УДК 004.42:004.65

ОБЗОР ПРОГРАММНОЙ ПЛАТФОРМЫ APACHE HADOOP ДЛЯ ОБРАБОТКИ И ХРАНЕНИЯ БОЛЬШИХ ДАННЫХ

Г.А. Пискун
Доцент кафедры проектирования информационно-компьютерных систем БГУИР, кандидат технических наук, доцент
piskunbsuir@gmail.com

В.Ф. Алексеев
Доцент кафедры проектирования информационно-компьютерных систем БГУИР, кандидат технических наук, доцент
alexvikt.minsk@gmail.com

Т.М. Воронко
Инженер-программист Центра информатизации и инновационных разработок БГУИР, магистрант БГУИР
voronko232001@gmail.com

Г.А. Пискун

Окончил Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов связана с моделирование и оптимальным проектированием информационно-компьютерных систем, организацией учебного и научно-исследовательского процессов в техническом университете.

В.Ф. Алексеев

Окончил Минский радиотехнический институт. Область научных интересов связана с разработкой методов и алгоритмов построения информационно-компьютерных систем, организацией учебного и научно-исследовательского процессов в техническом университете.

Т.М. Воронко

Магистрант 1 курса специальности «Электронные системы и технологии» кафедры проектирования информационно-компьютерных систем. Инженер-программист в отделе инновационных разработок в сфере образования Центра информатизации и инновационных разработок Белорусского государственного университета информатики и радиоэлектроники.

Аннотация. Выполнен обзор программной платформы для хранения и обработки больших данных на примере *Apache Hadoop*, которая представляет собой свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов. Рассмотрена концептуальная архитектура платформы через описание основных модулей, входящих в нее: *Hadoop Common*, *Hadoop Distributed File System*, *Hadoop YARN* и *Hadoop MapReduce.* Проанализирована область применения данной технологии. Установлено, что такие платформы для обработки и хранения больших данных как *Apache Hadoop*, являются одними из самых важных инструментов для работы с данными в современном мире, обеспечивая безопасность инфраструктуры и оптимизируя бизнес-процессы.

Ключевые слова: *BIg Data*, архитектура, инфраструктура.

Введение.

Платформа для работы с большими данными (*Big Data)* выполняет множество функций: «очищение» массива данных от лишней информации; обработка и структурирование массива данных; анализ массива данных; защита данных; обеспечение доступа ко всему объему постоянно изменяемых данных и так далее. Из всех вышеперечисленных функций приоритетное значение имеет анализ постоянно обновляемых данных. В современных условиях результаты такого анализа будут иметь решающее значение для компаний и предприятий при создании новых персонализированных товаров и услуг, позволят спрогнозировать дальнейшее направление развития [1].

В настоящее время существует множество проектов для работы с *big data*, которые отличаются разными моделями, структурой, особенностями анализа, разнообразными программными комплексами и техническими требованиями. Одной из самых распространенных и эффективных архитектур *big data* является *Apache Hadoop* (*Hadoop*) [1].

Несмотря на то, что платформа *Hadoop* старше и уступает в скорости во многих задачах своим аналогам, таким как *Apache Spark*, она все еще пользуется огромной популярностью среди разработчиков. Платформа надежна и хорошо протестирована, может быть развернута на большей части *UNIX*-подобных операционных систем, не будучи сильно требовательной к ресурсам. Благодаря предоставлению файловой системы и распределению рабочей нагрузки, использование платформы эффективно с финансовой точки зрения. Также *Hadoop* поддерживается многими облачными сервисами [2].

Рассмотрим более подробно концептуальную архитектуру платформы.

Концептуальная архитектура платформы *Hadoop*.

Hadoop – это основополагающая технология хранения и обработки больших данных, являющаяся проектом верхнего уровня фонда *Apache Software Foundation* [3].

Проект состоит из 4-х основных модулей [3]: *Hadoop Common*, *Hadoop Distributed File System*, *Hadoop YARN*, *Hadoop MapReduce*. Схематично концептуальная архитектура платформы *Hadoop* представлена на рисунке 1, где основные модули обведены красным цветом.

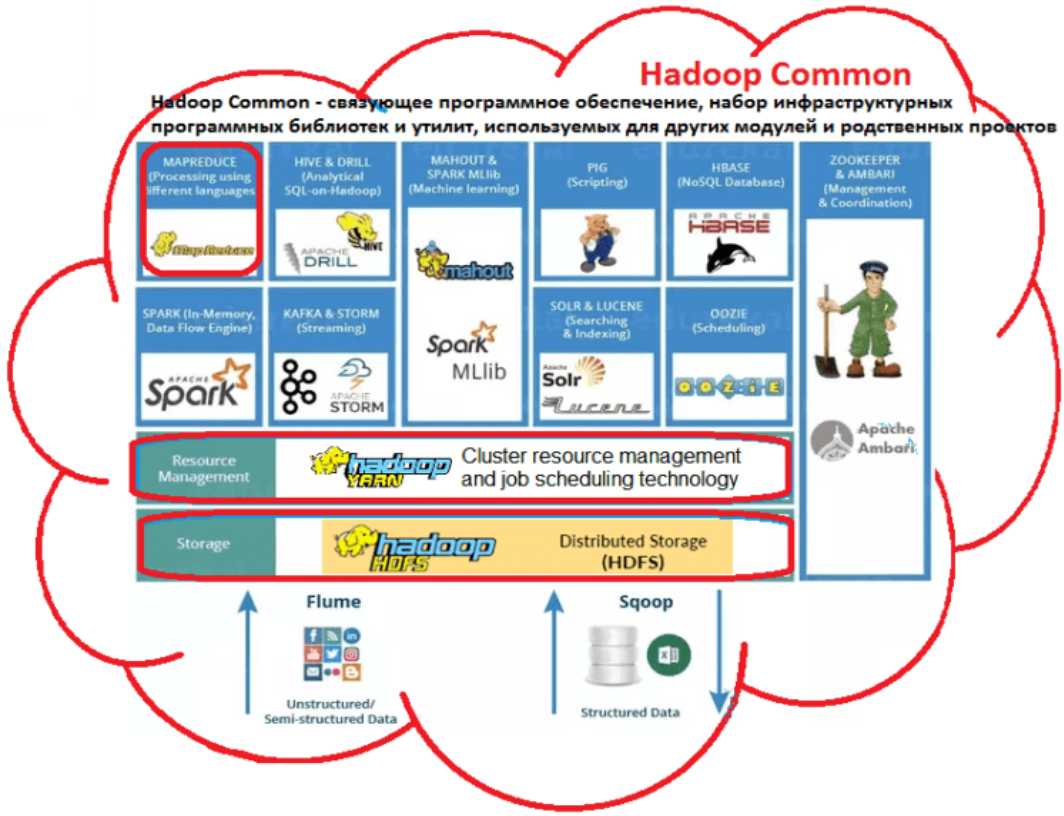

Рисунок 1. Концептуальная архитектура платформы *Hadoop*

Рассмотрим данные модули более подробно:

1. ***Hadoop Common*** – набор инфраструктурных программных библиотек и утилит, которые используются в других решениях и родственных проектах, в частности, для управления распределенными файлами и создания необходимой инфраструктуры [3].

В *Hadoop Common* входят библиотеки управления файловыми системами, поддерживаемыми *Hadoop*, и сценарии создания необходимой инфраструктуры и управления распределённой обработкой, для удобства выполнения которых создан специализированный упрощённый интерпретатор командной строки (*FS shell*, *filesystem shell*), запускаемый из оболочки операционной системы командой вида: *hdfs dfs -command URI*, где *command* – команда интерпретатора, а *URI* – список ресурсов с префиксами, указывающими тип поддерживаемой файловой системы, например *hdfs://example.com/file1* или *file:///tmp/local/file2*. Большая часть команд интерпретатора реализована по аналогии с соответствующими командами *Unix* (например, *cat*, *chmod*, *chown*, *chgrp*, *cp*, *du*, *ls*, *mkdir*, *mv*, *rm*, *tail*). При этом поддерживаются некоторые ключи аналогичных *Unix*-команд, например, ключ рекурсивности *-R* для *chmod*, *chown*, *chgrp*), есть команды специфические для *Hadoop* (например, *count* подсчитывает количество каталогов, файлов и байтов по заданному пути, *expunge* очищает корзину, а *setrep* модифицирует коэффициент репликации для заданного ресурса) [4].

2. ***Hadoop Distributed File System*** (***HDFS***) – распределенная файловая система *Hadoop* для хранения файлов больших размеров с возможностью потокового доступа к информации, поблочно распределённой по узлам вычислительного кластера, который может состоять из произвольного аппаратного обеспечения. *HDFS*, как и любая файловая система – это иерархия каталогов с вложенными в них подкаталогами и файлами [5].

На рисунке 2 отображено схематическое представление *HDFS* [6].

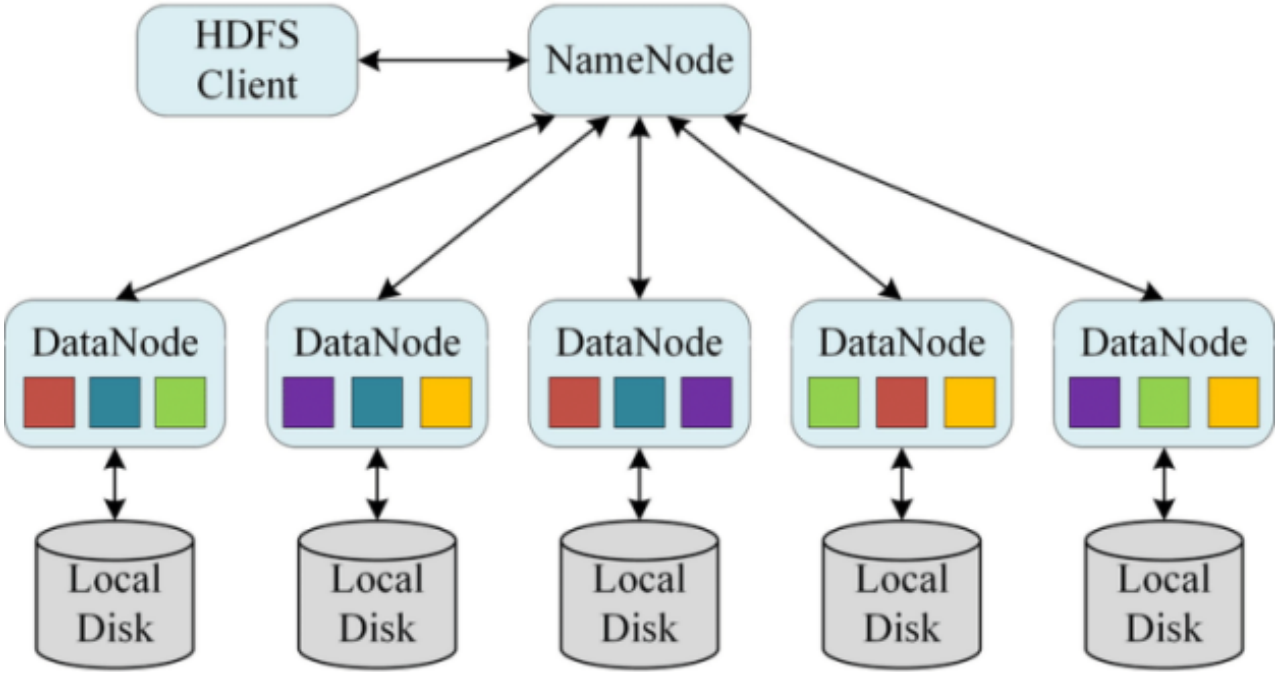

Рисунок 2. Схематическое представление *HDFS*

Кластер *HDFS* включает следующие компоненты [5]:

– управляющий узел, узел имен или сервер имен (*NameNode*) – отдельный, единственный в кластере, сервер с программным кодом для управления пространством имен файловой системы, хранящий дерево файлов, а также метаданные файлов и каталогов. *NameNode* – обязательный компонент кластера *HDFS*, который отвечает за открытие и закрытие файлов, создание и удаление каталогов, управление доступом со стороны внешних клиентов и соответствие между файлами и блоками, дублированными (реплицированными) на узлах данных. Сервер имён раскрывает для всех желающих расположение блоков данных на машинах кластера;

– *secondary NameNode* – вторичный узел имен, отдельный сервер, единственный в кластере, который копирует образ *HDFS* и лог транзакций операций с файловыми блоками во временную папку, применяет изменения, накопленные в логе транзакций к образу *HDFS*, а также записывает его на узел *NameNode* и очищает лог транзакций. *Secondary NameNode* необходим для быстрого ручного восстанавления *NameNode* в случае его выхода из строя;

– узел или сервер данных (*DataNode*, *Node*) – один их множества серверов кластера с программным кодом, отвечающим за файловые операции и работу с блоками данных. *DataNode* – обязательный компонент кластера *HDFS*, который отвечает за запись и чтение данных, выполнение команд от узла *NameNode* по созданию, удалению и репликации блоков, а также периодическую отправку сообщения о состоянии (*heartbeats*) и обработку запросов на чтение и запись, поступающих от клиентов файловой системы *HDFS*. Стоит отметить, что данные проходят с остальных узлов кластера к клиенту мимо узла *NameNode*;

– клиент (*client*) – пользователь или приложение, взаимодействующий через специальный интерфейс (*API – Application Programming Interface*) с распределенной файловой системой. При наличии достаточных прав, клиенту разрешены следующие операции с файлами и каталогами: создание, удаление, чтение, запись, переименование и перемещение. Создавая файл, клиент может явно указать размер блока файла (по умолчанию 64 Мб) и количество создаваемых реплик (по умолчанию значение равно 3-ем).

Взаимодействие узлов имен, узлов данных и клиентов осуществляется по протоколам на основе *TCP/IP*.

3. ***Hadoop YARN*** – система планирования заданий и управления кластером (*Yet Another Resource Negotiator*), которую также называют *MapReduce* 2.0 (*MRv2*) – набор системных программ (демонов), обеспечивающих совместное использование, масштабирование и надежность работы распределенных приложений.

Фактически, *YARN* является интерфейсом между аппаратными ресурсами кластера и приложениями, использующих его мощности для вычислений и обработки данных [3].

Архитектура и принцип работы *Hadoop YARN* представлены на рисунке 3 [7].

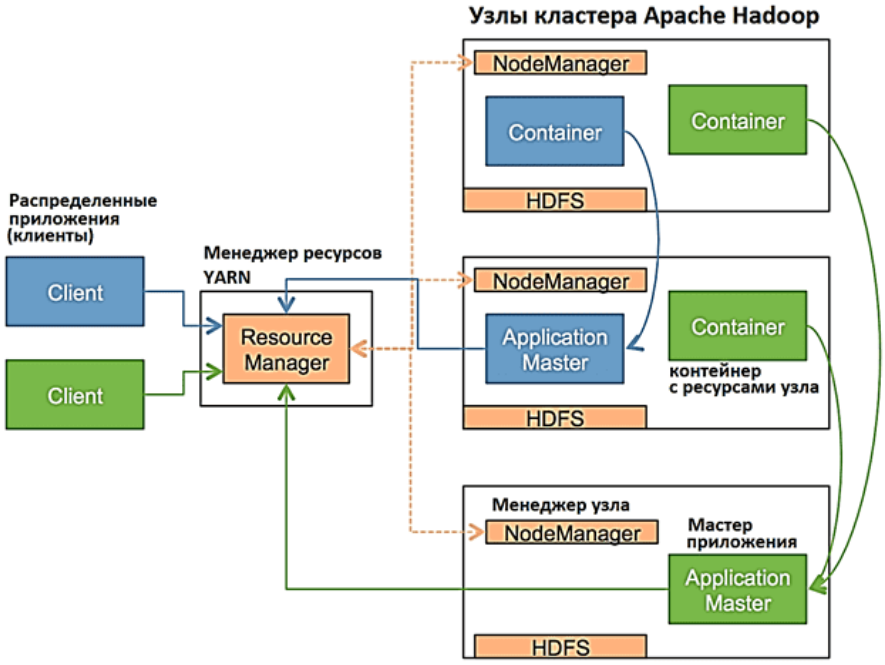

Рисунок 3. Архитектура и принцип работы *Hadoop YARN*

YARN состоит из следующих компонентов [7]:

– *ResourceManager* (*RM*) – менеджер ресурсов, которых отвечает за распределение ресурсов, необходимых для работы распределенных приложений, и наблюдение за узлами кластера, где эти приложения выполняются. *ResourceManager* включает планировщик ресурсов (*Scheduler*) и диспетчер приложений (*ApplicationsManager*, *AsM*);

– *ApplicationMaster* (*AM*) – мастер приложения, ответственный за планирование его жизненного цикла, координацию и отслеживание статуса выполнения, включая динамическое масштабирование потребления ресурсов, управление потоком выполнения, обработку *ошибок* и искажений вычислений, выполнение локальных оптимизаций. Каждое приложение имеет свой экземпляр *ApplicationMaster*. *ApplicationMaster* выполняет произвольный пользовательский код и может быть написан на любом языке программирования благодаря расширяемым протоколам связи с менеджером ресурсов и менеджером узлов;

– *NodeManager* (*NM*) – менеджер узла – агент, запущенный на узле кластера, который отвечает за отслеживание используемых вычислительных ресурсов (*CPU*, *RAM* и пр.), управление логами и отправку отчетов об использовании ресурсов планировщику. *NodeManager* управляет абстрактными контейнерами – ресурсами узла, доступными для конкретного приложения;

– Контейнер (*Container*) – набор физических ресурсов (ЦП, память, диск, сеть) в одном вычислительном узле кластера.

4. ***Hadoop MapReduce*** – платформа программирования и выполнения распределённых *MapReduce*-вычислений, с использованием большого количества компьютеров (узлов, *nodes*), образующих кластер [3].

Суть *MapReduce* состоит в разделении информационного массива на части, параллельной обработке каждой части на отдельном узле и финального объединения всех результатов [8].

Принцип работы *MapReduce* представлен на рисунке 4 [8].

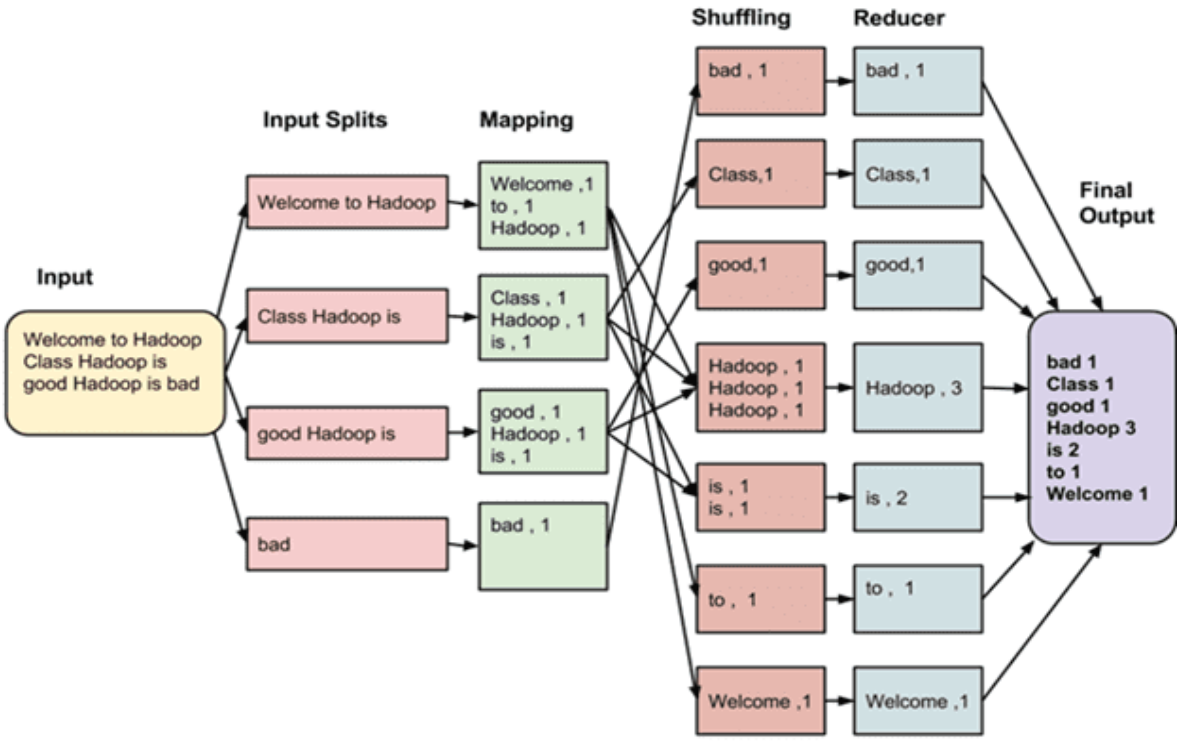

Рисунок 4. Принцип работы *MapReduce*

Вычислительная модель состоит из 3-хшаговой комбинации следующих функций [8]:

– *Map* – предварительная обработка входных данных в виде большого списка значений. При этом главный узел кластера (*master node*) получает этот список, делит его на части и передает рабочим узлам (*worker node*). Далее каждый рабочий узел применяет функцию *Map* к локальным данным и записывает результат в формате «ключ-значение» во временное хранилище.

– *Shuffle* – перераспределение рабочими узлами данных на основе ключей, ранее созданных функцией *Map*, таким образом, чтобы все данные одного ключа лежали на одном рабочем узле.

– *Reduce* – параллельная обработка каждым рабочим узлом каждой группы данных по порядку следования ключей и «склейка» результатов на *master node*. Главный узел получает промежуточные ответы от рабочих узлов и передаёт их на свободные узлы для выполнения следующего шага. Получившийся после прохождения всех необходимых шагов результат – это и есть решение исходной задачи.

Область применения *Hadoop*.

Платформа активно используется для регулирования рисков и безопасности инфраструктуры, кроме этого, она необходима для оптимизации бизнес-процессов, проведения финансового анализа, выполнения маркетингового анализа и изучения неструктурированной информации, которая собиралась во время продаж [9].

Несистематизированную информацию, собранную из различных источников, нередко называют «озером данных». Как правило, компания не нуждается в таких сведениях, но обязана хранить их по закону. Некоторые организации после завершения анализа данных находят им применение для будущих проектов или задач [9].

При хранении информации в разных источниках и форматах ее анализ, моделирование и прогнозирование затруднено. По сути, «озеро данных» является ненужным для компании, так как не может принести никакой практической пользы. С помощью данной технологии становится возможным распределить и классифицировать сведения, а затем – провести их аналитику и получить различные результаты [9].

Hadoop также часто применяется в следующих целях [9]:

1. *Изучение данных из социальных сетей*. Как правило, сведения из соцсетей позволяют организации лучше понять потребности аудитории. С помощью *Hadoop* анализируются интересы, уровень доходов, уровень образования. Такой подход позволяет настроить таргетированную рекламу, управлять репутацией бренда и повышать отклик из групп и аккаунтов.

2. *Анализ отношения к компании*. Если говорить о том, для чего используется *Hadoop*, то нельзя не упомянуть этот параметр. Технология позволяет анализировать мнения клиентов, которые они высказывают в соцсетях, блогах, отзывах. Удается проанализировать отношение покупателей к отдельным продуктам / услугам и всему бренду в целом. Это помогает спрогнозировать покупки, скорректировать маркетинговое продвижение товара или оценить имеющуюся репутацию на рынке.

3. *Поддержание безопасности инфраструктуры*. Решение позволяет отслеживать серверные журналы и выявлять любые нарушения безопасности. С помощью технологии можно выявить возможные риски, обнаружить сетевые атаки и спрогнозировать возможные проблемы.

4. *Исследование геоданных*. Компании в области ритейла и производства нередко (с согласия клиентов) собирают сведения об их местоположении. Полученные сведения позволяют в будущем спрогнозировать визиты пользователей или подобрать товары с учетом геолокации. С помощью платформы осуществляет оптимизация и обработка таких геоданных, что упрощает всю процедуру и сокращает время на ее выполнение.

5. *Сбор сведений о поведении клиентов*. Нередко технология оказывается полезной при сборе и обработке данных о поведении и вовлеченности пользователей на сайте. Например, платформа помогает обработать информацию о том, откуда пользователи перешли на сайт, на какую страницу попали, сколько времени провели за просмотром контента, с каких страниц чаще

всего уходили. Анализ подобных данных позволяет организации повысить конверсию ресурса и сделать его более удобным с точки зрения пользователей.

Заключение.

Сегодня *Hadoop* можно встретить в самых разных отраслях – от производства до госсектора. Финансовые и транспортные компании, частные клиники и многие другие организации активно используют данную платформу для управления, анализа и повышения эффективности рабочего процесса. В мире, где огромная часть информации хранится в цифровом виде, а объемы данных становятся все больше, такие платформы, как *Hadoop*, являются просто незаменимыми.

Список литературы

[1] Big Data: технология, принципы и архитектура [Электронный ресурс]. URL: http://ipcmagazine.ru/legal-issues/big-data-technology-principles-and-architecture (Дата обращения: 28.03.2023).

[2] Top Big Data frameworks: what will tech companies choose in 2023? [Электронный ресурс]. URL: https://jelvix.com/blog/top-5-big-data-frameworks (Дата обращения: 28.03.2023).

[3] Что такое Hadoop, где и зачем используется - хадуп для чайников [Электронный ресурс]. URL: https://www.bigdataschool.ru/wiki/hadoop (Дата обращения: 28.03.2023).

[4] Hadoop [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Hadoop (Дата обращения: 28.03.2023).

[5] Что такое HDFS: все о Hadoop Distributed File System [Электронный ресурс]. URL: https://www.bigdataschool.ru/wiki/hdfs (Дата обращения: 28.03.2023).

[6] The overview of the Hadoop Distributed File System HDFS [Электронный ресурс]. URL: https://www.researchgate.net/figure/The-overview-of-the-Hadoop-Distributed-File-System-HDFS_fig4_348387085 (Дата обращения: 28.03.2023).

[7] Что такое YARN: архитектура, принцип работы, роль в Apache Hadoop [Электронный ресурс]. URL: https://www.bigdataschool.ru/wiki/yarn (Дата обращения: 28.03.2023).

[8] MapReduce Architecture [Электронный ресурс]. URL: https://www.geeksforgeeks.org/mapreduce-architecture/ (Дата обращения: 28.03.2023).

[9] Hadoop, где и зачем используется [Электронный ресурс]. URL: https://www.xelent.ru/blog/chto-takoe-hadoop-gde-i-zachem-ispolzuetsya/ (Дата обращения: 28.03.2023).

REVIEW OF THE SOFTWARE PLATFORM FOR PROCESSING AND STORING BIG DATA ON THE EXAMPLE OF APACHE HADOOP

G.A. Piskun
Associate Professor, Department of Information Computer Systems Design, PhD of Technical sciences, Associate Professor

V.F. Alekseev
Associate Professor, Department of Information Computer Systems Design, PhD of Technical sciences, Associate Professor

T.M. Voronko
Software Engineer of the Center of Informatization and Innovative Developments of BSUIR, master student of BSUIR

Faculty of Computer Engineering
Belarusian State University of computer science and Radio Electronics, Republic of Belarus
E-mail: voronko232001@gmail.com

Abstract. A review of the software platform for storing and processing big data is made using the example of Apache Hadoop, as one of the most common and effective today. This is a freely distributed set of utilities, libraries and framework for developing and executing distributed programs running on clusters of hundreds and thousands of nodes. The conceptual architecture of the platform is considered through the description of the main modules included in it: Hadoop Common, Hadoop Distributed File System, Hadoop YARN, Hadoop MapReduce, and the scope of this technology is analyzed.

As a result of the analysis, it was found that such platforms for processing and storing big data, such as Apache Hadoop, are one of the most important tools for working with data in the modern world, ensuring infrastructure security and optimizing business processes.

Keywords: big data, architecture, infrastructure.