СОВРЕМЕННАЯ НАУКА: ТРАДИЦИИ И ИННОВАЦИИ

Сборник научных статей
по итогам II молодежного конкурса научных работ

НАУЧНЫЙ ИЗДАТЕЛЬСКИЙ ЦЕНТР «АБСОЛЮТ»

СОВРЕМЕННАЯ НАУКА: ТРАДИЦИИ И ИННОВАЦИИ

Сборник научных статей
по итогам II молодежного конкурса
научных работ

Волгоград 2020

УДК 62; 33; 1; 81; 34; 37; 61; 008; 55
ББК 3; 65; 87; 81; 67; 74; 51; 71; 26
С56

16+

Печатается по решению редакционно-издательского совета научного издательского центра «Абсолют».

Электронная версия размещается
в Научной электронной библиотеке (eLibrary.ru)
(лицензионный договор с ООО «НЭБ» № 221-02/2018К)

Председатель конкурсной комиссии:
А. В. Бабаян, д-р пед. наук, проф.,
Пятигорский государственный университет.

С56 Современная наука: традиции и инновации [Текст] : сборник научных статей по итогам II молодежного конкурса научных работ. – Волгоград: НИЦ «Абсолют», 2020. – 96 с.

ISBN 978-5-6041032-4-9

В сборнике рассматриваются теоретические и практические вопросы современной науки. В издание включены тексты научных статей, представленных на II молодежный конкурс научных работ «Современная наука: традиции и инновации».

Для преподавателей, аспирантов, магистрантов и студентов, а также всех интересующихся вопросами развития современного научного знания.

УДК 62; 33; 1; 81; 34; 37; 61; 008; 55
ББК 3; 65; 87; 81; 67; 74; 51; 71; 26

КОНКУРСНАЯ КОМИССИЯ

Председатель:

А. В. Бабаян, д-р пед. наук, проф., Пятигорский государственный университет.

Члены комиссии:

О. Е. Ваганова, канд. экон. наук, доц., Российский экономический университет имени Г.В. Плеханова;

Н. Н. Давидчук, доц., Донецкий национальный университет экономики и торговли имени Михаила Туган-Барановского;

О. А. Фомкина, д-р мед. наук, Саратовский ГМУ им. В.И. Разумовского;

Е. А. Кириллова, канд. юрид. наук, Юго-Западный государственный университет;

Л. В. Мальцева, д-р пед. наук, проф., Кубанский государственный университет;

А. Э. Кенжабаев, канд. пед. наук, доцент, Термезский государственный университет

ПОБЕДИТЕЛЬ В НОМИНАЦИИ
«ЛУЧШЕЕ ПРАКТИЧЕСКОЕ ИССЛЕДОВАНИЕ»

Нотфуллина Р.Р., Гаврилов А.С.
ЭНЕРГОСБЕРЕЖЕНИЕ В СИСТЕМАХ ВОЗДУХОСНАБЖЕНИЯ ПРОМЫШЛЕННЫХ ПРЕДПРИЯТИЙ

ПОБЕДИТЕЛЬ В НОМИНАЦИИ
«ЛУЧШЕЕ ОБЗОРНОЕ ИССЛЕДОВАНИЕ»

Трепов Е.С.
УСТРОЙСТВА, ИСПОЛЬЗУЕМЫЕ ПРИ РАССЛЕДОВАНИИ ПРЕСТУПЛЕНИЙ В СФЕРЕ КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ

ПОБЕДИТЕЛЬ В НОМИНАЦИИ
«ЛУЧШИЙ ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ»

Ходжатов К.Б.
АВТОРИТАРНЫЙ РЕЖИМ НА ПОСТСОВЕТСКОМ ПРОСТРАНСТВЕ:
ЕСТЬ ЛИ АЛЬТЕРНАТИВА?

ПОБЕДИТЕЛЬ В НОМИНАЦИИ
«ЛУЧШИЙ ЭМПИРИЧЕСКИЙ МАТЕРИАЛ»

Лукьянченко А.Ю.
КЛИНИКО-РЕНТГЕНОЛОГИЧЕСКИЕ ОСОБЕННОСТИ
ТЕЧЕНИЯ ВНЕБОЛЬНИЧНОЙ ПНЕВМОНИИ У БОЛЬНЫХ
ОБУЗ «КУРСКАЯ ГОРОДСКАЯ БОЛЬНИЦА № 6» ЗА 2019 ГОД .....

ПОБЕДИТЕЛЬ В НОМИНАЦИИ
«РАЗВЕРНУТОСТЬ ИССЛЕДОВАНИЯ»

Кортелева А.В., Ильинская П.В., Савонова С. М.
КОНФУЦИАНСТВО В ПОЛИТИКЕ СОВРЕМЕННОГО КИТАЯ

ПОБЕДИТЕЛЬ В НОМИНАЦИИ
«НОВАТОРСТВО В НАУКЕ»

Галенина А.А.
ТЕХНОЛОГИИ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

ТЕХНИЧЕСКИЕ НАУКИ

УДК 004.42

Галенина А.А. ©
студент кафедры программного обеспечения и администрирования информационных систем
Факультет физики математики и информатики

Курский государственный университет
г. Курск

ТЕХНОЛОГИИ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

«Большие данные» – термин, который слышал практически каждый в современном мире. Но что же конкретно подразумевается под этим термином? Как осуществляется обработка больших данных? Данная статья даст ответы на эти вопросы. Приводится четкое определение термина, история возникновения и основные характеристики больших данных. В данной статье также рассмотрена модель программирования MapReduce и фреймворки, используемые для аналитики больших данных. Приведено описание и примеры нереляционных баз данных. Рассмотрено применение искусственного интеллекта для эффективной обработки больших данных, а также интеграция больших данных с блокчейном.

Ключевые слова: *анализ неструктурированных данных, большие данные, характеристики больших данных, технология обработки больших данных, модель программирования MapReduce, фреймворк Hadoop, фреймворк Apache Spark, нереляционные базы данных NoSQL, резидентная база данных, искусственный интеллект, машинное обучение, блокчейн.*

Big Data бесспорно является одним из самых популярных терминов в современном мире. Количество собранных данных продолжает расти. Поэтому анализ и обработка больших данных являются одной из наиболее важных задач, над которыми работают исследователи, чтобы найти наилучшие подходы для их обработки с высокой производительностью, низкой стоимостью и высокой точностью.

Большие данные – это термин, используемый для сбора больших и сложных наборов данных, которые трудно хранить и обрабатывать с использованием доступных инструментов управления базами данных или традиционных прило-

жений для обработки данных. Большие данные определяются на основе «3V» [6, с. 18]: volume (огромные объемы данных), variety (разнообразие форм и структур) и velocity (скорость обработки данных). Данные определения ввел аналитик компании Gartner Дуг Лейни (Doug Laney) в 2001 году.

Корни обработки технологий больших данных уходят в 2002 год, когда Дуг Каттинг (Doug Cutting) работал над проектом с открытым исходным кодом Nutch, целью которого была индексация веб-страниц и использование уже проиндексированных страниц для поиска. При работе он столкнулся с проблемами масштабируемости как с точки зрения хранения, так и вычислений. В 2003 году Google опубликовал GFS (файловая система Google), а в 2004 году Nutch создал NDFS (распределенную файловую систему Nutch). В этом же году компания Google разработала свою парадигму программирования MapReduce. Авторами данной модели являются Джеффри Дин (Jeffrey Dean) и Санджай Гемават (Sanjay Ghemawat). В 2005 году Dough смог запустить Nutch на NDFS и использовал MapReduce. Это послужило зарождением Hadoop – платформы с открытым исходным кодом для хранения и обработки больших массивов данных. В январе 2006 года Hadoop был выделен в отдельный проект.

Таким образом, технология обработки больших данных началась с платформы Hadoop и парадигмы программирования MapReduce.

MapReduce – программная модель, предназначенная для распределенных вычислений. Реализован MapReduce преимущественно на языке Java. Алгоритм MapReduce включает в себя три стадии [1, с.85]:

1. Стадия Map. На данной стадии используется функция map(), которая берет заданный набор данных и преобразует его в другой набор данных, где отдельные элементы разбиваются на кортежи (пары «ключ/значение»).

2. Стадия Shuffle. На данной стадии происходит передача данных из Mapper в Reducer. Данная стадия необходима для Reducer, иначе будут отсутствовать входные данные. Эта стадия может начаться еще до завершения предыдущей, что существенно экономит время. Также происходит сортировка всех промежуточных пар «ключ-значение» по ключу. Таким образом, на одном рабочем узле лежат пары с одинаковым ключом.

3. Стадия Reduce. На данной стадии применяется функция Reduce, которая на вход получает ключ и список всех значений, которые были сгенерированы для этого ключа в качестве параметра. Ключи представлены в отсортированном порядке. Для каждого ключа вызывается функция Reduce, которая возвращает финальный результат. Таким образом, происходит свертывание данных и список преобразуется к единственному атомарному значению.

Программная модель MapReduce – сердце платформы Hadoop. Данная платформа представляет из себя целую экосистему, которая является масштабируемой, надежной средой, используемой для распределенных вычислений.

MapReduce был выпущен с начальными версиями платформы Hadoop. Но основной недостаток заключался в том, что фреймворк выполнял как задачу обработки, так и задачу управления ресурсами.

Map Reduce 2 – долгожданное обновление для методов, связанных с планированием и управлением ресурсами. Улучшения отделяют возможность управления ресурсами от логики, специфичной для MapReduce, и такое разделе-

ние было достигнуто благодаря появлению YARN в более поздних версиях Hadoop.

Ядро современной платформы Hadoop [1, с.53] представляет распределенная файловая система Hadoop (HDFS), Hadoop YARN, Hadoop MapReduce и Hadoop Common (набор общих утилит и библиотек, которые поддерживают другие модули Hadoop).

Так как Hadoop представляет собой набор всех модулей, то он может включать в себя и другие языки программирования.

Hadoop решает проблемы, связанные с объемом и скоростью в горизонтальном масштабе. Он считается универсальным решением. Поэтому данная платформа стремительно набрала популярность. Помимо этого, Hadoop имеет открытый исходный код. Экосистема непрерывно развивалась в последние годы, что сделало ее максимально безошибочной.

Hadoop максимально эффективно работает с небольшим количеством больших файлов. Но данная платформа терпит неудачу, когда ставится задача обработки большого числа маленьких файлов. Также Hadoop осуществляет считывание и запись данных с диска, что делает операции чтения и записи весьма затратными.

Данная платформа поддерживает только механизм пакетной обработки. Hadoop не может производить вывод в режиме реального времени с низкой задержкой. Он работает только с данными, которые собираются и хранятся в файле заранее перед обработкой. Помимо этого, безопасность данной платформы является понятием неоднозначным. Hadoop использует аутентификацию Kerberos, которой сложно управлять.

Необходимость более быстрой обработки наборов данных привела к появлению Apache Spark. Это фреймворк с открытым исходным кодом, предназначенный для кластерных вычислений в режиме реального времени. В отличии от Hadoop, который основан на концепции пакетной обработки, Spark может обрабатывать данные в режиме реального времени и быть примерно в 100 раз быстрее. Основные особенности Spark: кластерные вычисления в памяти, параллелизм данных и отказоустойчивость. Архитектура основана на двух основных абстракциях: устойчивый распределенный набор данных (RDD) и направленный ациклический граф (DAG).

Экосистема Apache Spark включает в себя следующее [5, с. 21]:

1. Spark Core – ядро, предназначенное для распределенной крупномасштабной и параллельной обработки данных. Основные задачи: управление памятью, устранение неисправностей, планирование, распределение и мониторинг заданий в кластере и взаимодействие с системами хранения.

2. Spark Streaming – компонент, который используется для обработки потоковых данных в реальном времени. Он обеспечивает высокую пропускную способность и отказоустойчивую обработку потоков данных.

3. GraphX – компонент, предназначенный для графов и выполнении параллельных вычислений над ними.

4. MLlib – библиотека машинного обучения.

5. Spark SQL – новый модуль в Spark, который объединяет реляционную обработку с API функционального программирования Spark. Он поддерживает запросы данных либо через SQL, либо через Hive Query Language.

6. SparkR – пакет R. Под R [2, с.28] здесь понимается как язык программирования, так и программная среда, предназначенная для работы со статистическими данными. Интегрированные среды разработки, такие как Eclipse и Visual Studio, поддерживают этот язык. Некоторые компании считают, что по своей популярности данный язык уже обогнал SQL.

Таким образом, Spark превосходит Hadoop, когда данные извлекаются из разных источников и необходимо получить потоковую обработку больших данных в режиме реального времени.

Из недостатков Spark можно перечислить следующее. Хранение данных в памяти может стать узким местом, когда речь идет об экономически эффективной обработке больших данных. Также гораздо удобнее запускать все на одном узле, Spark же требует распределения по нескольким кластерам. Spark потребляет огромное количество данных по сравнению с Hadoop.

Еще одна технология обработки больших данных – нереляционные базы данных NoSQL. Они предназначены для неструктурированных данных. Такие базы данных не требуют схем и определения типов данных. К основным достоинствам можно еще причислить следующее: открытый исходный код, горизонтальную масштабируемость, распределенность, экономичность, высокую производительность, обработку данных в режиме реального времени и гибкость [4, с. 5]. Примеры баз данных NoSQL: MongoDB, Redis и Cassandra.

Одна из разновидностей нереляционной базы данных – резидентная база данных, которая использует оперативную память для хранения данных. Резидентная база данных предназначена для достижения минимального времени отклика за счет исключения необходимости доступа к дискам. Идеально подходит для таких приложений как таблицы лидеров игр, торги в реальном времени и т.д.

Говоря об обработке больших данных, нельзя не упомянуть про искусственный интеллект. Практически любой современный продукт для обработки больших данных использует искусственный интеллект, который продвигается через машинное обучение [3, с. 42]. Отдельно стоит выделить нейронные сети и глубинное обучение. Комбинация искусственного интеллекта и больших данных является причиной феноменального роста во многих отраслях.

Искусственный интеллект успешно применяется в аналитике больших данных. Это главным образом включает применение различных алгоритмов интеллектуального анализа к конкретному набору данных, которые затем помогут компании более эффективно принять решение. Типы аналитики больших данных: описательная (descriptive analytics), прогнозирующая аналитика (predictive analytics), предписывающая аналитика (prescriptive analytics), диагностирующая аналитика (diagnostic analytics).

Объединения технологий обработки больших данных с блокчейном дает отличные результаты. Самое большое преимущество блокчейна – это безопасность. Особенно эффективно применение данной комбинации для экономической отрасли, сферы государственного управления.

Благодаря быстрому росту данных и огромному стремлению организаций к повышению эффективности своей деятельности, технологии обработки больших данных принесли на рынок столько зрелых идей и проектов. Это помогло решить многие бизнес-задачи и проблемы, а также качественным образом улучшить жизнь людей. Но объем информации до сих пор продолжает расти и есть

риск, что те технологии, которые используются на данный момент времени в скором могут быстро потерять свою актуальность. Поэтому, можно сказать, что развитие технологий обработки больших данных находится в прямой зависимости от темпов роста данных.

СПИСОК ЛИТЕРАТУРЫ

1. DeRoos Dirk. Hadoop For Dummies / published by: John Wiley & Sons, Inc. – 2014. – 411 p.
2. Prasad Y.L. Big Data Analytics Made Easy. – Published by: Notion Press, Inc. – 1 edition (December 3, 2016). – 192 p.
3. Бринк Хенрик, Ричардс Джозеф, Феверолф Марк. Машинное обучение. – СПб.: Питер, 2017. – 336 с.: ил. – (Сер. «Библиотека программиста»).
4. Воронова Л.И., Воронов В.И. Big Data. Методы и средства анализа: учебное пособие. – М.: Московский технический университет связи и информатики, 2016. – 33 с.
5. Карау Х., Конвински Э., Венделл П., Захария М. Изучаем Spark: молниеносный анализ данных. – М.: ДМК Пресс, 2015. – 304 с.
6. Силен Дэви, Мейсман Арно, Али Мохамед. Основы Data Science и Big Data. Python и наука о данных. – СПб.: Питер, 2017. – 336 с.: ил. – (Сер. «Библиотека программиста»).

BIG DATA PROCESSING TECHNOLOGIES

"Big Data" is the term that practically everyone has heard in the modern world. However, what is exactly meant by this term? How Big Data processing is implemented? This article answers all these questions. A clear definition of the term, the history of the origin and Big Data principal features are considered here. MapReduce programming model and frameworks used for Big Data analytics are also given in the article. Description and examples of non-relational databases are considered there. The use of artificial intelligence for Big Data effective processing and the integration of Big Data and blockchain are also given in the article.

Key words: *unstructured data analysis, Big Data, Big Data characteristics, big data processing technology, MapReduce programming model, Hadoop framework, Apache Spark framework, NoSQL Non-Relational Databases, in-memory database, artificial intelligence, machine learning, blockchain.*

Научное издание

СОВРЕМЕННАЯ НАУКА: ТРАДИЦИИ И ИННОВАЦИИ

Сборник научных статей
по итогам II молодежного конкурса
научных работ

Подписано в печать 05.06 2020 г. Формат 60×84/16.
Бумага офсетная. Гарнитура Таймс.
Уч.-изд. л. 9,6. Тираж 50 экз.

Отпечатано в типографии «Сфера».
г. Волгоград, улица им. Менделеева, д. 43, офис 2/1.