Algorithmic Enhancements to Big Data Computing Frameworks for Medical Image Processing

Shunxing Bao*, Bennett A. Landman (adviser)* and Aniruddha Gokhale (adviser)*
*Dept of EECS, Vanderbilt University, Nashville, TN 37235, USA.
Email: {shunxing.bao,bennett.landman,a.gokhale}@vanderbilt.edu

***Abstract*—Large-scale medical imaging studies to date have predominantly leveraged in-house, laboratory-based or traditional grid computing resources for their computing needs, where the applications often use hierarchical data structures (e.g., NFS file stores) or databases (e.g., COINS, XNAT) for storage and retrieval. The resulting performance for laboratory-base approaches reveal that performance is impeded by standard network switches since they can saturate network bandwidth during transfer from storage to processing nodes for even moderate-sized studies. On the other hand, the grid may be costly to use due to the dedicated resources used to execute the tasks and lack of elasticity. With increasing availability of cloud-based Big Data frameworks, such as Apache Hadoop, cloud-based services for executing medical imaging studies have shown promise. Despite this promise, our preliminary studies have revealed that existing Big Data frameworks illustrate different performance limitations for medical imaging applications, which calls for new algorithms that optimize their performance and suitability for medical imaging. For instance, Apache HBase's load distribution strategy of region split and merge is detrimental to the hierarchical organization of imaging data (e.g., project, subject, session, scan, slice). To address these challenges, this doctoral research is developing a range of performance optimization algorithms. This paper describes preliminary research we have conducted in this realm and presents a list of research tasks that will be undertaken as part of this doctoral research.**

***Keywords*—Hadoop, HBase, Medical imaging, Grid computing.**

I. PROBLEM DEFINITION AND RESEARCH CHALLENGES

A. Motivation

Big data in medical imaging offer an opportunity to study specific control populations (age / sex / demographics / genetics) and identify substantive homogeneous sub-cohorts so that one may understand the role that individual factors play in treatment response. To support this cause, existing medical imaging studies have often leveraged either in-house, laboratory-sized computing resources or grid computing resources. Both these existing approaches, however, incur various limitations. Consider, for example, a medical imaging application that converts Digital Imaging and Communications in Medicine (DICOM) files to NiFTI (a research file format). If converting a 50 MB volume takes 15 seconds, an ideal gigabit network (about 100 MB/s) saturates with slightly less than 30 simultaneous processes.

These constraints are significant for medical imaging applications. For instance, consider how contemporary grid computing approaches separate data storage from computation. To analyze data, each dataset must be copied from a storage archive, submitted to an execution node, processed, synthesized to a result, and results returned to a storage archive. When imaging datasets become massive, which is clearly the trend, the bottleneck associated with copying and ensuring consistency overwhelms the benefits of increasing the number of computational nodes, i.e., performance gains no longer scale with the number of computational nodes, but are limited by the network.

The network issues are not the only challenges. Despite the presence of vast repositories of magnetic resonance imaging (MRI) and computed tomography (CT) images, which are accumulating at a rate of nearly 100 million examinations per year in the U.S., today we lack the image processing, statistical, and informatics tools for large-scale analysis and integration with other clinical information (e.g., genetics and medical histories). What is needed is an efficient mechanism for query, retrieval, and analysis of all patient data (including imaging) which would enable clinicians, statisticians, image scientists, and engineers to better design, optimize, and translate systems for personalized care into practice. All of this must be affordable and performs well. These needs preclude both in-house and grid-based approaches.

Consequently, cloud-based services to address these growing needs hold promise. Specifically, Big Data processing frameworks such as Apache Hadoop ecosystem are promising in this context. The Hadoop framework provides flexible, distributed, scalable and fault tolerant storage and parallel computational modules, and the associated HBase, which is a NoSQL database built atop Hadoop's distributed file system, has the potential for building up a big data platform for medical imaging.

B. Problem Definition

Although Big Data frameworks like Hadoop have been applied in online commerce, social media, video streaming, high-energy physics, and proprietary corporate applications, many existing approaches attempting to use these for medical imaging have focused on general machine learning literature, and seek to implement algorithms specifically designed to take advantage of big data architecture [5], [8], [19], exploit the MapReduce framework to sift through datasets [14], or use the distributed file system [18], [20]. While such approaches have been effective for genetics studies [7], [20], they have not yet proven effective within current medical image computing workflows and have not been widely integrated with medical imaging data formats (e.g., DICOM) for medical image processing.

978-1-5090-5817-4/17 $31.00 © 2017 IEEE
DOI 10.1109/IC2E.2017.47

The key issue is that substantial resources have been invested in creating existing algorithms, software tools, and pipelines, and there is a substantive (often prohibitive) cost associated with algorithm re-implementation and re-design specifically for use in big data frameworks. Yet, at the same time, it is increasingly becoming important to leverage Big Data frameworks hosted in the cloud to solve medical imaging problems. To address these requirements, new research is needed, and the different dimensions of the research problems we solve form the contours of this doctoral dissertation. Specifically, we aim to address the following questions:

1) Are Big Data frameworks like Apache Hadoop suitable for Big Data medical imaging?
2) What enhancements and performance optimizations are needed for the Apache Hadoop Ecosystem to be used in Big Data medical image processing?
3) Are there theoretical bounds that inform the user under what conditions cloud-based Big Data frameworks should or should not be used for medical imaging applications?
4) Can the Apache Hadoop Ecosystem be blended with distributed machine learning algorithms to re-implement and re-design existing medical image computing workflows?

The rest of this paper describes our current work in addressing these research questions and proposed tasks along with a completion timeline.

II. ONGOING AND PROPOSED APPROACH

A. Task 1: Apache HBase Optimizations

The task of processing medical images at scale requires a distributed image processing architecture that is aware of the underlying hierarchical imaging and meta-data. Our system is based upon the Hadoop framework. We combine Hadoop with Apache HBase, a NoSQL database which implements Google's BigTable [6], [10]. The specific contribution of this work [3] is a novel data storage mechanism that uses the hierarchical structure of imaging studies to co-locate data to physical machines. This proposed co-location strategy provides an efficient processing environment in which data do not need to be transferred between machines, thus avoiding network overhead and saturation. Herein, we develop and demonstrate a new data model for use with distributed storage and computation systems that provides practical access to distributed imaging archives, integrates with existing data workflows, and effectively functions with commodity hardware. The proposed Hadoop system was implemented on a production lab-cluster alongside a standard Sun Grid Engine(SGE) [9]. Our experiments promote a general framework for medical imaging processing (e.g., structured data retrieval, access to locally installed binary executables/system resources, structured data storage) without comingling idiosyncratic issues related to image processing (e.g., parameter settings for local tissue models, smoothing kernels for denoising, options for image registration).

1) Putting the Pieces Together: Figure 1 presents the overall structure of our modified Apache Hadoop ecosystem focusing particularly on the HBase modifications. HBase resides upon HDFS. Zookeeper monitors the health status of RegionServer. When users create a HBase table, they need to pinpoint the RegionSplitPolicy to HMaster, and the pre-set split policy is automatically triggered once when a Table region needs to be split. Our custom split policy is made the default split policy. The input DICOM image is de-identified (for privacy preservation purposes) and is normalized to the hierarchy structure by a local row key generator before storing into HBase. XNAT is an open-source imaging informatics software platform dedicated to helping you perform imaging-based research (https://www.xnat.org/).

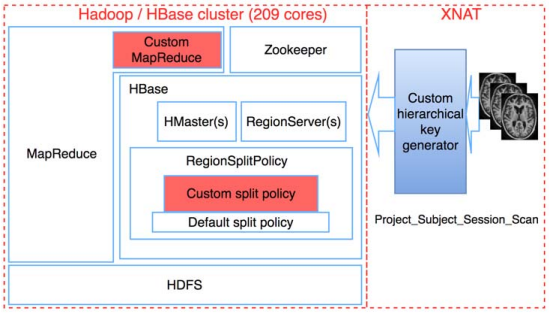

Fig. 1. Overall Structure of Hadoop / HBase / Zookeeper Cluster with the Proposed Custom Row Key and Region Split Policy

2) Methods: To investigate the performance of our HBase schemas, we characterized standard DICOM to NiFTI file format conversion using three test scenarios using HBase and Hadoop and one with Network Attached Storage (NAS). First, we test our complete design with our proposed row key and RegionSplitPolicy (Scenario: "Custom Key / Custom Split HBase"). Second, we test our proposed row key with the default Re- gionSplitPolicy (Scenario: "Custom Key / Default Split HBase"). Third, we test using a random key, and MD5 hash of the data, as the key in HBase (Scenario: "Naïve HBase"). The MD5 hash was used in placed of the GUID because GUID's were re- moved during the de-identification process associated with data retrieval. With this comparison, we test the native abilities of Hadoop and HBase without any of our pro- posed advances. Fourth, we use a traditional Sun Grid Engine to distribute portable bash script (PBS) jobs to computational nodes accessing data from a Network Attached Storage (NAS) device (Scenario: "Grid Engine NAS").

3) Results: This row key architecture improves throughput by 60% and reduces latency by 577% over the naïve approach. The custom split policy strongly enforces data collocation to further increase throughput by 21% and reduce latency by 29%. In summary, colocation of imaging data drastically improves processing time and the proposed approach illustrates how Hadoop and HBase can be translated to typical medical image processing tasks. All software has been made publicly available.

B. Task2: Establishing Theoretical Bounds

With the innovations in Task 1, Apache Hadoop and HBase can readily be deployed as a service using commodity networks to address the needs of high throughput medical image computing. However, the benefits of using such a framework must be formally evaluated against a traditional approach to characterize the point at which simply "large scale" processing

transitions into "big data" and necessitates alternative computational frameworks. This creates new questions, e.g., when does this novel Hadoop/HBase framework perform better than traditional high performance computing clusters like SGE? In this case, there are many parameters of concern, such as the cluster size, machine cores, node memory, distribution of resources, image processing job, etc. [2], [15]–[17]. The state of the art for theoretical models to characterize the performance of SGE and Hadoop and empirical verification of the models has been defined in our recent accepted work [4].

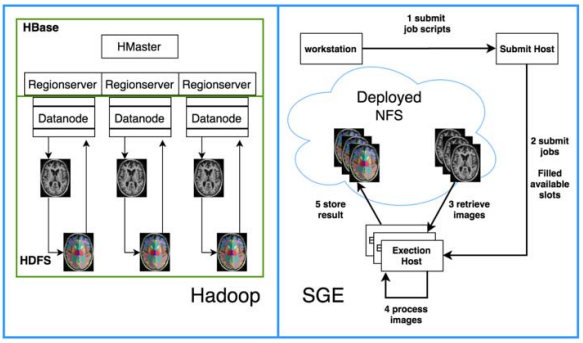

Fig. 2. Hadoop and SGE data retrieval, processing and storage working flow basing on Multi-atlas CRUISE (MaCRUISE) segmentation [12], [13]. The data in an HBase table is approximately balanced to each Regionserver. The Regionserver collocates with a Hadoop Datanode to fully utilize the data collocation and locality [1]. We design our proposed computation models using only the map phase of Hadoop's MapReduce. Within the map phase, all necessary data is retrieved and saved on a local directory and gets furtherly processed by locally installed binary executables command-line program. After that, the results of processing are uploaded back to HBase. For SGE, the user submits a batch of jobs to a submit host, and this host dispatches the job to execution hosts. Each execution host retrieves the data within a shared NFS and stores the result back to the NFS.

1) Theoretical Model: Hadoop and HBase enable co-location of data storage and computation while minimizing data transfer, while SGE separates data storage from computation. Figure 2 summarizes the workflows for both methods, where the goal for the theoretical models is to illustrate the workflow for both scenarios.

The theoretical models have two parts. The first is wall-clock time, which represents the total time as experienced by the user. The second part is resource time, which measures elapsed time on each node when a process starts across all nodes. In [4], the hypothesis we have been made is dispatching a huge amount of jobs with similar type, input, output and running time to a cluster with fixed network bandwidth (1 Gigabit Ethernet). We primarily discuss the relationship between average bandwidth that is shared for each job and cluster's bandwidth when there is network saturation. We also discuss how data/core allocation for Hadoop may affect the models (Cluster with SGE do not care where the data is).

2) Identifying the Limits of Apache Hadoop for Medical Imaging: In initial demonstration pilot verification, empirical results match the theoretical models. Figure 3 is a conclusion of the usage of theoretical models, which helps us understand what kind of jobs is suitable for a given cluster. In Figure 3, the common logarithm ratio (log10) for 5000 jobs' (with different job datasets with different processing time) performance transition is shown on ideal/real cluster setup. The ratio stands for Hadoop's wall-clock/resource time divides SGE's time (log10(SGE_TIME/HADOOP_TIME). If ratio close to zero, Hadoop's performance is similar with sge's. And if ratio is smaller than 0, it represents that Hadoop performs better, vise versa.), and the log10 ratio is get scales in range from [-1,1]. As Figure 3(A/B) represents the ratio when core/data allocation are balanced (the ideal scenario for Hadoop), which are based on an assumed lab-based cluster. The red lines in Figure 3(C/D) indicate the parameters for which Hadoop and SGE result in equivalent performance for the specified setup (data in both scenarios is approximately balanced; 72 cores are balanced on 6 machines. 209 cores is a cluster of unbalanced core allocation).

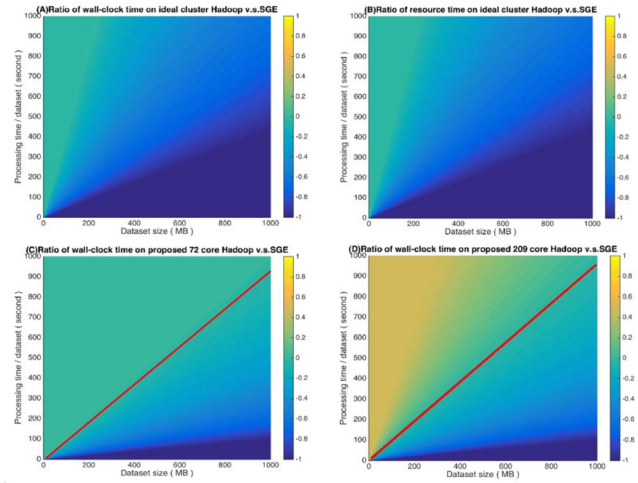

Fig. 3. Time ratio of Hadoop v.s. SGE based on core / data balanced and unbalanced cluster. The balancing can only affect the total job running time. (A) Ratio of total running time on a data / core balanced cluster. (B) Ratio of resource time either on a core / data balanced cluster. (C) Ratio of total running time on a proposed 72 cores grid setup (approximate balanced cluster). (D) Ratio of total running time on a proposed 209 cores grid setup (core / data unbalanced cluster).

C. Task3: Enhancing Big Data Frameworks for Heterogeneous Mix of Medical Imaging Applications

1) In progress: The Vanderbilt University Institute for Imaging Science (VUIIS) Center for Computational Imaging (CCI) has developed a database built on XNAT housing over a quarter of a million scans. The database provides framework for (1) rapid prototyping, (2) large scale batch processing of images and (3) scalable project management. The system uses the web-based interfaces of XNAT and REDCap to allow for graphical interaction [11]. The system resources are hosted on the Advanced Computing Center for Research and Education (ACCRE) at Vanderbilt University, which is deployed with 10 Gigabit Ethernet and over 6,000 cores.

We collect anonymous completed job statistics of the XNAT system for the past 4 years to analyze and filter out the IO-intensive jobs that could potentially benefit from the big data frameworks. The job lengths range from 15 second to 9 days, and the input image size of jobs and output images also varies. Furthermore, we design simulation models of traditional cluster and Hadoop big data framework to schedule mix types of jobs (IO-intensive and compute-intensve) with actual time

order according to history record and find the running time of all the jobs. The simulation scales the size of cluster cores from 10 to 6000, and using two kinds of network bandwidth scenarios (1 Gigabit and 10 Gigabit Ethernet).

Meanwhile, according to the result from Task 2, a Region-server hypervisor is designed in for better allocating HBase data according to the core distribution of the cluster of each machine. We assume that in this way, better data locality could be achieved resulting in increased processing throughput.

2) Opportunities for Optimizations: (1) In HBase, Memstore is used to buffer new data that is written into database, Blockcache is used to cache the most recent read data. Our Current framework use default Memstore and Blockcache size configuration, a trade off analysis for reducing the overhead of Hard Disc data read/write is worth working. (2) Training system access model can predictive data collocation strategies rather than static hierarchical data group structure. The trained model can also design efficient query models to reduce data retrieval latency.

D. Task 4: Blending Big Data Frameworks and Machine Learning

This final task is still being investigated and a concrete research direction is not yet well-articulated. A potential approach concerns designing a distributed machine learning (e.g. Spark) related framework for rapidly diagnosing the correctness of algorithm or image processing pipeline. We seek answers to the following questions: Can current distributed machine learning work in medical image processing? What algorithms are suitable? Is this just faster? Is this scalable on cheap hardware? For instance, we need to classify two eye diseases with multiple custom classifiers containing different feature sets. We want to know if there is a way to know which classifiers are wrong using any distributed machine learning algorithm. Another example is designing a fitting pipeline for modeling images. Rather than processing millions of images for a couple of days or even months and finally concluding that the the models are wrong, it would be preferable to have a system that can perform incremental learning, validate the intermediate results, and pinpoint wrong models in real time.

III. Timeline

Figure 4 provides a timeline for this research being conducted under the supervision of Dr. Bennett A. Landman, Associate Professor, Vanderbilt University and Dr. Aniruddha Gokhale, Associate Professor, Vanderbilt University.

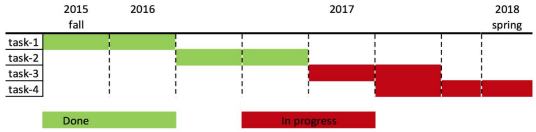

Fig. 4. Research timeline

References

[1] Apache HBase Team. *Apache hbase reference guide*. Apache, version 2.0.0 edition, Apr. 2016.

[2] R. Appuswamy, C. Gkantsidis, D. Narayanan, O. Hodson, and A. Rowstron. Scale-up vs scale-out for hadoop: Time to rethink? In *Proceedings of the 4th annual Symposium on Cloud Computing*, page 20. ACM, 2013.

[3] S. Bao, A. Plassard, B. A. Landman, and A. Gokhale. Cloud Engineering Principles and Technology Enablers for Medical Image Processing-as-a-Service. In *IEEE International Conference on Cloud Engineering (IC2E)*, Vancouver, Canada, Apr. 2017. IEEE.

[4] S. Bao, F. D. Weitendorf, A. J. Plassard, H. Yuankai, A. Gokhale, and L. A. Bennett. Theoretical and empirical comparison of big data image processing with apache hadoop and sun grid engine. SPIE Medical Imaging, 2017(accepted).

[5] T. Bednarz, D. Wang, Y. Arzhaeva, R. Lagerstrom, P. Vallotton, N. Burdett, A. Khassapov, P. Szul, S. Chen, C. Sun, et al. Cloud Based Toolbox for Image Analysis, Processing and Reconstruction Tasks. In *Signal and Image Analysis for Biomedical and Life Sciences*, pages 191–205. Springer, 2015.

[6] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A Distributed Storage System for Structured Data. *ACM Transactions on Computer Systems (TOCS)*, 26(2):4, 2008.

[7] S. Chen, T. Bednarz, P. Szul, D. Wang, Y. Arzhaeva, N. Burdett, A. Khassapov, J. Zic, S. Nepal, T. Gurevey, et al. Galaxy+ Hadoop: Toward a Collaborative and Scalable Image Processing Toolbox in Cloud. In *Service-Oriented Computing–ICSOC 2013 Workshops*, pages 339–351. Springer, 2013.

[8] J. Freeman, N. Vladimirov, T. Kawashima, Y. Mu, N. J. Sofroniew, D. V. Bennett, J. Rosen, C.-T. Yang, L. L. Looger, and M. B. Ahrens. Mapping Brain Activity at Scale with Cluster Computing. *Nature methods*, 11(9):941–950, 2014.

[9] W. Gentzsch. Sun grid engine: Towards creating a compute power grid. In *Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on*, pages 35–36. IEEE, 2001.

[10] S. Ghemawat, H. Gobioff, and S.-T. Leung. The Google File System. In *ACM SIGOPS operating systems review*, volume 37, pages 29–43. ACM, 2003.

[11] R. L. Harrigan, B. C. Yvernault, B. D. Boyd, S. M. Damon, K. D. Gibney, B. N. Conrad, N. S. Phillips, B. P. Rogers, Y. Gao, and B. A. Landman. Vanderbilt university institute of imaging science center for computational imaging xnat: A multimodal data archive and processing environment. *NeuroImage*, 124:1097–1101, 2016.

[12] Y. Huo, A. Carass, S. M. Resnick, D. L. Pham, J. L. Prince, and B. A. Landman. Combining multi-atlas segmentation with brain surface estimation. In *SPIE Medical Imaging*, pages 97840E–97840E. International Society for Optics and Photonics, 2016.

[13] Y. Huo, A. J. Plassard, A. Carass, S. M. Resnick, D. L. Pham, J. L. Prince, and B. A. Landman. Consistent cortical reconstruction and multi-atlas brain segmentation. *NeuroImage*, 138:197–210, 2016.

[14] D. Markonis, R. Schaer, I. Eggel, H. Müller, and A. Depeursinge. Using MapReduce for Large-scale Medical Image Analysis. *arXiv preprint arXiv:1510.06937*, 2015.

[15] E. Medernach. Workload analysis of a cluster in a grid environment. In *Workshop on Job Scheduling Strategies for Parallel Processing*, pages 36–61. Springer, 2005.

[16] A. Rosset, L. Spadola, and O. Ratib. Osirix: an open-source software for navigating in multidimensional dicom images. *Journal of digital imaging*, 17(3):205–216, 2004.

[17] N. Sadashiv and S. D. Kumar. Cluster, grid and cloud computing: A detailed comparison. In *Computer Science & Education (ICCSE), 2011 6th International Conference on*, pages 477–482. IEEE, 2011.

[18] T. S. Soares, M. A. Dantas, D. D. De Macedo, and M. A. Bauer. A Data Management in a Private Cloud Storage Environment Utilizing High Performance Distributed File Systems. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2013 IEEE 22nd International Workshop on*, pages 158–163. IEEE, 2013.

[19] B. Tripathy and D. Mittal. Hadoop based Uncertain Possibilistic Kernelized C-means Algorithms for Image Segmentation and a Comparative Analysis. *Applied Soft Computing*, 2016.

[20] C.-T. Yang, W.-C. Shih, L.-T. Chen, C.-T. Kuo, F.-C. Jiang, and F.-Y. Leu. Accessing Medical Image File with Co-allocation HDFS in Cloud. *Future Generation Computer Systems*, 43:61–73, 2015.