УДК 004.42

# ПОТЕНЦИАЛ APACHE HADOOP В ОБЛАСТИ ОБРАБОТКИ И АНАЛИЗА БОЛЬШИХ ДАННЫХ

С.Э. ЛАРИН, В.Ю. БЕЛАШ

*ФГБОУ ВО «Калужский государственный университет имени К.Э. Циолковского», г. Калуга*

*Ключевые слова и фразы: Hadoop*; *big data*; анализ данных; машинное обучение; информационные технологии; распределенная обработка данных.

*Аннотация:* Цель проведенного исследования заключается в изучении технологии *Apache Hadoop*, используемой для обработки и анализа больших данных. Гипотеза исследования состоит в повышении эффективности обработки больших данных с применением рассматриваемой технологии. Методы исследования – анализ учебной и научной литературы, а также моделирование на языке *Python*. Достигнутые результаты: обозначена область применения платформы, выделены ее преимущества, изучены перспективы развития, проведена демонстрация алгоритмов работы.

Беспрерывный рост объемов информации, генерируемой в различных сферах, ставит перед нами новые задачи. Обработка и анализ этих данных становятся ключом к прогрессу в науке, бизнесе и многих других областях.

*Apache Hadoop* – это фреймворк с открытым исходным кодом, который оказывает новые горизонты в работе больших данных. Он позволяет эффективно распределять задачи обработки и хранения информации на кластерах компьютеров.

Основополагающими компонентами *Apache Hadoop* являются:

1. *Hadoop Distributed File System* (***HDFS***) – распределенная файловая система, обеспечивающая надежное хранение больших объемов данных;
2. *Apache MapReduce* – модель программирования, реализующая параллельную обработку данных на множестве компьютеров.

Перейдем к подробному изучению каждого компонента.

На рис. 1 представлена архитектура *HDFS*. Она включает в себя четыре основных компонента: клиент, главный узел, вторичный главный узел и узлы данных [2]. Рассмотрим более детально логику работы архитектуры *Hadoop Distributed File System*.

Например, клиент запрашивает операцию над файлом – запись данных. Главный узел получает запрос от клиента и определяет, на каких узлах хранятся блоки данных. Главный узел сообщает клиенту адреса этого узла. Клиент напрямую взаимодействует с узлом данных, чтобы выполнить операцию, и этот узел реплицирует данные в соответствии с настройками *HDFS*.

Несмотря на свою универсальность, *HDFS* имеет два конкретных ограничения.

1. Неэффективность при работе с небольшими файлами, поскольку при хранении большого количества маленьких файлов накладные расходы на метаданные могут стать значительнее, что негативно повлияет на производительность.
2. Несоответствие приложениям с низкой задержкой. В приложениях, где требуется высокая скорость доступа к данным, например, интерактивных системах, *HDFS* будет показывать неудовлетворительную производительность.

На рис. 2 представлена архитектура *MapReduce*. В нее входят следующие компоненты: *Input*, *Map*, *Shuffle*, *Reduce*, *Output*. Компонент *Input* отвечает за загрузку данных из различных источников, таких как *HDFS*, базы данных и т.д. *Map* преобразует входные данные в промежуточные и применяется к каждому блоку и выходные данные представляют собой

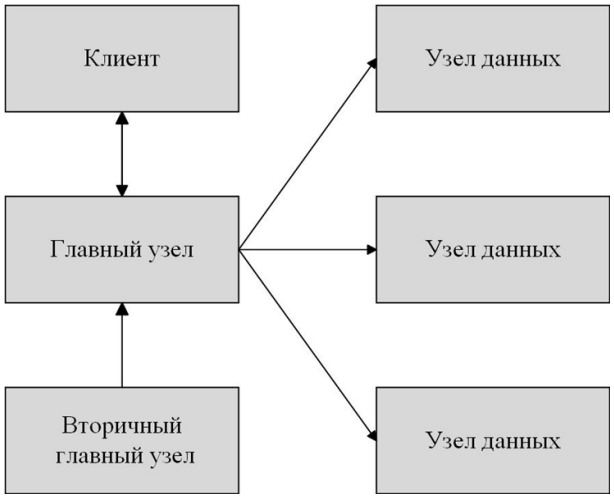


**Рис. 1.** Архитектура *HDFS*

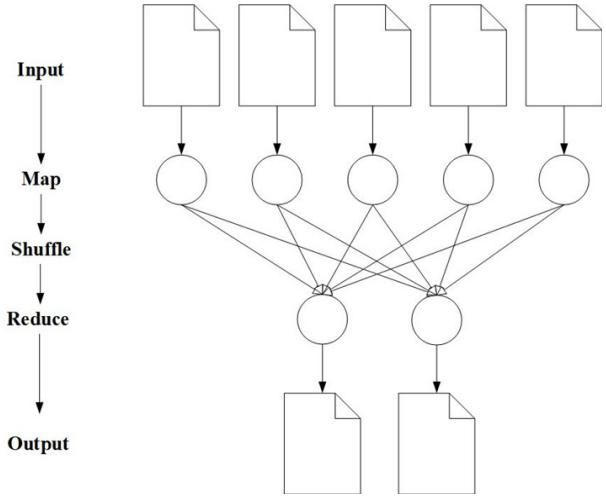


**Рис. 2.** Архитектура *MapReduce*

ключ-значение. *Shuffle* отвечает за перемешивание и сортировку промежуточных данных. Данные с одинаковыми ключами объединяются. *Reduce* объединяет промежуточные данные в итоговый набор данных. *Output* сохраняет итоговые данные в *HDFS* или другие системы хранения.

Помимо *HDFS* и *MapReduce*, платформа *Hadoop* включает в себя ряд других инструментов.

*Apache Flume*: инструмент для сбора, агрегации и перемещения больших потоков данных, который не является частью *Hadoop*, но взаимодействует с ним. Служит мостом между различными источниками данных (социальные сети, журналы серверов, датчики) и *Hadoop*. Обеспечивает реальную обработку данных и масштабируемость.

*Apache Mahout* – это программная библиотека машинного обучения, которая имеет открытый исходный код, может эффективно предоставить пользователям такие аналитические возможности, как кластеризация, анализ данных и т.д., на распределенном кластере *Hadoop*. *Mahout* очень эффективен при работе с большими массивами данных. Алгоритмы, предоставляемые *Mahout*, оптимизированы для работы с фреймворком *MapReduce* на *HDFS*.

*Apache Pig* представляет собой слой абстракции поверх *MapReduce*. Он является языком программирования, который позволяет создавать программы *MapReduce* с помощью *Pig*. *Pig Latin* – это язык высокого уровня, на котором разработчики могут писать высокоуровневое программное обеспечение для анализа данных. *Pig* генерирует задачи параллельного выполнения, поэтому эффективно использует распределенный кластер *Hadoop*. Изначально *Pig* был разработан в *Yahoo*! *Research*, чтобы позволить разработчикам создавать специальные задания *MapReduce* для *Hadoop*. С тех пор *Apache Pig* начали использовать многие крупные организации, такие как *eBay*, *LinkedIn* и *Twitter*.

*Apache HBase* – это распределенная база данных с произвольным доступом и ориентированная на столбцы. *HBase* работает непосредственно поверх *HDFS* и позволяет разработчикам приложений напрямую читать/записывать данные *HDFS*. *HBase* не поддерживает *SQL*, поэтому ее также называют базой данных *NOSQL*. Однако она предоставляет интерфейс на основе командной строки, а также богатый набор *API* для обновления данных. Данные в *HBase* хранятся в *HDFS* в виде пар ключ-значение.

*Apache Hadoop* – доминирующая платформа в сфере анализа больших данных. В последние годы она получила широкое распространение и применяется в различных областях, например, имеет возможность анализа данных из любого рода источников, таких как журналы серверов, социальные сети, данные о транзакциях и так далее [1]. С помощью *Hadoop* можно выполнять поиск, агрегацию, классификацию и машинный перевод текста. Также платформа предоставляет инструменты для обучения моделей на массивах данных для задач прогнозирования, классификации и регрессии в рамках

машинного обучения.

*Apache Hive* реализует возможности хранения данных с использованием *Big Data*. *Hive* работает поверх *Apache Hadoop* и использует *HDFS* для хранения данных. С *Hive* разработчики вообще не пишут *MapReduce*. *Hive* предоставляет разработчикам приложений *SQL*-подобный язык запросов под названием *HiveQL*, позволяющий быстро писать специальные запросы, аналогичные *SQL*-запросам в СУБД.

*Apache HCatalog* предоставляет услуги управления метаданными поверх *Apache Hadoop*. Это означает, что все программы, работающие на *Hadoop*, могут эффективно использовать *HCatalog* для хранения своих схем в *HDFS*. *HCatalog* помогает любому стороннему программному обеспечению создавать, редактировать и представлять (используя остальные *API*) сгенерированные метаданные или определения таблиц.

Таким образом, любой пользователь или скрипт может эффективно работать с *Hadoop*, не зная, где на самом деле данные физически хранятся на *HDFS*. *HCatalog* предоставляет команды *DDL* (Язык определения данных), с помощью которого запрашиваемые задания *MapReduce*, *Pig* и *Hive* могут быть поставлены в очередь на выполнение и впоследствии отслеживаться по мере необходимости.

*Apache Hadoop* обладает рядом преимуществ: масштабируемость, отказоустойчивость и возможность обработки различных типов данных. Однако среди ограничений можно отметить сложность конфигурации и управления кластером, а также относительно низкую производительность для некоторых типов задач.

*Apache Hadoop* является мощной платформой, которая доминирует в сфере анализа больших данных. Обладает достаточным количеством преимуществ, а также несмотря на свои ограничения, остается одним из популярных решений для предприятий и организаций, которые сталкиваются с потребностью в обработке больших данных.

В будущем *Hadoop* продолжит развиваться, и его возможности расширятся. Ожидается, что *Hadoop* будет играть все более важную роль в различных областях, таких как медицина, финансы, анализ социальных сетей.

**Литература**

1. HDFS Architecture [Electronic resource]. – Access mode : https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html.

2. HDFS для новичков // HDFS: как хранятся большие данные [Электронный ресурс]. – Режим доступа : https://skillbox.ru/media/code/hdfs-kak-khranyatsya-bolshie-dannye.

3. Sammer, E. Hadoop Operations / E. Sammer, 2012. – 297 p.

4. Карамбелкар, Х. Масштабирование больших данных с помощью Hadoop и Solr / Х. Карамбелкар. – М. : Додэка-XXI, 2013. – 320 с.

5. Лаврентьев, Д.О. Разработка клиент-серверного кроссплатформенного приложения с использованием современных технологий / Д.О. Лаврентьев, В.Ю. Белаш // Наука и бизнес: пути развития. – М. : ТМБпринт. – 2023. – № 3(141). – С. 35–38.

**References**

2. HDFS dlya novichkov // HDFS: kak khranyatsya bolshie dannye [Electronic resource]. – Access mode : https://skillbox.ru/media/code/hdfs-kak-khranyatsya-bolshie-dannye.

4. Karambelkar, KH. Masshtabirovanie bolshikh dannykh s pomoshchyu Hadoop i Solr / KH. Karambelkar. – M. : Dodeka-XXI, 2013. – 320 s.

5. Lavrentev, D.O. Razrabotka klient-servernogo krossplatformennogo prilozheniya s ispolzovaniem sovremennykh tekhnologij / D.O. Lavrentev, V.YU. Belash // Nauka i biznes: puti razvitiya. – M. : TMBprint. – 2023. – № 3(141). – S. 35–38.