Научная статья
УДК 681.3; DOI: 10.61260/2307-7476-2023-4-45-52

ПРОГРАММНЫЕ СРЕДСТВА ОБРАБОТКИ БОЛЬШИХ ОБЪЕМОВ ДАННЫХ

✉Лабинский Александр Юрьевич.
Санкт-Петербургский университет ГПС МЧС России, Санкт-Петербург, Россия
✉labynsciy@yandex.ru

Аннотация. Рассмотрены возможности программных средств обработки больших объемов данных (Big Data). В центре внимания статьи находятся инструменты платформы Apache NiFi, входящие в набор Hadoop-инструментов для бизнес-экосистем.

Подробно рассмотрены такие средства, как свободно распространяемый набор утилит и библиотек для разработки и выполнения распределенных программ (Hadoop Common), включающий в себя библиотеки управления системами файлов и сценарии по управлению распределённой обработкой данных и созданию инфраструктуры, необходимой для этой обработки.

Рассмотрены инструменты платформы Apache NiFi, в том числе набор современных ETL-инструментов (Extract, Transform, Load) для разработки хранилища большого объема данных, а также основные понятия платформы Apache NiFi, использующей концепцию «Flow Based Programming» (FBP).

Произведена оценка эффективности параллельной обработки данных.

Ключевые слова: программные средства, большие объемы данных, параллельная обработка данных, платформа Apache NiFi, ETL-инструменты, Hadoop-инструмены, бизнес-экосистема, концепция Flow Based Programming, дистрибутив Hortonworks Data Platform

Для цитирования: Лабинский А.Ю. Программные средства обработки больших объемов данных // Природные и техногенные риски (физико-математические и прикладные аспекты). 2023. № 4 (48). С. 45–52. DOI: 10.61260/2307-7476-2023-4-45-52.

Scientific article

PROCESSING SOFTWARE BIG DATA

✉Labinskiy Alexander Yu.
Saint-Petersburg university of State fire service of EMERCOM of Russia, Saint-Petersburg, Russia
✉labynsciy@yandex.ru

Abstract. The article considers possibilities of software tools for processing large volumes of data (Big Data). The article focuses on the Apache NiFi platform tools, which are part of the Hadoop suite of tools for business ecosystems.

Tools such as Hadoop Common, which include libraries for managing the file systems supported by Hadoop, and scenarios for creating the necessary infrastructure and managing distributed data processing, are discussed in detail.

The tools of the Apache NiFi platform are considered, including a set of modern ETL-tools (Extract, Transform, Load) for the development of a large data storage, as well as the basic concepts of the Apache NiFi platform, based on the concept of «Flow Based Programming» (FBP).

The evaluation of the efficiency of parallel data processing has been made, which has shown that with the increase of the share of consecutive operations in the computer program of data processing the degree of acceleration of calculations decreases.

The topic of the article is relevant, as large data sets are now used everywhere and their processing daily gives a significant positive effect.

Keywords: software, large amounts of data, parallel data processing, Apache NiFi platform, ETL-tools, Hadoop-tools, business ecosystem, Flow Based Programming concept, Hortonworks Data Platform distribution

For citation: Labinskiy A.Yu. Processing software Big Data // Prirodnye i tekhnogennye riski (fiziko-matematicheskie i prikladnye aspekty) = Natural and man-made risks (physico-mathematical and applied aspects). 2023. № 4 (48). P. 45–52. DOI: 10.61260/2307-7476-2023-4-45-52.

Введение

При обработке больших объемов данных (Big Data) используется понятие бизнес-экосистемы (Business ecosystem) [1]. Бизнес-экосистема – это набор собственных или партнерских сервисов, объединённых вокруг одной компании. Бизнес-экосистема может быть сосредоточена вокруг одной сферы жизни клиента или проникать сразу в несколько из них.

Самые яркие примеры бизнес-экосистем в России – Сбербанк, Яндекс, Тинькофф, VK и МТС. Они строят бизнес-экосистемы так, чтобы затронуть как можно больше повседневных потребностей клиента. При этом граница между банками и небанковскими компаниями размывается. Например, все пять экосистем присутствуют в таких категориях, как «Финансы», «Автомобиль», «Коммуникации», «Медиа», «Развлечения» и «Здоровье».

Сформулируем постановку задачи, результаты решения которой представлены в данной статье. Нужно произвести обзор программных средств обработки больших объемов данных, реализующих массово-параллельную архитектуру обработки данных.

Новизна исследования, отражающая личный вклад автора, заключается в том, что произведена оценка эффективности параллельной обработки данных, в результате которой показано, что с увеличением доли выполняемых последовательно операций в компьютерной программе обработки данных степень ускорения вычислений уменьшается.

Актуальность обработки больших объемов данных

Технологии обработки больших объемов данных (Big Data) позволяют извлекать значительную пользу из больших данных, которые встречаются повсеместно. Ниже приведены примеры эффекта, получаемого от использования BigData [1].

Банковская деятельность – привлечение новых клиентов и поддержка взаимодействия с ними.

Телекоммуникации – существенное уменьшение времени обработки запросов пользователей.

Медицина – снижение смертности пациентов, благодаря анализу электронных медицинских карт.

Энергетика – повышение эффективности использования энергии потребителями.

Средства обработки больших объемов данных

Для обработки больших объемов данных используется HDFS (Hadoop Distributed File System) – файловая система, предназначенная для хранения файлов больших размеров, поблочно распределённых между узлами вычислительного кластера. Свободно распространяемый набор библиотек и утилит, используемый для разработки и выполнения распределенных программ (Hadoop), работает на кластерах, состоящих из многих тысяч узлов компьютерной сети [2, 3].

Программный продукт Hadoop Common содержит библиотеки управления системами файлов и сценарии по управлению распределённой обработкой данных и созданию инфраструктуры, необходимой для этой обработки. Кроме того, в состав Hadoop Common входит специальный упрощенный интерпретатор командной строки, облегчающий ввод, отладку и запуск команд сценариев по управлению распределённой обработкой данных. Указанный интерпретатор может быть запущен из оболочки операционной системы.

Программный продукт Hadoop Common является ключевой технологией обработки больших объемов данных, что дает основания предполагать возможность его массового использования в области распределённой обработки данных (рис. 1) [4].

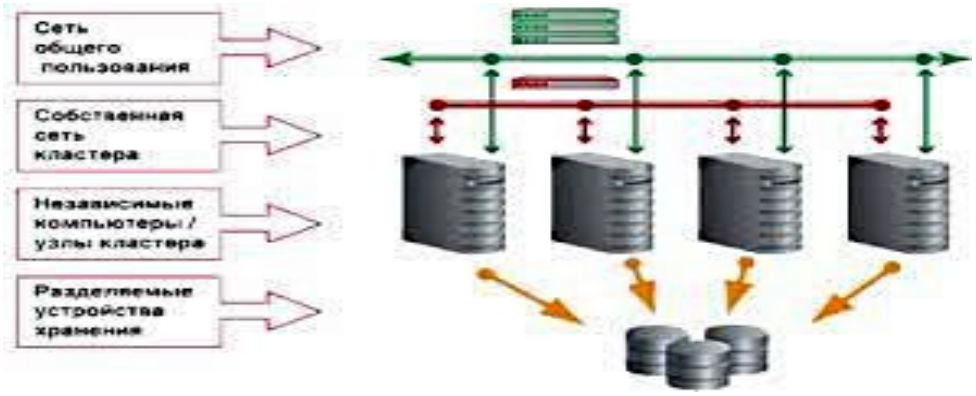

Рис. 1. **Массово-параллельная архитектура обработки данных**

В настоящее время в области распределённой обработки данных широко используется класс архитектур параллельных вычислительных систем, называемый массово-параллельной архитектурой (massive parallel processing), особенностью которой является физическое разделение памяти.

Система, реализующая массово-параллельную архитектуру, состоит из узлов, соединенных коммуникационными каналами. Узлами данной системы могут быть: адаптеры сети, устройства ввода-вывода, жесткие диски, банки оперативной памяти и процессоры данного узла, а также другие устройства.

Для хранения файлов больших размеров, распределённых между узлами вычислительного кластера, используется файловая система HDFS, которая обеспечивает безотказную работу системы с массово-параллельной архитектурой. Надежность работы HDFS обеспечивается тем, что файлы больших размеров разделяются на блоки, каждый из которых хранится на нескольких узлах сети (репликация данных), что делает распределённую систему устойчивой к отказам отдельных узлов.

Программный продукт Hadoop разрабатывался с целью обеспечения возможности построения системы, реализующей массово-параллельную архитектуру на основе оборудования массового класса (серверов, устройств хранения данных, адаптеров, устройств ввода-вывода), составляющую основу сетей хранения данных [5]. В качестве примеров таких сетей хранения данных могут быть названы следующие сети сервисов Интернет:

– Yahoo (емкость хранения 15 Пбайт, 4 000 узлов);
– Facebook (емкость хранения 21 Пбайт, 2 000 узлов);
– Ebay (емкость хранения 16 Пбайт, 700 узлов).

Обычно один узел такой сети способен обрабатывать 100 млн имен файлов.

Оценка эффективности параллельной обработки данных

Вопросам оценки эффективности параллельной обработки данных посвящена работа [1]. Выбор рациональных способов параллельной обработки данных зависит от конкретных технологий параллельного программирования в соответствии с рациональными планами выполнения параллельных программ (ПВПП) при описании алгоритмов в виде графов. При этом предпочтительным считается получение ПВПП с максимальным использованием вычислительных ресурсов, так как такая цель соответствует представлению о высокой плотности кода.

При оценке эффективности используется технология получения ПВПП на основе модели потокового (Data-Flow) вычислительного процесса [6–8]. В целом с помощью предложенного подхода можно численно оценивать способность алгоритмов, представленных информационными графами в ярусно-параллельной форме, к реорганизации путём применения эвристических методов. Далее оценка эффективности параллельной обработки данных может производиться на основе закона Амдала.

Пусть в компьютерной программе доля операций, которые нужно выполнять последовательно, равна P, где 0 <= P <= 1. Тогда при P = 0 все операции выполняются параллельно и при P = 1 все операции выполняются последовательно. Для оценки ускорения вычислений на компьютере, содержащем N процессоров, можно использовать закон Амдала в виде формулы:

$$V \leq 1/[F + (1 - F)/N],$$

где V – степень ускорения вычислений.

Результат оценки ускорения вычислений представлен на рис. 2:

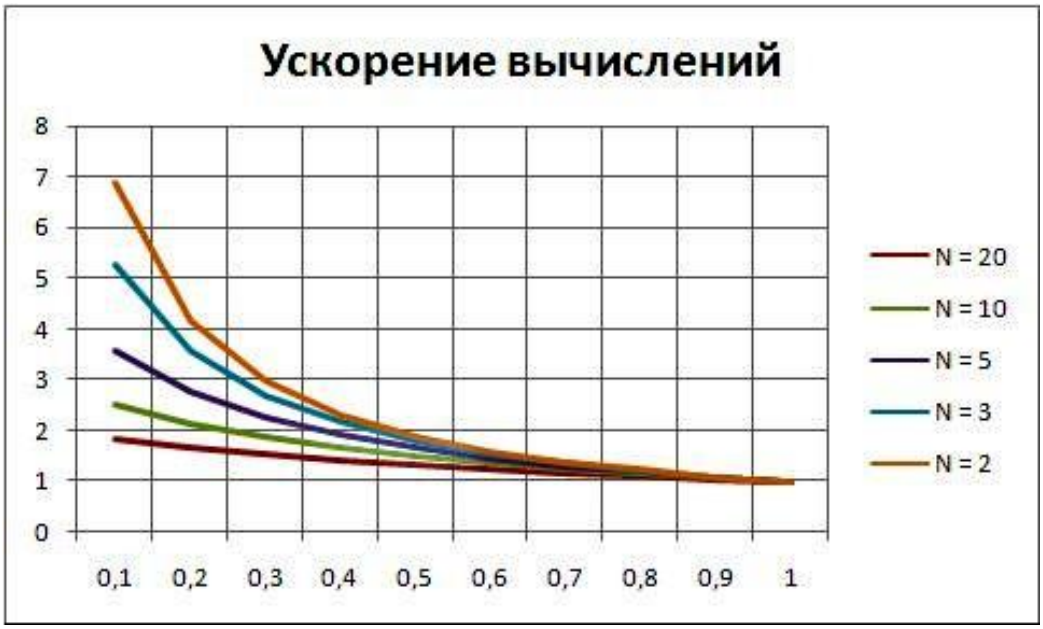

Рис. 2. **Оценка ускорения вычислений**

Как видно на графике (рис. 2), с увеличением доли выполняемых последовательно операций от 0 до 1 и уменьшением числа процессоров от 20 до 2 степень ускорения вычислений уменьшается.

Инструменты платформы Apache NiFi

В настоящее время для разработки ПО, используемого для обработки больших объемов данных, активно используются инструменты платформы Apache NiFi, входящие в набор Hadoop-инструментов для бизнес-экосистем. Apache NiFi – это набор современных ETL-инструментов (Extract, Transform, Load – дословно «извлечение, преобразование, загрузка») для разработки хранилища большого объема данных, которое включает в себя такие процессы, как извлечение данных из внешних источников, трансформацию и очистку данных с целью их соответствия потребностям бизнес-модели, а также загрузку данных в хранилище данных [9–11].

Архитектура платформы Apache NiFi представлена на рис. 3:

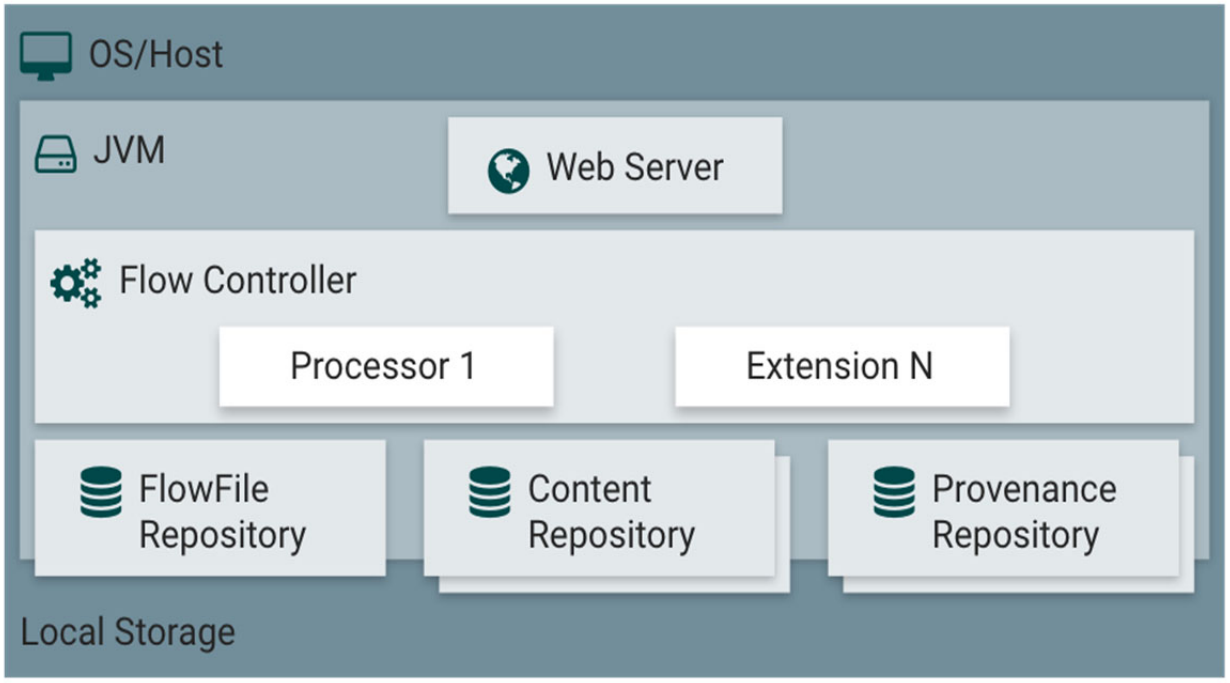

Рис. 3. **Архитектура платформы Apache NiFi**

Архитектуру хранилища данных можно представить в виде трёх компонентов:

– источник данных (FlowFile Repository) содержит структурированные данные в виде таблиц, совокупности таблиц или просто файла (данные в котором разделены символами-разделителями);

– промежуточная область (Content Repository) содержит вспомогательные таблицы, создаваемые временно и исключительно для организации процесса выгрузки.

– получатель данных (Provenance Repository) – хранилище данных или база данных, в которую должны быть помещены извлечённые данные.

Перемещение данных от источника к получателю называют потоком данных (Data Flow).

Кроме базового набора модулей Hadoop от компании Apache Software Foundation (HDFS, MapReduce, Yarn и Hadoop Common) существует программный продукт HDP американской компании HortonWorks, который также содержит дополнительные решения Apache для работы с большими данными и машинным обучением [4]:

1) инструменты для управления потоками данных;
2) инструменты для обеспечения безопасности;
3) инструменты для планирования и координирования распределенной обработки задач;
4) реляционные и NoSQL системы управления базами данных;
5) инструменты для программирования запросов к большим слабоструктурированным наборам данных;
6) инструменты для потоковой обработки данных;
7) инструменты для полнотекстового и фасетного поиска, динамической кластеризации, интеграции с базами данных и обработка документов со сложным форматом.

Основные понятия платформы Apache NiFi

Платформа Apache NiFi опирается на концепцию «Flow Based Programming» (FBP), включающую в себя следующие понятия:

1. FlowFile – объект, который содержит значения в виде байтов и соответствующие атрибуты. Указанные значения являются данными в виде потока сообщений, а также могут быть

представлены как результат работы процессора (например, команды обработки данных SQL), содержащий атрибуты (метаданные FlowFile), сгенерированные в результате выполнения запроса.

2. FlowFile Processor представляет собой процессор, который выполняет основную работу в открытом программном обеспечении (Apache NiFi) проекта фонда Apache. Указанный процессор имеет одну или несколько функций по работе с объектом FlowFile, в том числе изменение содержимого, маршрутизация, создание, чтение и запись содержимого, чтение, запись и изменение атрибутов. Приведем примеры процессора FlowFile Processor:

– «ListenSyslog», который на входе принимает данные по syslog-протоколу, а на выходе создает объект FlowFile;

– «RouteOnAttribute», который на входе читает атрибуты входного объекта FlowFile, а на выходе перенаправляет объект FlowFile с целью подключения его к другому процессору.

3. Connection представляет собой коммутатор, который обеспечивает подключение и передачу объекта FlowFile к другим процессорам и некоторым другим сущностям открытого программного обеспечения Apache NiFi. Коммутатор Connection помещает объект FlowFile в очередь, после чего передает его далее по цепочке. Кроме того, коммутатор Connection позволяет настраивать порядок выбора объекта FlowFile из очереди.

4. Process Group представляет собой набор процессоров в рамках понятий открытого программного обеспечения Apache NiFi. В данном контексте понятие Process Group представляет собой также механизм организации множества компонентов в одну логическую структуру.

5. FlowFile Repository представляет собой хранилище файлов, доступных для дальнейшего распространения по сети. В этом хранилище содержится вся известная информация о каждом объекте FlowFile, существующем в данный момент в системе.

6. Content Repository представляет собой хранилище, в котором находится содержимое всех объектов FlowFile (передаваемые данные).

7. Provenance Repository представляет собой хранилище, в котором содержится информация о событиях, происходящих с каждым объектом FlowFile.

8. Date Like представляет собой метод хранения данных в натуральном (RAW) формате.

9. Web Server представляет собой веб-интерфейс и интерфейс прикладного программирования.

Hortonworks Data Platform

Hortonworks Data Platform (HDP) – дистрибутив Apache Hadoop с набором программ, библиотек и утилит компании Apache Software Foundation, адаптированных компанией Hortonworks для больших данных (Big Data) и машинного обучения (Machine Learning), бесплатно распространяемый и коммерчески поддерживаемый.

Помимо HDP, компания Hortonworks предлагает и другие продукты для Big Data и Machine Learning, также основанные на проектах Apache Software Foundation: Hortonworks DataFlow (HDF) – NiFi, Storm и Kafka, а также сервисы Hortonworks DataPlane: Apache Atlas и Cloudbreak для интеграции со сторонними решениями.

Структура Hortonworks Data Platform [11] представлена на рис. 4:

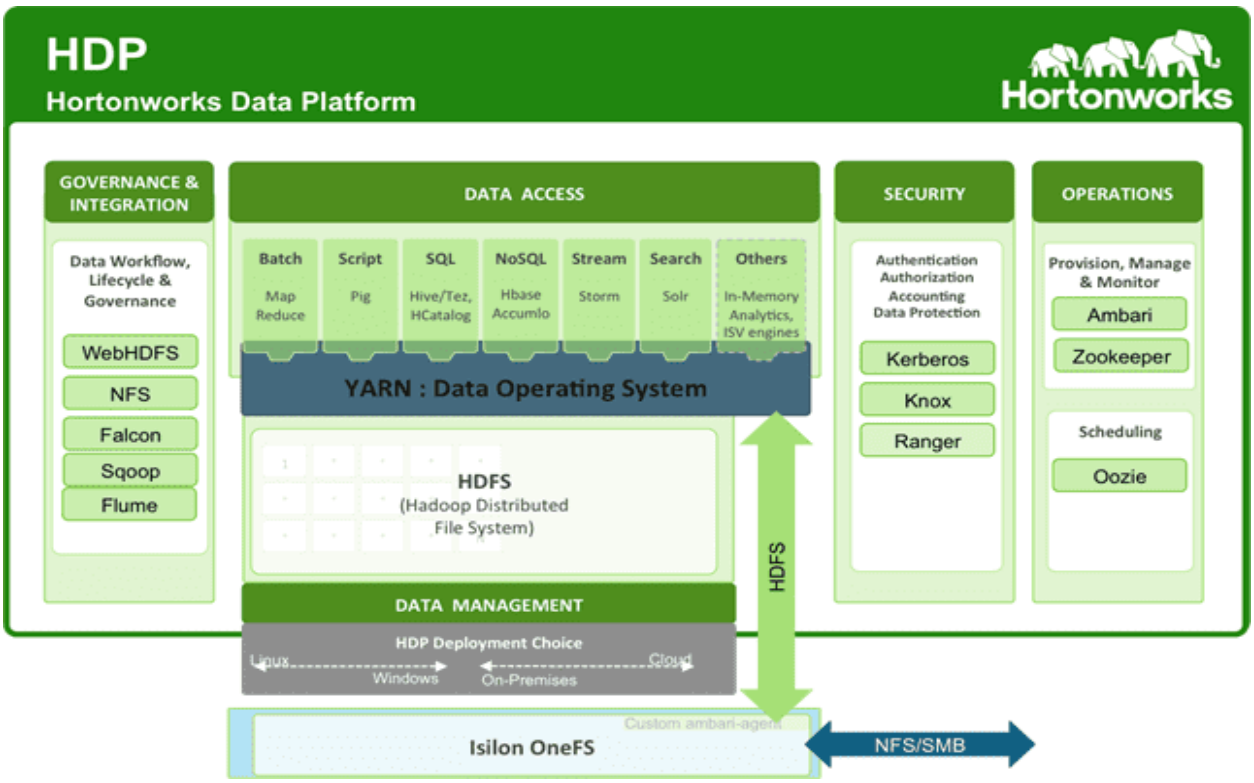

Рис. 4. **Структура дистрибутива Hortonworks Data Platform**

Вывод

Так как большие массивы данных в настоящее время используются повсеместно, и их обработка ежедневно дает существенный положительный эффект, тема статьи актуальна.

Для обработки больших массивов данных используется массово-параллельная архитектура, при которой осуществляется параллельная обработка данных. Произведена оценка эффективности параллельной обработки данных, в результате которой показано, что с увеличением доли выполняемых последовательно операций в компьютерной программе обработки данных степень ускорения вычислений уменьшается.

В настоящее время для разработки программного обеспечения, применяемого для обработки больших объемов данных, активно используются инструменты платформы Apache NiFi, входящие в набор Hadoop-инструментов для бизнес-экосистем.

В качестве примера можно привести использование компанией «Ростелеком» инструментов платформы Apache NiFi, в результате чего весь процесс обработки данных стал надежнее и удобнее.

Список источников

1. Баканов В.И. Динамика потоковых вычислений. М.: Труды НИУ ВШЭ, 2021.
2. Лэм Чак. Hadoop в действии. ДМК Пресс, 2012.
3. Уайт Том. Hadoop. Подробное руководство. СПб.: Питер, 2013.
4. Vance Ashlee. Hadoop, a Free Software Program, Finds Uses Beyond Search. N.Y.: The New York Times, 2009.
5. Shvachko Konstantin. Apache Hadoop. Coriolis, 2011.
6. Sharp J.A. Data Flow Computing: Theory and Practice. Intellect Limited, 1992.
7. Carkci M. Dataflow and Reactive Programming Systems: A Practical Guide. CreateSpace Independent Publishing Platform, 2014.
8. Wesley M. Johnston, J.R. Paul Hanna, Richard J. Millar. Advances in Dataflow Programming Languages. N.Y. and London, 2015.

9. David Loshin. ETL (Extract, Transform, Load) // Business Intelligence and Analytics. Morgan Kaufmann, 2012.

10. David Haertzen. ETL Tools // Business Intelligence and Analytics. Technics Publications, 2012.

11. Ralph Kimball, Joe Caserta. The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. John Wiley & Sons, 2004.

References:

1. Bakanov V.I. Dinamika potokovyh vychislenij. M.: Trudy NIU VSHE, 2021.
2. Lem Chak. Hadoop v dejstvii. DMK Press, 2012.
3. Uajt Tom. Hadoop. Podrobnoe rukovodstvo. SPb.: Piter, 2013.
4. Vance Ashlee. Hadoop, a Free Software Program, Finds Uses Beyond Search. N.Y.: The New York Times, 2009.
5. Shvachko Konstantin. Apache Hadoop. Coriolis, 2011.
6. Sharp J.A. Data Flow Computing: Theory and Practice. Intellect Limited, 1992.
7. Carkci M. Dataflow and Reactive Programming Systems: A Practical Guide. CreateSpace Independent Publishing Platform, 2014.
8. Wesley M. Johnston, J.R. Paul Hanna, Richard J. Millar. Advances in Dataflow Programming Languages. N.Y. and London, 2015.
9. David Loshin. ETL (Extract, Transform, Load) // Business Intelligence and Analytics. Morgan Kaufmann, 2012.
10. David Haertzen. ETL Tools // Business Intelligence and Analytics. Technics Publications, 2012.
11. Ralph Kimball, Joe Caserta. The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. John Wiley & Sons, 2004.

Информация о статье:
Поступила в редакцию: 28.08.2023
Принята к публикации: 10.11.2023
The information about article:
Article was received by the editorial office: 28.08.2023
Accepted for publication: 10.11.2023

Информация об авторах:
Лабинский Александр Юрьевич, доцент кафедры прикладной математики и информационных технологий Санкт-Петербургского университета ГПС МЧС России (196105, Санкт-Петербург, Московский пр., д. 149), кандидат технических наук, доцент, e-mail: labynsciy@yandex.ru, https://orcid.org/0000-0001-2735-4189, SPIN-код: 8338-4230
Information about the authors:
Labinsky Alexander Yu., associate professor of the department of applied mathematics and information technologies of Saint-Petersburg university of State fire service of EMERCOM of Russia (196105, Saint-Petersburg, Moskovsky ave., 149), candidate of technical sciences, associate professor, e-mail: labynsciy@yandex.ru, https://orcid.org/0000-0001-2735-4189, SPIN: 8338-4230