УДК 004.65

Informatics and Mathematical Methods in Simulation
Vol. 8 (2018), No. 2, pp. 146-155

РАСПРЕДЕЛЕННЫЙ ПРОГРАММНЫЙ КОМПЛЕКС НА БАЗЕ ФРЕЙМВОРКА APACHE SPARK ДЛЯ ОБРАБОТКИ ПОТОКОВЫХ BIG DATA ОТ СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ

В.В. Вычужанин

Одесский национальный морской университет,
ул. Мечникова, 34, Одесса, 65029, Украина; e-mail: 126.ist.onpu@gmail.com

Проведенный анализ систем, предназначенных для обработки данных, поступающих с информационно-измерительных систем сложных технических систем показал, что используемые в этих целях, например SCADA-системы, применяются главным образом для обеспечения обзора контролируемых процессов в сложных технических системах с возможностью выполнять способы Process Analyzer для анализа состояния систем и в основном для статистической обработки данных. В этой связи новые технологии обработки и методы анализа Big Data для этой сферы становятся более востребованными. Для решения поставленной задачи, связанной с обработкой данных, поступающих с информационно-измерительных систем сложных технических систем, в статье проведен анализ характеристик фреймворков Hadoop MapReduce и Apache Spark для обработки Big Data и их аналитики, обладающих внутри-гетерогенной памятью. Рассмотрено влияние на производительность и отказоустойчивость приложений Hadoop MapReduce и Apache Spark. Рассматриваются способы создания Resilient Distributed Data: распараллеливания переданной коллекции в программе; использование ссылок на внешнюю файловую систему в Hadoop. Описан распределенный программный комплекс на базе массово-параллельной технологии для обработки потоковых Big Data, поступающих с информационно-измерительных систем сложных технических систем. Отличительными особенностями системы являются ее способность работы в режиме реального времени с потоковыми Big Data, а также применение существующих алгоритмов, не предназначенных для распределенной обработки, на множестве узлов без изменения реализации последних. Предложено обработку потоковых Big Data в Apache Spark, поступающих с информационно-измерительных систем сложных технических систем, осуществлять на языке Scala с использованием библиотек SparkContext и RDD. Предлагаемый распределенный программный комплекс на базе массово-параллельной технологии с облачными вычислениями для обработки потоковых Big Data, поступающих с информационно- измерительных систем сложных технических систем, обладает способностью работы в режиме реального времени с большими объемами потоковых данных для управления технологическими процессами в сложных технических системах.

Ключевые слова: система, технология, большие данные, информация, база данных, обработка, анализ

Введение

С расширением функциональных возможностей развивающихся сложных технических систем (СТС) повышаются требования к надежности их работы [1,2]. Это оказывает существенное влияние на количество информационных источников в СТС, а также объем вычислительных действий по обработке Big Data, поступающих с информационно-измерительных систем (ИИС), выполнение которых необходимо для эффективного функционирования СТС [3], снижения риска их отказов [4].

Для сбора, обработки данных с ИИС СТС, а также для управления технологическими процессами в них используются SCADA-системы, работающие в

реальном времени [5]. Однако топологическая сложность подобных систем связана с существенными затратами, связанными с масштабированием и адаптацией к большому количеству информационно-измерительных сигналов, собранных для реконфигурации структуры управления СТС. Следует также отметить, что большинство SCADA-систем используется главным образом для обеспечения обзора контролируемых процессов в СТС с возможностью выполнять способы Process Analyzer в целях анализа состояния систем в основном для статистической обработки данных. В этой связи новые технологии обработки и методы анализа Big Data для этой сферы становятся более востребованными. Методы обработки Big Data [6,7] предусматривают анализ больших массивов данных с терабайтным или петабайтным масштабами в реальном времени. Сложной задачей является быстрая (с низкой задержкой) аналитика полного объема Big Data. Это означает необходимость сканирования терабайтов данных за секунды, что возможно только при обработке данных с большим параллелизмом.

Основные направления для анализа Big Data: Data Mining; краудсорсинг; машинное обучение; имитационное моделирование; визуализация данных; искусственные нейронные сети. Часто базовым принципом обработки Big Data является SN-архитектура, обеспечивающая параллельную, масштабируемую обработку без деградации на сотни и тысячи узлов кластера. Основные технологии обработки Big Datат: NoSQL; MapReduce (MR); Apache Hadoop; Apache Spark.

На основе проведенного анализа интерпретации информационных потоков в распределенных информационных системах Big Data установлено, что при создании таких систем не существует единых методов и технологий, объединяющих все этапы построения соответствующих кластерных систем. Проведение анализа характеристик двух перспективных файловых систем Hadoop MR и Spark, а также создание системы обработки Big Data в распределенных ИИС СТС является актуальным.

Целью работы является создание системы обработки Big Data в режиме реального времени на базе массово-параллельной технологии в распределенных ИИС СТС.

Основная часть

Анализ характеристик файловых систем Hadoop MR и Spark. Одной из архитектур по обработке Big Data, использующей поисковые алгоритмы, является реляционная система управления базами данных (СУБД). Недостатки использования таких СУБД привели к разработке адаптивно-подстраиваемой архитектуре, способной расширятся и масштабироваться при необходимости и постоянном увеличении данных. К такой технологии относится NoSQL [8]. Использование NoSQL архитектур, как систем хранения и обработки данных, не всегда является оптимальным, что связано со скоростью и производительностью жестких дисков. Решение части проблем найдено в поисковой архитектуре – экосистеме Hadoop. Фреймворк Apache Hadoop [9,10] - среда с открытым исходным кодом на Java для разработки и выполнения распределённых программ, работающих на вычислительных кластерах из сотен и тысяч узлов. Базовыми модулями Apache Hadoop [11] являются: Hadoop Common (набор инфраструктурных программных библиотек и утилит, используемых для других модулей и родственных проектов); HDFS (Hadoop Distributed File System) – распределенная файловая система; Hadoop YARN – модуль управления ресурсами кластера для выполнения приложений пользователя; Hadoop MR – модель программирования для обработки Big Data. Примерами проектов, входящих в экосистемы Hadoop, являются Apache Hive, Apache Pig, Apache HBase, Apache Spark и др. HDFS построена на основной подчиненной архитектуре, где «Name Node» - мастер, а «Data Nodes» - подчиненные узлы, в которых находятся фактические данные (рис.1). Для упрощения доступа к данным в хранилище Hadoop тспользуется SQL-подобный язык Hive, являющийся своего рода SQL для MR [12]. Масштабируемость достигается

параллельной обработкой фрагментов на узлах с использованием программной модели параллельных распределенных вычислений MR парадигмы, согласно которой приложение разделяется на большое количество одинаковых элементарных заданий, выполнимых на узлах кластера и естественным образом сводимых в конечный результат. Стандартным использованием в Hadoop архитектуре является применение MR задач (рис. 2). MR соответствует модели функционального программирования [13] и выполняет явную синхронизацию на этапах вычислений. MR предоставляет простой API программирования с точки зрения функций map () и reduce () [14]. Apache Hadoop MR - свободная платформа для организации обработки Big Data в петабайтах с использованием легко масштабируемой парадигмы MR, являющейся отказоустойчивым решением.

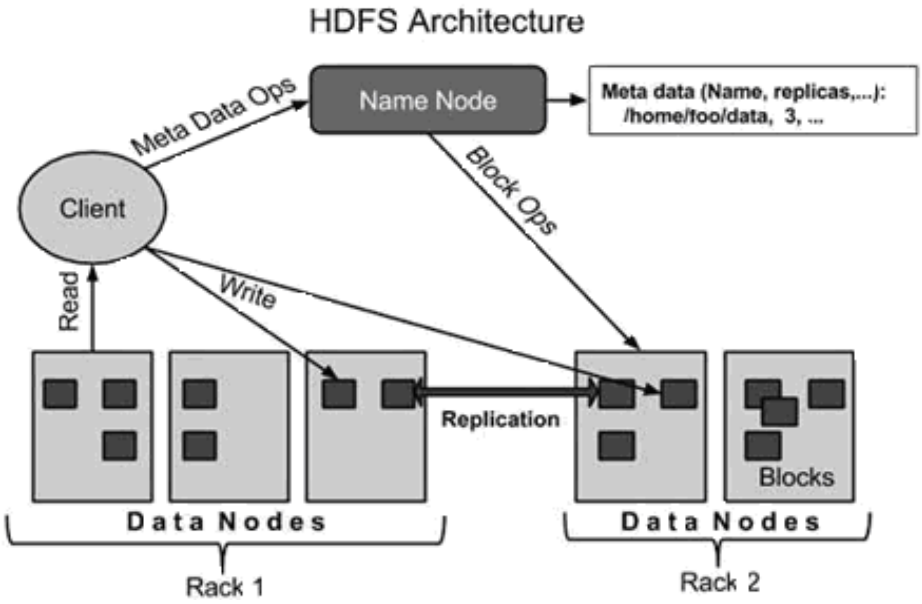

Рис. 1. Архитектура файловой системы Hadoop

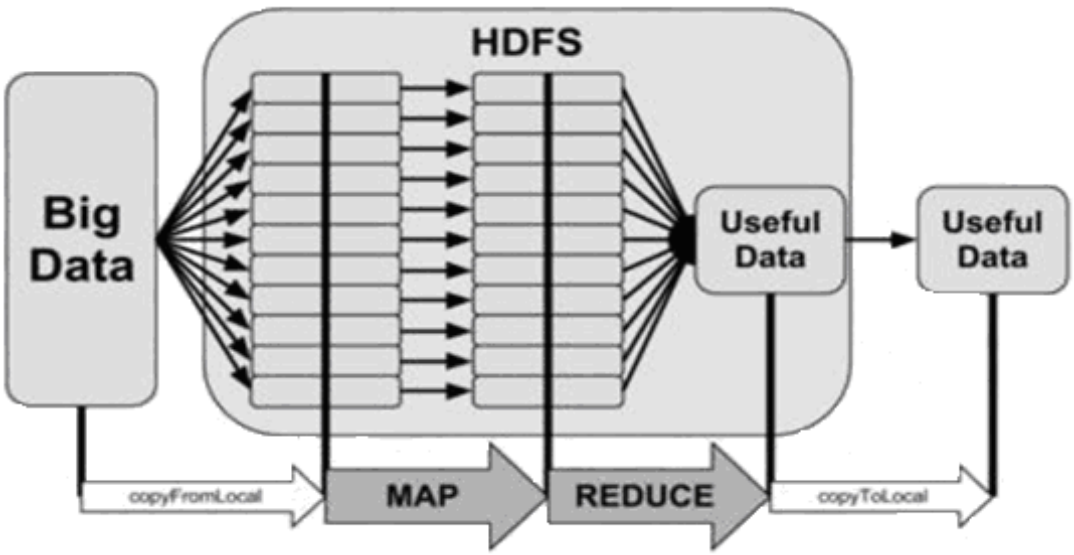

Рис. 2. Операции MapReduce

MR функционирует в режиме реального времени с потоковыми Big Data, применяя алгоритмы, не предназначенные для распределенной обработки на множестве узлов без изменения реализации последних. Hadoop MR позволяет создавать задания,

как с базовыми обработчиками, так и со свертками, написанными без использования Java. Утилиты Hadoop streaming используют в качестве базовых обработчиков и сверток любой исполняемый файл, работающий со стандартным вводом-выводом операционной системы (например, утилиты командной оболочки UNIX). Есть также SWIG-совместимый прикладной интерфейс программирования Hadoop pipes на C++. В состав дистрибутивов Hadoop входят реализации различных базовых обработчиков и свёрток, наиболее типично используемых в распределённой обработке. Ограничение MR состоит в том, что она предполагает пакетное выполнение, при котором значительные объемы данных целиком проходят цепочку трансформаций, занимающих много времени, что не пригодно для итеративных ответов пользователям.

Для быстрого анализа потоковых данных в режиме реального времени также используется Apache Spark [14,15] - универсальное средство анализа информации Big Data, позволяющее обрабатывать данные, размещенные на платформах хранения данных Hadoop, Cassandra, Mesos, S3 и т.д. Apache Spark предоставляет стек библиотек, включая SQL и DataFrames, MLlib для машинного обучения, GraphX и Spark Streaming. По сравнению с Hadoop MR, Apache Spark обеспечивает в 100 раз большую производительность при обработке данных в памяти и в 10 раз больше - при размещении данных на дисках. Apache Spark предоставляет пользователю дружественный интерфейс программирования для уменьшения усилий по кодированию, используя концепцию распределенных коллекций данных Resilient Distributed Data (RDD). В RDD хранятся данные в памяти между запросами без репликации, увеличивая производительность пакетной обработки, восстанавливая утраченные данные. Apache Spark можно использовать в интерактивном режиме из оболочек Scala, Python, Java,, R и SQL. RDD поддерживают множество итерационных алгоритмов, а также интерактивный интеллектуальный анализ данных и высокоэффективный механизм SQL Shark. На рис.3, 4 показаны итерационные операции на Spark RDD с промежуточными результатами в распределенной памяти вместо стабильного хранилища (диск). Если распределенной памяти достаточно для хранения промежуточных результатов, то она сохранит эти результаты на диске.

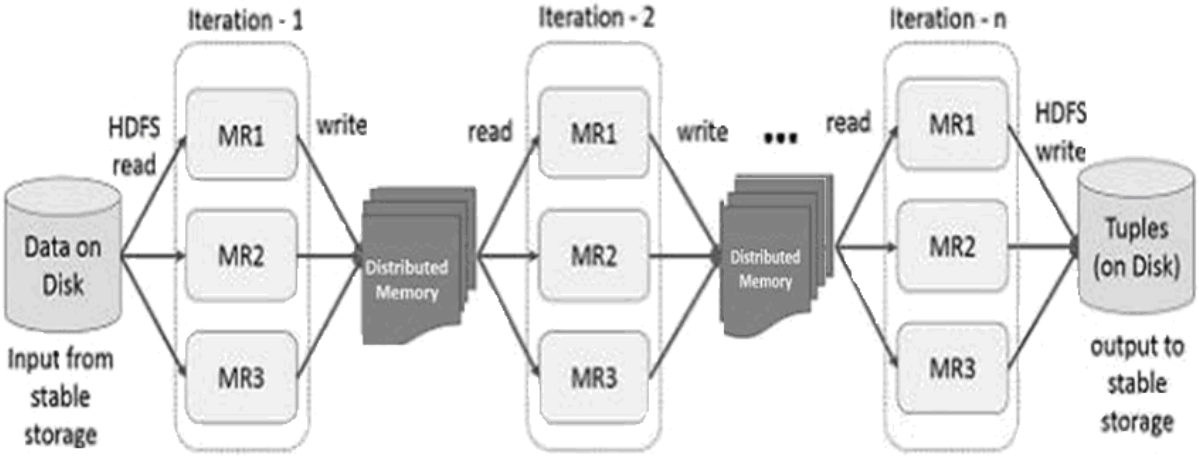

Рис. 3. Итеративные операции с Apache Spark RDD

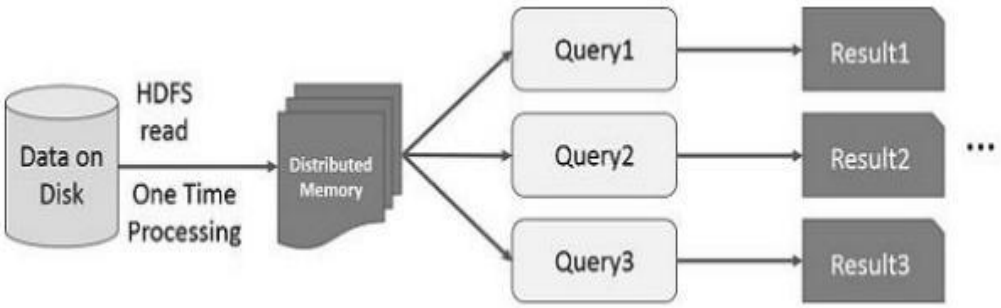

Рис. 4. Интерактивные операции с Apache Spark RDD

Преимуществами Apache Spark по сравнению с Hadoop MR являются: высокая масштабируемость добавлением новых узлов в вычислительный кластер без необходимости внесения изменений в применяемые алгоритмы; встроенная возможность работы в режиме реального времени, позволяющая построить алгоритмы потоковой обработки данных; большое количество вспомогательных программных решений, необходимых для организации системы, поддерживающей полный цикл предметных задач. Сравнение технологий распределенных вычислений позволяет сделать выбор в пользу технологии Apache Spark.

Традиционная локальная реализация управления СТС требует значительных инвестиций в соответствующее оборудование и программное обеспечение. В целях аналитики Big Data, поступающих непрерывно с ИИС СТС, а также управления технологическими процессами в них возможно использование инфраструктуры облачных вычислений. В этих целях могут применяться методы Process Analyzer (PA), создающих среду, в которой почти все этапы функционирования СТС могут быть записаны и использованы не только для безопасности систем, но и для оптимизации процесса анализа Big Data в реальном времени. Применение облачных вычислений для обработки Big Data открывает возможности для значительного снижения издержек аналитики Big Data, поступающих с измерительных устройств СТС. Необходимо учитывать, как обрабатываются и управляются данные на различных кластерах, оказывающих существенное влияние при проектировании и эксплуатации систем управления и обработки данных на нескольких уровнях. Самым сложным является быстрое рассмотрение в полном большом наборе данных, что возможно при их подготовке с высоким параллелизмом.

На рис.5 показано, как доступные и обслуживаемые данные через интернет, где в Front – end tier обрабатываются данные через веб-сервер в виде запросов MySql или NoSql и далее аналитика данных Back-end уровня, выполняемая на Apache Spark по HDFS. В общей архитектуре предлагаемой облачной архитектуры используется распределённый программный *брокер сообщений* Apache Kafka - проект с *открытым исходным кодом*, написанным на языке программирования *Scala* и позволяющим отправлять и получать потоковую информацию через облако. Одной из особенностей реализации инструмента является применение техники, сходной с *журналами транзакций*, используемыми в *системах управления базами данных*. Apache Kafka это распределённая система легко масштабируемая, поддерживающая высокую пропускную способность, как со стороны источников, так и систем-подписчиков с возможностью временного хранения данных для последующей пакетной обработки. На рис.6 представлено взаимодействие Apache Kafka, Spark-кластера при передачах потоков данных в реальном времени о состоянии СТС с задействованным сервисом Kafka в облаке. В кластере данные проверяются, очищаются, агрегируются, организовываются и направляются в систему оптимального управления СТС с определением соответствующих рекомендаций по ее эксплуатации.

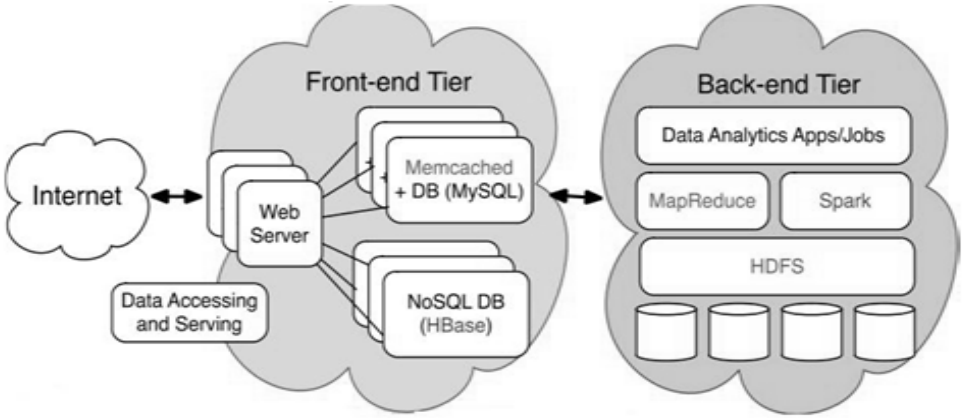

Рис. 5. Доступ к данным и обслуживание через интернет

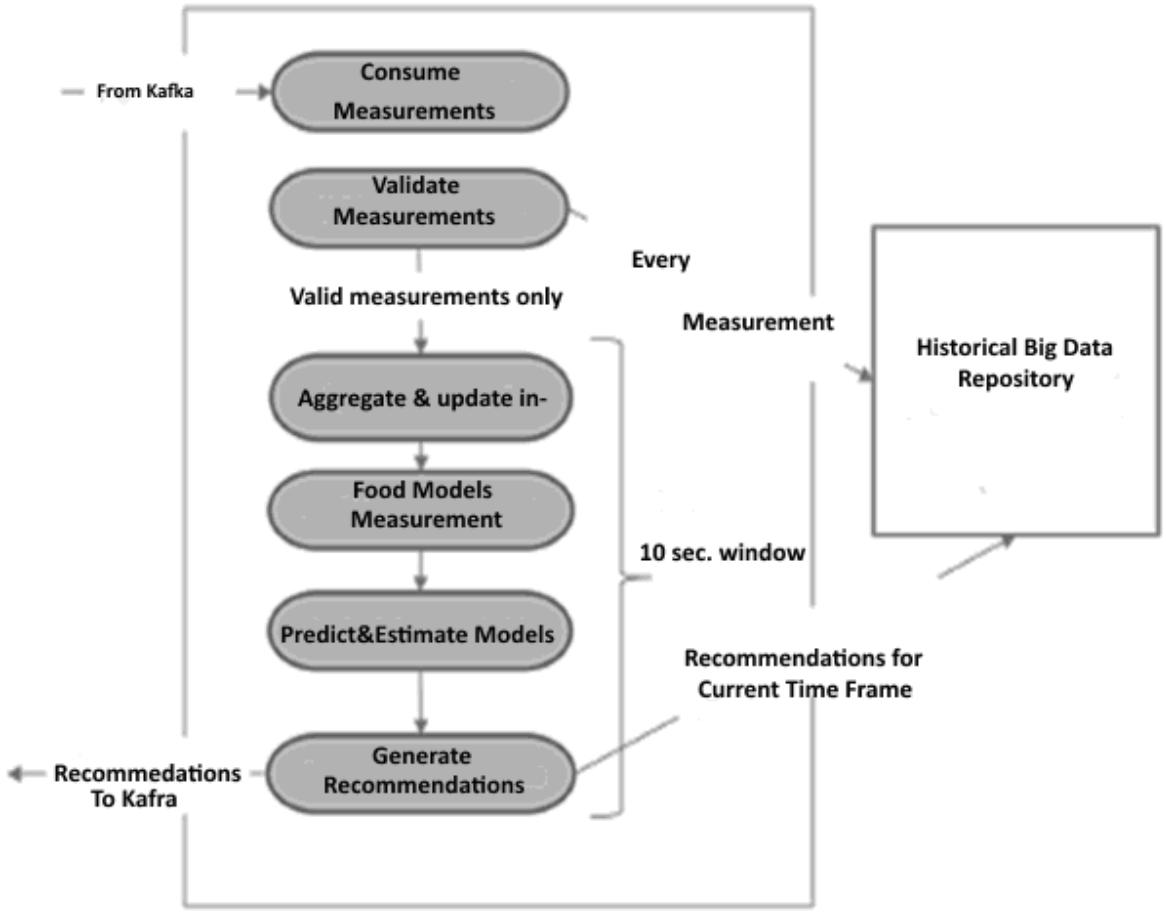

Рис. 6. Обзор потокового процесса

При переносе данных на облако, используется ПЭВМ, подключенная к технологическому процессу СТС, на основе применения MefosService, позволяющего выполнять синхронизацию данных с элементов СТС, а также создавать файл в json-структуре с полями, количество которых соответствует числу элементов СТС. Входные данные обрабатываются в Kafka сервере. Spark получает данные измерений с сервера Kafka, хранит их в памяти и передает имеющиеся модели технологических процессов в элементах СТС через установленный интервал времени. Процесс передачи данных отражен на рис. 6. В процессе Spark-Streaming метаданные синхронизируются и предварительно обрабатываются и выводятся с ПЭВМ Mefos-Service в Kafka, оттуда в Spark-кластер. В Spark-Streaming исходные данные накапливаются в памяти и сохраняются в репозитории данных. Данные рекомендаций по управлению также

накапливаются в памяти и сохраняются в репозитории Big Data. Большой репозиторий данных при необходимости позволяет провести углубленное их исследование в случае необходимости для разработки новых моделей технологических процессов в элементах СТС. Рекомендуется использовать Apache Spark с поддержкой кластера Kubernetes (рис.7). Учитывая, что Kubernetes является стандартом де-факто для управления контейнерными средами, вполне естественно поддерживать API-интерфейс Kubernetes в Spark. Приложение Spark в Kubernetes действует как пользовательский контроллер, создающий ресурсы Kubernetes в ответ на запросы, сделанные планировщиком Spark.

Чтобы просмотреть ресурсы Apache Spark, созданные в кластере, можно использовать команду kubectl в отдельном окне терминала.

```
$ kubectl get pods -l 'spark-role in (driver, executor)' -w
NAME    READY STATUS RESTARTS AGE
sparkpi-driver   1/1 Running 0 14s
        spark-pi-da1968a859653d6bab93f8e6503935f2-exec-1   0/1 Pending 0 0s
```

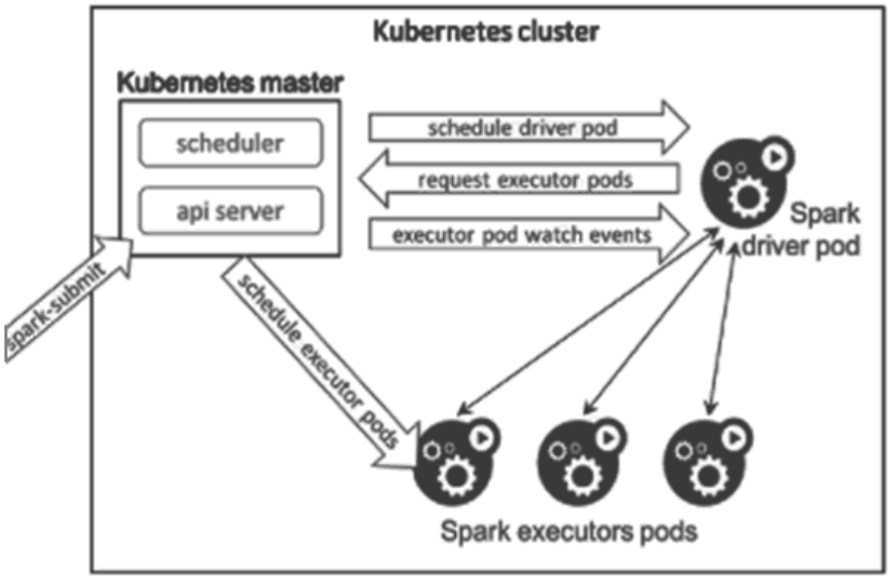

Рис. 7. Apache Spark 2.3 с поддержкой кластера Kubernetes

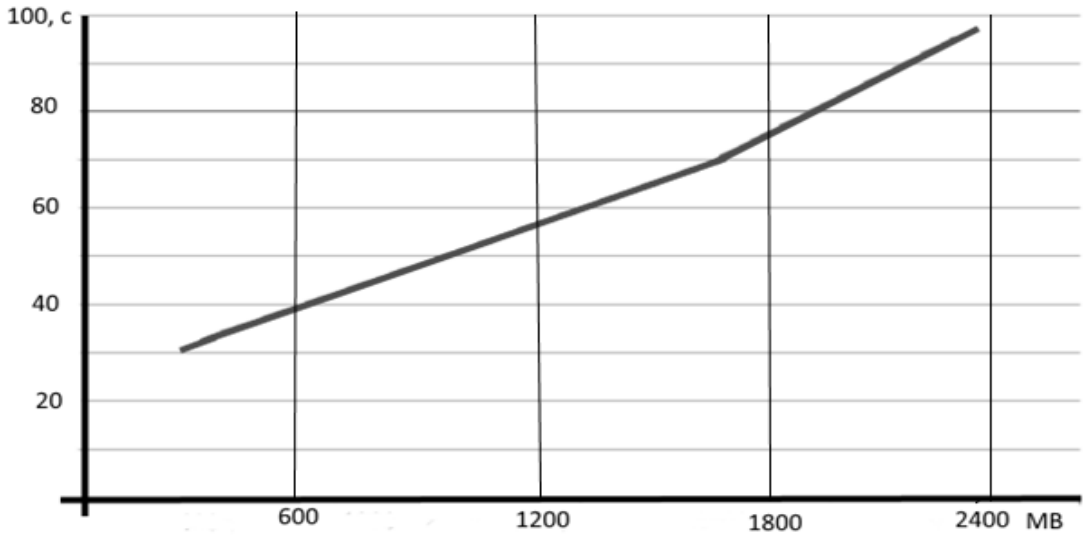

Рис. 8. График зависимости времени обработки Big Data от объема информации

Обработка потоковых Big Data в Apache Spark, поступающих с ИИС СТС, осуществлялось на языке Scala с использованием библиотек SparkContext и RDD. График зависимости времени обработки данных технологии Apache Spark в зависимости от объема анализируемой информации представлен на рис.8.

Выводы

Проведенный сравнительный анализ характеристик двух файловых систем Hadoop MapReduce и Apache Spark MapReduce и Spark позволил сделать выбор в пользу системы Apache Spark.

Предлагаемый распределенный программный комплекс на базе массово-параллельной технологии с облачными вычислениями для обработки потоковых Big Data, поступающих с информационно- измерительных систем СТС, обладает способностью работы в режиме реального времени с большими объемами потоковых данных для управления технологическими процессами в СТС

Список литературы

1. Vychuzhanin, V. Devising a method for the estimation and prediction of technical condition of ship complex systems / V. Vychuzhanin, N. Rudnichenko, V. Boyko, N. Shibaeva, S. Konovalov // Eastern-European Journal of Enterprise Technologies, 2016. — V. 84, № 6/9. — Pp. 4 -11.
2. Vychuzhanin, V.V., Assessment of risks structurally and functionally complex technical systems/ V.V. Vychuzhanin, N.D. Rudnichenko // Eastern-European Journal of Enterprise Technologies, 2014. — V. 1, № 2. — Pp. 18-22.
3. Рудниченко, Н.Д. Информационная когнитивная модель технологической взаимозависимости сложных технических систем / Н.Д. Рудниченко, В.В. Вычужанин // Информатика и математические методы в моделировании. — 2013. — №3. — С. 240-247.
4. Бойко, В.Д. Модель оценки живучести судовых технических систем / Бойко В.Д. Вычужанин В.В // Вестник Миколаївського кораблебудівного університету, 2012. — № 3. — С. 62-67.
5. D. Bailey and E. Wright, Practical SCADA for industry. Newnes, 2003.
6. P. Zadrozny and R. Kodali, Big Data Analytics using Splunk, Berkeley, CA, USA: Apress, 2013.
7 F. Ohlhorst, Big Data Analytics: Turning Big Data into Big Money, Hoboken, N.J, USA: Wiley, 2013.
8. Jaroslav Pokorny. NoSQL databases: a step to database scalability in web environment. Proceedings of the 13th International Conference on Information Integration and Web-based Application and Services, ACM New York, NY, USA, 2011. — Pp. 278-283.
9. Apache Hadoop Documentation 2014. Mode of access: http://hadoop.apache.org/.
10. White, T. Hadoop: The Definitive Guide. O'Reilly, 2012. Mode of access: http://cdn.oreillystatic.com/oreilly/booksamplers/9781449311520_sampler.pdf.
11. Shvachko, K., Hairong Kuang, Radia S, Chansler, R The Hadoop Distributed File System Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium
12. Вычужанин, В.В. Оптимізація відбору та аналізу інформації в різноструктурних сховищах даних / В.В. Вычужанин, Д.С. Шибаев, Н.О. Шибаева, Н.Д. Рудниченко // Інформатика та математичні методи в моделюванні, 2017. — Том 7, №4. — С.318-325.
13. P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar and R. Pasquin, "Incoop: MapReduce for incremental computations," Proc. of the 2nd ACM Symposium on Cloud Computing, 2011.
14. Apache Spark documentation 2014. Mode of access: https://spark.apache.org/documentation.html.
15. Apache Spark Research 2014. Mode of access: https://spark.apache.org/research.html.

РОЗПОДІЛЕНИЙ ПРОГРАМНИЙ КОМПЛЕКС НА БАЗІ ФРЕЙМВОРКА APACHE SPARK ДЛЯ ОБРОБКИ ПОТОКОВИХ BIG DATA ВІД СКЛАДНИХ ТЕХНІЧНИХ СИСТЕМ

В.В. Вичужанін

Одеський національний морський університет,
вул. Мечникова, 34, Одеса, 65029, Україна; e-mail: 126.ist.onpu@gmail.com

Проведений аналіз систем, призначених для обробки даних, що надходять з інформаційно-вимірювальних систем складних технічних систем показав, що використовувані в цих цілях, наприклад SCADA-системи, застосовуються головним чином для забезпечення огляду контрольованих процесів в складних технічних системах з можливістю виконувати способи Process Analyzer для аналізу стану систем і в основному для статистичної обробки даних. У зв'язку з цим нові технології обробки та методи аналізу Big Data для цієї сфери стають більш затребуваними. Для вирішення поставленого завдання, пов'язаної з обробкою даних, що надходять з інформаційно-вимірювальних систем складних технічних систем, в статті проведено аналіз характеристик фреймворків Hadoop MapReduce і Apache Spark для обробки Big Data і їх аналітики, що володіють внутрішньо-гетерогенної пам'яттю. Розглянуто вплив на продуктивність і відмовостійкість додатків Hadoop MapReduce і Apache Spark. Розглядаються способи створення Resilient Distributed Data: розпаралелювання переданої колекції в програмі; використання посилань на зовнішню файлову систему в Hadoop. Описано розподілений програмний комплекс на базі масово-паралельної технології для обробки потокових Big Data, що надходять з інформаційно-вимірювальних систем складних технічних систем. Відмінними рисами системи є її здатність роботи в режимі реального часу з потоковими Big Data, а також застосування існуючих алгоритмів, не призначених для розподіленої обробки, на безлічі вузлів без зміни реалізації останніх. Запропоновано обробку потокових Big Data в Apache Spark, що надходять з інформаційно-вимірювальних систем складних технічних систем, здійснювати на мові Scala з використанням бібліотек SparkContext і RDD. Пропонований розподілений програмний комплекс на базі масово-паралельної технології з хмарними обчисленнями для обробки потокових Big Data, що надходять з інформаційно-вимірювальних систем складних технічних систем, має здатність роботи в режимі реального часу з великими обсягами потокових даних для управління технологічними процесами в складних технічних системах.

Ключові слова: система, технологія, великі дані, інформація, база даних, обробка, аналіз

DISTRIBUTED SOFTWARE COMPLEX ON THE BASIC FORMER APACHE SPARK FOR PROCESSING THE FLOW BIG DATA FROM COMPLEX TECHNICAL SYSTEMS

V.V. Vychuzhanin

Odesa National Maritime University,
34, Mechnikova Str., Odesa, 65029, Ukraine; e-mail: 126.ist.onpu@gmail.com

The analysis of systems designed to process data from information systems of complex technical systems has shown that the SCADA systems used for this purpose are used mainly to provide an overview of the controlled processes in complex technical systems with the ability to perform the Process Analyzer methods for analysis of the state of systems and mainly for statistical data processing. In this regard, new processing technologies and Big Data analysis methods for this sphere are becoming more in demand. To solve the task of processing data from information and measuring systems of complex technical systems, the article analyzes the characteristics of the Hadoop MapReduce and Apache Spark frameworks for processing Big Data and their analytics with intra-heterogeneous memory. The impact on performance and fault tolerance of Hadoop MapReduce and Apache Spark applications is considered. The ways of creating Resilient Distributed Data are considered: the parallelization of the transferred collection in the program; the use of links to an external file system in Hadoop. The distributed program complex on the basis of mass-parallel technology for processing streams Big Data, coming from information-measuring systems of complex technical systems is described. Distinctive features of the system are its ability to work in real time with streaming Big Data, as well as the application of existing algorithms that are not designed for distributed processing on multiple nodes without changing the implementation of the latter. It is proposed to process streaming Big Data in Apache Spark, coming from information and measuring systems of complex technical systems, to implement in Scala language using the SparkContext and RDD libraries. The proposed distributed software package on the basis of mass-parallel technology with cloud computing for processing streaming Big Data, coming from information and measurement systems of complex technical systems, has the ability to work in real time with large volumes of streaming data for managing technological processes in complex technical systems.

Keywords: system, technology, large data, information, database, processing, analysis