Big Data Application in Functional Magnetic Resonance Imaging using Apache Spark

Saman Sarraf
Department of Electrical and Computer Engineering
McMaster University
Hamilton, ON, L8S 4L8, Canada
Rotman Research Institute at Baycrest
University of Toronto
Email: samansarraf@ieee.org

Mehdi Ostadhashem
Rogers Canada
Email: ostadhashem@gmail.com

***Abstract*—Recently, big data applications have been rapidly expanding into various industries. Healthcare is among those industries that are willing to use big data platforms, and as a result, some large data analytics tools have been adopted in this field. Medical imaging, which is a pillar in diagnostic healthcare, involves a high volume of data collection and processing. A massive number of 3D and 4D images are acquired in different forms and resolutions using a variety of medical imaging modalities. Preprocessing and analysis of imaging data is currently a costly and time-consuming process. However, few big data platforms have been created or modified for medical imaging purposes because of certain restrictions, such as data format. In this paper, we designed, developed and successfully tested a new pipeline for medical imaging data (in particular, functional magnetic resonance imaging - fMRI) using the Big Data Spark / PySpark platform on a single node, which allowed us to read and load imaging data, convert the data to Resilient Distributed Datasets in order to manipulate and perform in-memory data processing in parallel, and convert final results to an imaging format. Additionally, the pipeline provides an option to store the results in other formats, such as data frames. Using this new pipeline, we repeated our previous work, in which we extracted brain networks from fMRI data using template matching and the sum of squared differences (SSD) method. The final results revealed that our Spark (PySpark) based solution improved the performance (in terms of processing time) approximately fourfold when compared with the previous work developed in Python.**

***Keywords*—Big Data; Apache Spark; fMRI**

I. Introduction

Imaging modalities produce a significant amount of data. For example, in functional MRI (fMRI), which is among the most important neuroimaging methods, blood oxygen-level dependent (BOLD) signals of the whole brain are captured across time. This approach collects three dimensional data (x,y,z) over time (t) and stores four dimensional (4D) data [8]. Imaging data preprocessing and analysis is expensive in terms of infrastructure and processing time. A massive amount of data must be analyzed, and this creates a huge network of results. Furthermore, the results must be stored on disks for potential retrieved in the event that additional data analysis becomes necessary. One of the challenges in this field lies in how current medical image processing tools will evolve to incorporate big data resources more efficiently, as big data analytics platforms have been developed for large datasets. Therefore, the development of a pipeline allowing for the merger of big data and imaging appears necessary. This potential pipeline will allow us to read imaging data in an environment that can communicate with a big data platform. Next, using features of the big data platform, data manipulation, including data analysis, is performed in parallel, which not only improves processing times also allows for large datasets to be easily managed. Traditionally, the results of analyzed medical images are stored in any standard medical imaging format. However, the developed pipeline provides the option of storing results in other formats such as data frames which can be easily written and read by other big data analytics tools.

II. Background and Algorithms

A. Big Data

Big data is a developing term which is used to describe any voluminous amount of structured, semi-structured and unstructured data that has the potential to be mined for information. To describe the big data phenomenon, the four Vs Fig.1 are invoked: Volume, Velocity, Variety and Veracity [4][5]. Volume refers to the massive amount of data generated every moment. All manner of data emanating from the Internet, sensor and machine data, and healthcare data, to name just a few are collected and stored. Traditional analytics methods are not able to process this vast amount of data. In addition, storage and retrieval of big data requires more sophisticated infrastructure. By using big data technology, these datasets are able to be stored and utilized with the help of distributed systems in which parts of the data are stored in different locations that are connected by networks and combined through software. Velocity in big data refers to the speed at which new data is generated, and the speed of data retrieval and analysis is the focus of this paper. Big data technology enables us to analyze data both offline and in real time, before the data stream has even been stored in a database. Variety refers to the different types of data that are generated and used for various purposes, such as big data analytics. Over the past decade, not only has the volume of data recorded dramatically increased, but also the various types of data, as well as new varieties of data that have begun to be collected. In the past, the focus was more on structured data that fit into tables or relational databases, such as finance or healthcare data.

However, approximately 80 percent of the world's data is now unstructured and therefore requires big data tools to easily be transformed into new style tables or databases. For example, how can we collect imaging data in research centers and store in traditional databases? Big data technology has reliable solutions for different types of data, including sensor data and healthcare-related data. This technology brings unstructured or semi-structured data together with more traditional, structured data. Veracity speaks to the trustworthiness of the data. The accuracy and precision of collected data affect data analyses. There are a variety of forms of big data in which quality and accuracy are less controllable. Additionally, large volumes of data occasionally result in a lack of quality or accuracy. Big data tools and analytics technology now provide us with a means of analyzing these types of data big data [4][5] .

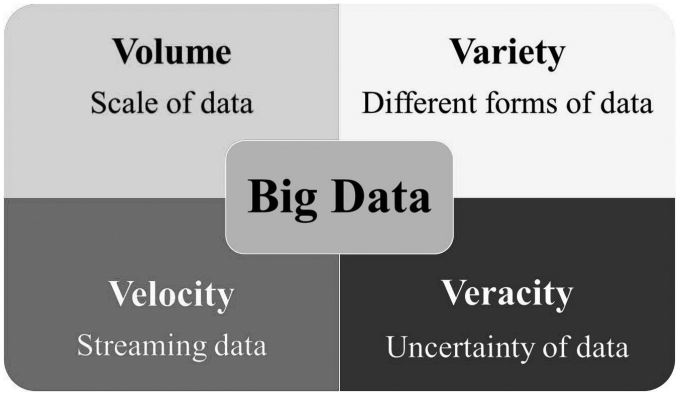

Fig. 1. 4 Vs in big data Volume, Variety, Velocity and Veracity

B. Big Data in Healthcare

In the healthcare industry, a huge volume of data is generated in different ways. Electronic health records, imaging data, and biosignals are some of the well-known types of data that are collected and stored in healthcare. Although healthcare data are typically stored as structured data, the lack of a golden standard for recording data in different healthcare divisions causes a variety of semi-structured or unstructured data to be stored. Big data analytics technology provides infrastructure to categorize and analyze healthcare data [1] [14] . Furthermore, by using this technology, novel and more accurate predictive models [7] are generated, and complicated patterns are extracted from the big data [1]. Two of the largest areas of healthcare that are grappling with all four V's of big data is medical imaging and imaging informatics. Different imaging modalities, a variety of acquisition times and resolutions, several imaging data formats, and finally, different source of noise in data clearly indicate that medical imaging encompasses the 4 Vs. However, few big data tools have been developed specifically for medical imaging. Given the lack of cutting-edge technology, imaging data are often processed and analyzed in a classic manner in which the performance (speed) is not high enough. The design of a new processing and analysis pipeline using big data tools appears necessary these days.

C. Big Data tools - Spark

Big data technologies and tools such as Hadoop or Apache Hadoop are open-source software programming platforms and projects for reliable, scalable, distributed computing [hadoop.apache.org]. The Apache Hadoop software library is a framework installed on specific hardware infrastructure that enables developers and users to perform distributed and parallel processing of large data sets across clusters and nodes by using programming models. In theory, Hadoop is designed to function in single servers or thousands of nodes performing local computation and storage [hadoop.apache.org]. This platform includes several sub-projects, such as HDFS, MapReduce and YARN. [3]. However, some of the difficulties in using the Hadoop framework, such as complicated installation processes and a high dependency on hardware structures served as motivations to develop other big data platforms. Additionally, Hadoop supports few programming languages and is not user friendly for data analysts who do not have a programming background. Therefore, big data platform developers strove to develop a software library supporting more programming languages, with less dependency on hardware and greater memory efficiency. Spark and Apache Spark are among the modern big data platforms [http://spark.apache.org/] that were originally developed at the University of California, Berkeleys AMP Lab. Apache Spark is a practical and fast general engine for big data processing. In memory data processing, Spark improved performance by as much as tenfold in terms of speed when compared with on-disk data processing. In addition, Spark offers more than 80 high-level applications that can interact with different programming languages such as Java, Scala, R and Python (which is more important in this paper). This feature demonstrates Spark's ease of use when compared to the Hadoop platform. Another feature of Spark is generality. Spark can easily handle data streaming and real-time data processing. It can also easily interact with SQL databases and data frames. The machine learning library of Spark (MLlib) and GraphX, which is the graph analysis tool of this platform, allows users to perform data processing and analysis in a fast and parallelized environment that can be installed either on a single node as a stand-alone version or on thousands of nodes.

D. Functional MRI - Brain Networks

Functional magnetic resonance imaging (fMRI) is a technique that measures brain activity by detecting associated changes in blood flow. This MRI technique uses the changes in magnetization between oxygen-rich and oxygen-poor blood in the brain as its primary outcome measure, with greater consumption of oxygen corresponding to greater neural recruitment within the brain [6] [8] . Our brain is an efficient and precise network. It is a network made up of a large number of brain regions that have their own tasks and functions while remaining highly interactive by continuously sharing information [2] [11]. As such, these regions form a complex integrative system in which information is continuously processed and transferred between structurally and functionally linked brain regions: the brain network [6] [8]. The data that are collected in fMRI modality are four dimensional (4-D), and volumes of images are acquired across time. The preprocessing and analysis of this huge volume of data is always time-consuming and costly, requiring a parallelized infrastructure. Therefore, fMRI data analysis addresses at least 2 Vs of

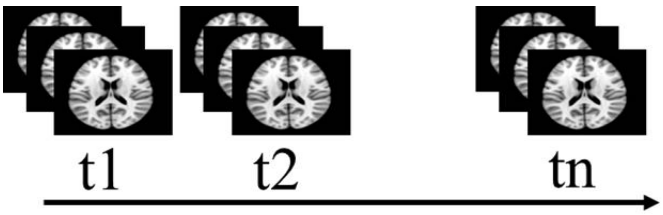

Fig. 2. fMRI data acquisition 3D brain volume across time

big data analytics: Volume and Velocity. In small to large fMRI datasets, giga to tera bytes of data are preprocessed and analyzed. Additionally, each day imaging research and healthcare centers collect more and more data, and archiving this volume of data has become challenging. Data format is another issue causing some restrictions, as formats can be directly stored in a database and preprocessed or analyzed by standard big data analytics tools. To merge big data analytics and medical imaging, we proposed and developed a pipeline that can be used on a single node PC or large clusters and that is able to process data much faster than the current methods. This method also enables users to store the pre/post processed data in a different data format that is compatible with big data plat-forms, especially Spark.

III. Results and Discussion

In this study, we tested our proposed pipeline (Fig.5) against the results from our previous papers [6]. Briefly, in past studies, 7 males and 9 females with a mean age of 21.1 2.2 years were recruited, and structural and functional MRI data was collected. The standard fMRI preprocessing steps were applied to raw data using FMRIB Software Library v5.0 [10]. The goal of this study was to extract the brain networks (especially Default Mode Network) from independent components of brain imaging data. Probabilistic Independent Components of the preprocessed data were calculated by FSL-MELODIC, resulting in 84 components. Next, using our template matching algorithm [6], the DMN was reconstructed from probabilistic independent components. In our brain network extractor and decision-making algorithm, different methods were employed, including normalized cross-correlation, sum of squared differences and dice coefficient. In our current study, we only used the sum of squared errors (SSE) in order to test our Spark solution. In this work, we used the PySpark stand-alone version (http://spark.apache.org/) on a single node to test and explore the potential application of our proposed pipeline. The 84 brain components in the Neuroimaging Informatics Technology Initiative (Nifti) format (standard format for NeuroImgaing data) were loaded into memory using a Nibable package (http://nipy.org/nibabel) providing interfaces for neuroimaging data manipulation in Python. Next, the data in memory was converted to the Resilient Distributed Datasets (RDD) format. RDD is a fundamental data structure of Spark. It is an immutable distributed collection of objects, and each dataset in RDD is divided into logical partitions that may be computed on different nodes of the cluster. Formally, an RDD is a read-only, partitioned collection of records. RDDs can be created through deterministic operations on either data on stable storage or other RDDs. RDD is a fault-tolerant collection of elements that can be operated in parallel. As mentioned above, we were able to extract the default mode network from the components. Therefore, we used the DMN template developed by our team. The template was also loaded and converted to RDD. SSD between 84 components and DMN templates were calculated using the following Equation 1 in PySpark after flatmapping and zipping RDDs.

$$SSD = \sum_{x,y} [f(x,y) - t(x-u, y-v)]^2 \quad (1)$$

The performance was measured in PySpark, which was equal to 6.43799 seconds. The same experiment was repeated in Python. The performance was measured, and it was equal to 23.86625. This testing revealed that using a PySpark big data platform even on a single node runs the data about four times faster than on pure Python (exactly 3.7) in our case. In addition, if the number of images increases, the difference between PySpark and Python performance will increase as well. Fig.3 and Tabel I compare the measured performance between Python and PySpark, which ran against our template-matching script. It expedited the running time roughly fourfold, which is promising. It is important to note that this program was executed on a single node and was not completely designed for parallel computing and a big data platform. In other words, if we develop our serial template matching algorithm for the PySpark environment and parallel processing and also use a high performance cluster instead of a stand-alone, single node version of Spark, the performance will potentially improve by up to 10 to 20 times, especially when a very large dataset is processed. The reconstructed default mode network is demonstrated in Fig.4.

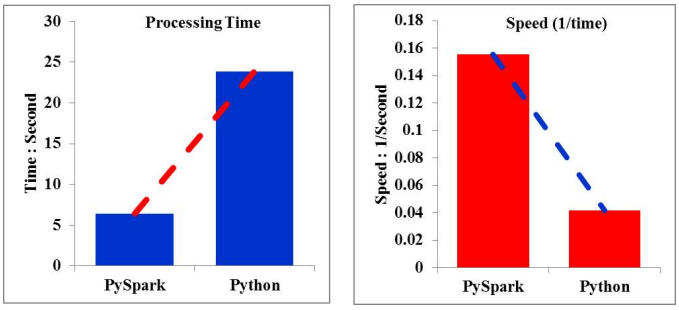

Fig. 3. The processing time and speed comparison shows PySpark-based pipeline performs faster

TABLE I. Comparison Table between PySpark and Python in Brain Extraction Application

Platform	Time (second)	1/Time
PySpark	6.437999964	0.15533
Python	23.86625409	0.0419

IV. Conclusion

We have developed and successfully tested our new PySpark-based pipeline on a single node to analyze functional MRI data for extracting brain networks. The new pipeline improved the processing time, proving itself four times faster than previous works, while accuracy remained at the same value. Furthermore, ease of use, in-memory data processing, and storage results in different data structures are important

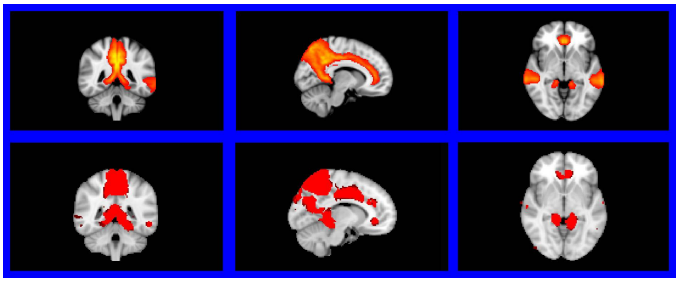

Fig. 4. Using our algorithm described in [7] [9], we reconstructed the brain networks from independent components

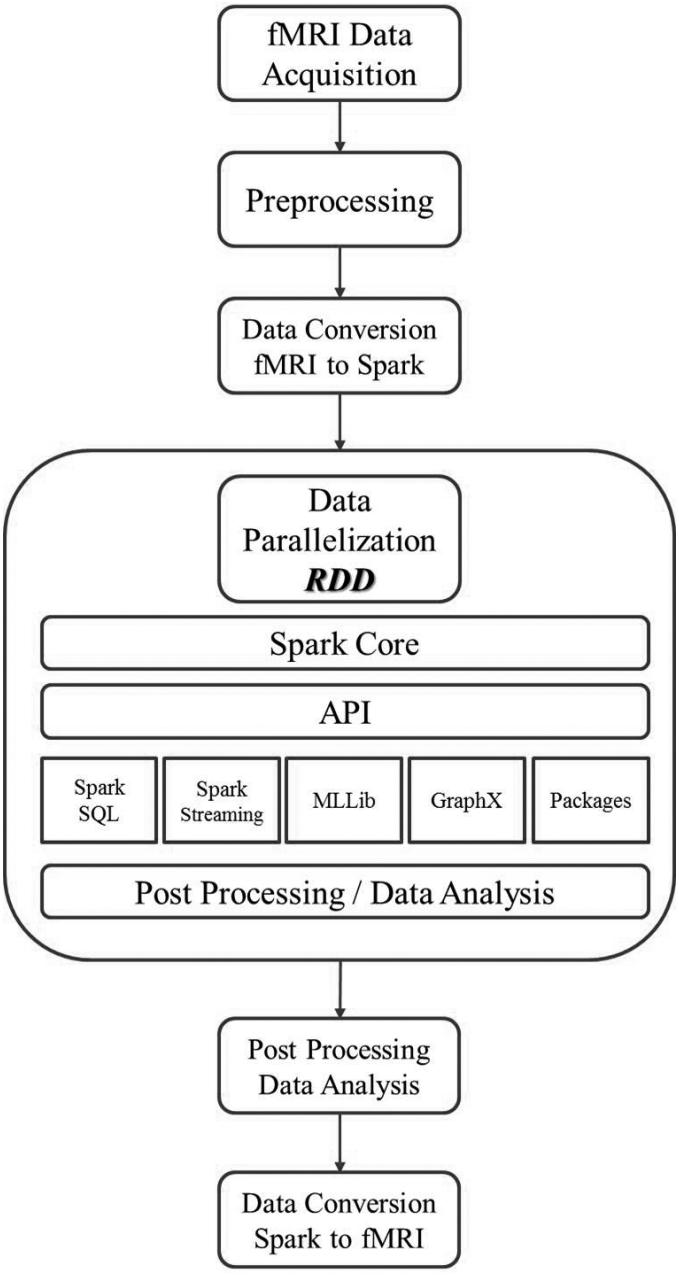

Fig. 5. PySpark-based pipeline for fMRI data processing and analysis

features of this pipeline. Additionally, this pipeline can easily expand to several nodes and high performance computing clusters for massive data analysis on large datasets, which will definitely improve the processing time and the performance of the pipeline much more than a single node.

ACKNOWLEDGMENTS

We would like to express our gratitude to Drs. Ali Mohammad Golestani, post-doctoral fellow at Rotman Research Institute at Baycrest, and Dr. Cristina Saverino, post-doctoral fellow at Toronto Rehabilitation Institute-University Health Network, for extending their help and support in this study.

REFERENCES

[1] J. Andreu-Perez, C. C. Poon, R. D. Merrifield, S. T. Wong, and G.-Z. Yang, "Big data for health," *Biomedical and Health Informatics, IEEE Journal of*, vol. 19, no. 4, pp. 1193–1208, 2015.

[2] C. Grady, S. Sarraf, C. Saverino, and K. Campbell, "Age differences in the functional interactions among the default, frontoparietal control and dorsal attention networks," *Neurobiology of Aging*, 2016.

[3] A. Kala Karun and K. Chitharanjan, "A review on hadoop?hdfs infrastructure extensions," in *Information & Communication Technologies (ICT), 2013 IEEE Conference on*, pp. 132–137, IEEE, 2013.

[4] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, "Big data," *The management revolution. Harvard Bus Rev*, vol. 90, no. 10, pp. 61–67, 2012.

[5] S. Sagiroglu and D. Sinanc, "Big data: A review," in *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pp. 42–47, IEEE, 2013.

[6] S. Sarraf, C. Saverino, H. Ghaderi, and J. Anderson, "Brain network extraction from probabilistic ica using functional magnetic resonance images and advanced template matching techniques," in *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*, pp. 1–6, IEEE, 2014.

[7] S. Sarraf, E. Marzbanrad, and H. Mobedi, "Mathematical modeling for predicting betamethasone profile and burst release from in situ forming systems based on plga," in *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*, pp. 1–6, IEEE, 2014.

[8] S. Sarraf and J. Sun, "Functional brain imaging: A comprehensive survey," *arXiv preprint arXiv:1602.02225*, 2016.

[9] S. M. Smith, "Fast robust automated brain extraction," *Human brain mapping*, vol. 17, no. 3, pp. 143–155, 2002.

[10] S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. Behrens, H. Johansen-Berg, P. R. Bannister, M. De Luca, I. Drobnjak, D. E. Flitney, *et al.*, "Advances in functional and structural mr image analysis and implementation as fsl," *Neuroimage*, vol. 23, pp. S208–S219, 2004.

[11] S. C. Strother, S. Sarraf, and C. Grady, "A hierarchy of cognitive brain networks revealed by multivariate performance metrics," in *Signals, Systems and Computers, 2014 48th Asilomar Conference on*, pp. 603–607, IEEE, 2014.

[12] E. E. Tripoliti, D. I. Fotiadis, and M. Argyropoulou, "A supervised method to assist the diagnosis and classification of the status of alzheimer's disease using data from an fmri experiment," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pp. 4419–4422, IEEE, 2008.

[13] P. Vemuri, D. T. Jones, and C. R. Jack Jr, "Resting state functional mri in alzheimer's disease," *Alzheimer's research & therapy*, vol. 4, no. 1, pp. 1–9, 2012.

[14] M. Viceconti, P. Hunter, and R. Hose, "Big data, big knowledge: Big data for personalized healthcare," *Biomedical and Health Informatics, IEEE Journal of*, vol. 19, no. 4, pp. 1209–1215, 2015.

[15] X. Zhang, B. Hu, X. Ma, and L. Xu, "Resting-state whole-brain functional connectivity networks for mci classification using l2-regularized logistic regression," *NanoBioscience, IEEE Transactions on*, vol. 14, no. 2, pp. 237–247, 2015.