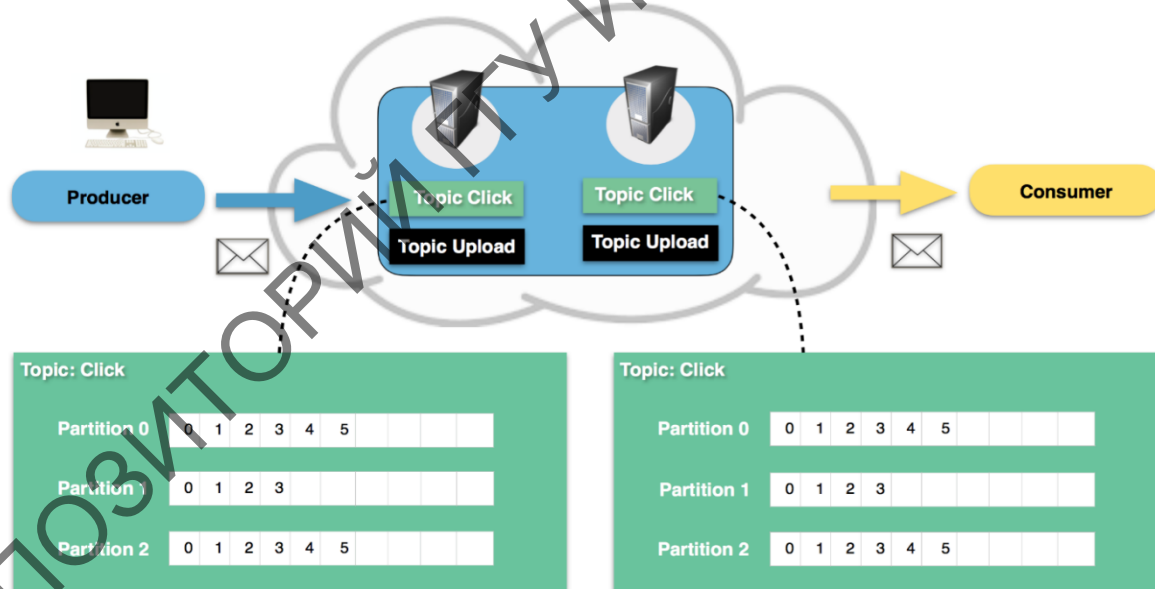


Рисунок 2 – Схема работы Apache Kafka

В разработанном приложении Apache Kafka служит инструментом для постоянного мониторинга обновленных данных, обеспечивает их хранение и отправку на следующий этап обработки – Spark Streaming. Apache Kafka – это консольное приложение, поэтому все управление ведется через консоль с использованием команд [2]. Кроме этого, существует возможность настройки UI управления, которое представлено на рисунке 3.

Данные записываются в Apache Kafka в определенный топик, который уже прослушивается пользователем Consumer'ом. Spark-приложение начинает работать при поступлении информации в топик. Реализуется функционал Streaming обработки и анализа данных: данные поступают и производится их трансформация.

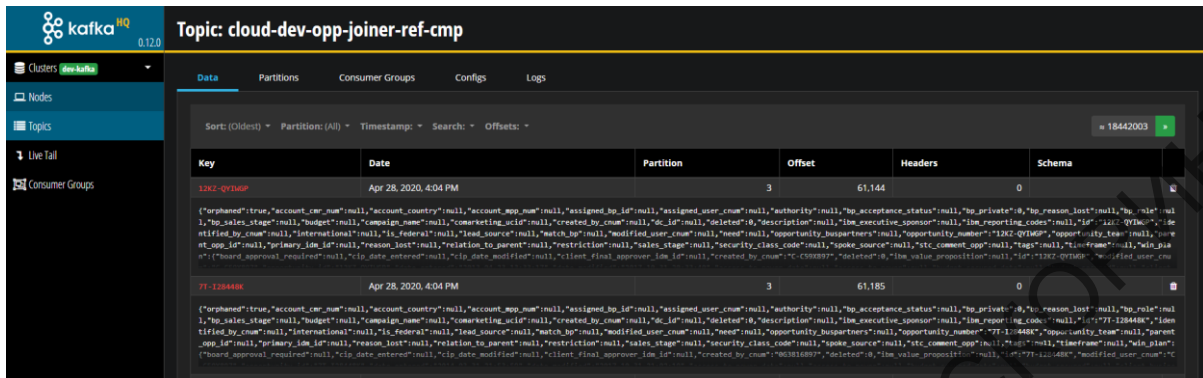


Рисунок 3 – Внешний вид инструмента управления Apache Kafka

В процессе работы системы анализа данные фиксируются, систематизируются, обрабатываются. Результаты анализа информации отображаются в виде графиков (рисунок 4). В качестве инструмента для хранения данных используется Elasticsearch – поисковая система для работы с JSON форматом данных (именно в этом формате и хранятся данные в Apache Kafka). Инструментом визуализации данных в виде различных графиков или диаграмм служит Kibana.



Рисунок 4 – Интерфейс Kibana с результирующими данными

В качестве языка для разработки приложения для имитации устройства для генерации данных был выбран язык программирования Java, который превратился из просто универсального языка в целую платформу и экосистему, объединяющую различные технологии, используемые в ряде задач: от создания десктопных приложений до написания крупных веб-порталов и сервисов, обеспечит доступность системы на обычных ПК, планшетах, смартфонах. Это гарантирует его популярность и востребованность как в среде новичков, так и программистов, изучающих новый вид программного обеспечения. В Java имеется поддержка работы с Apache Kafka, которая позволяет записывать данные в брокер в виде JSON файла.

Заключение. Таким образом, применение при разработке приложений современных технологий обработки больших объемов информации актуально, востребовано, ведет к созданию многофункциональной системы по обработке информации и является важной составляющей процесса создания архитектуры систем обработки данных.

Литература

1 Advanced Analytics with Spark: Patterns for Learning from Data at Scale / R. Sandy [et al.]. – O'Reilly Media, 2015. – 276 p.

2 Narkhede, N. Kafka: The Denitive Guide / N. Narkhede, G. Shapira, T. Palino. – O'Reilly Media, 2017. – 322 p.

УДК 333.71

И. С. Давыдов

ИССЛЕДОВАНИЕ ОДНОРОДНОСТИ РАЗВИТИЯ МАЛОГО И СРЕДНЕГО БИЗНЕСА В РАЗЛИЧНЫХ СТРАНАХ

Статья посвящена исследованию развития малого и среднего бизнеса в различных странах. Выделены методы и формы оказания поддержки малому и среднему предпринимательству. Проведена классификация стран по факторам макроэкономических показателей, оказывающих влияние на развитие малого предпринимательства. Группировка стран по уровню развития и проверка однородности развития малого и среднего предпринимательства может послужить основой для проведения дальнейших государственных мер по модернизации сферы малого и среднего бизнеса и сокращению различий.

Развитие малого и среднего бизнеса играет важную роль для экономики страны, так как способствует совершенствованию деятельности небольших предприятий в сфере социально-бытового обслуживания, розничной торговли, общественного питания. Появление таких предприятий способствует решению, в первую очередь, проблем занятости и увеличения доходов населения. Малый и средний бизнес также благоприятно влияет на повышение уровня и качества жизни населения [1].

В работе проводится анализ и классификация стран Евросоюза и стран Шанхайской организации сотрудничества (ШОС) по уровню развития и их государственной поддержки малого и среднего бизнеса. Информационной базой послужили статистические данные по следующим показателям: ВВП, ВВП на душу населения, инфляция, численность населения и уровень безработицы, ставка налога на прибыль, расходы на открытие бизнеса, индекс финансовой свободы, уровень контроля коррупции, конкурентоспособность различных стран с 2011 по 2017 годы [2–4].

Проведена классификация стран Европейского Союза с учетом макроэкономических показателей, влияющих на развитие малого и среднего бизнеса. Группировка стран проводилась на основе метрики Евклида, основанной на принципе полных связей для обнаружения однородных кластеров. Однородность групп проверяли дисперсионным анализом на уровне значимости 0,05 %. В результате в 2013 году по развитию малого и среднего бизнеса страны Европейского Союза сформировали 2 кластера. Первый кластер включал в себя Словению, Болгарию, Словакию, Чехию, Венгрию, Италию и Мальту. Во второй кластер вошли Великобритания, Австрия, Ирландия, Франция, Бельгия, Люксембург, Финляндия и Германия. Третий, обобщенный для двух предыдущих кластеров, включал Эстонию, Кипр, Швецию, Литву, Латвию, Грецию, Нидерланды,