

Architecture of distributed picture archiving and communication systems for storing and processing high resolution medical images

Victoria Tokareva^{1,*}

¹Laboratory of Nuclear Problems, Joint Institute for Nuclear Research,
6 Joliot-Curie, Dubna, Moscow region, 141980, Russia

Abstract. New generation medicine demands a better quality of analysis increasing the amount of data collected during checkups, and simultaneously decreasing the invasiveness of a procedure. Thus it becomes urgent not only to develop advanced modern hardware, but also to implement special software infrastructure for using it in everyday clinical practice, so-called Picture Archiving and Communication Systems (PACS). Developing distributed PACS is a challenging task for nowadays medical informatics.

The paper discusses the architecture of distributed PACS server for processing large high-quality medical images, with respect to technical specifications of modern medical imaging hardware, as well as international standards in medical imaging software. The MapReduce paradigm is proposed for image reconstruction by server, and the details of utilizing the Hadoop framework for this task are being discussed in order to provide the design of distributed PACS as ergonomic and adapted to the needs of end users as possible.

1 Introduction

Medical imaging (MI) is a common name representing several technologies (CT, PET, MPI, ultrasound, etc.), which are used to produce images of internal aspect of a human body for the purposes of diagnosis, monitoring and treatment. Also MI is employed for achieving scientific goals in anatomy and physiology by providing a broad database of examinations, which is helpful for the identification of abnormalities. The key benefit of MI is its relative noninvasiveness, i.e. in most of MI techniques no instruments are introduced inside a patient's body to produce images.

During MI we observe a signal (for example, attenuated X-rays) and make a conclusion about the properties of tissues and organs. So, in technical sense, obtaining medical images from the raw data means searching for the solution of mathematical inverse problems.

MARS (Medipix All Resolution System) micro-CT scanner [1], investigated at DLNP JINR, allows obtaining high quality medical images, resulting in large volumes of both raw data and reconstructed images. In order to improve speed and quality of processing, storing, and transfer of medical images for the project, we performed the investigation of relevant standards and industry solutions in the field of big medical data processing, in particular, employing distributed computational systems.

*e-mail: tokareva@jinr.ru

2 DICOM standard and PACS systems

In the last few years, DICOM [2] (Digital Imaging and Communications in Medicine), the industry standard by NEMA (National Electrical Manufacturers Association, USA) for creating, storing, transferring and visualizing medical images and documents of patients, has de facto become a world-wide used protocol, that defines a single storage format of medical images and a single protocol for their transmission.

Medical institutions possess systems that control DICOM files and these systems are called Picture archiving and communication systems (PACS). PACS are now designed to handle medical images stored at each hospital or facility.

One of the most striking trends in modern Russian medicine is the widespread informatization, and one of its consequences is a request to ensure that medical data, stored distributedly over various hospitals, at the same time would be available to physicians of a regional network in order to support collaboration in the work of medical institutions. Thus, in the near future several PACS existing in different hospitals or institutions should be able to work together. So, there is a new challenge to develop “meta-PACS” capable of storing and processing region-wide or even nation-wide corpus of medical imaging data.

3 Computed tomography

Both availability and resolution of medical imaging devices are constantly improving, in particular, in the area of computed tomography (CT), which is one of the most popular medical imaging methods. CT is a technique for cross-sections imaging of an object using a series of X-ray measurements taken from different angles around the object [3].

Over the last years, the following trends are being observed in this field.

On the one hand, the improvements in medical image acquisition systems resulted in much better pixel resolutions of detectors and, consequently, in the increase of the amount of medical image data. In recent decades a typical size of a scan has been grown from Kilo- to Gigabytes and is expected to reach Terabytes soon [4]. For example, the new Sky Scan 2011 tomograph with the resolution of 200 nm per pixel could provide 64 Megabyte (MB) per slice.

On the other hand, the popularity of CT as a means of diagnostics is constantly growing: while up until 2010, 5 billion medical imaging studies had been conducted worldwide [5], now more than one billion diagnostic images per year are taken only in the USA, and this number is expected to keep growing [6].

Thus, big data in medical imaging occurs in two ways: first, a large data size from a single scan, and second, thousands and millions of images stored in a PACS server. In practice, both ways multiply.

Not only the increased volumes of data, but also complicated image reconstruction algorithms provide challenges to the future PACS development.

The reconstruction algorithms can be split in two categories. Reconstructing images using *iterative algorithms* could take hours for large scans. *Back-projection algorithms* have computational complexity of $O(n^4)$ [3], while hierarchic back-projection has one of $O(\log(n) \times n^3)$ [7].

At DLNP JINR we investigate the MARS micro-CT scanner [1], employing Medipix electronics based on GaAs semiconductor detectors. It allows obtaining high quality medical images while decreasing the irradiation of an examined object compared to currently used scanner models. Scanner design makes it possible to take up to 720 shadow projections of an object with a spatial resolution up to $55 \mu\text{m}^2$. Such accuracy helps us to obtain high quality resulting images through the reconstruction employing the FDK [8] (Feldkamp-Davis-Kress, 1984) method. The opposite side of such accuracy is

a significant growth of both raw data and reconstructed samples, and consequently, slower image processing and network access compared to samples of a smaller size. The solution to these challenges could be a distributed storage and processing data in PACS.

4 Map-reduce based distributed PACS

A popular approach for the implementation of distributed PACS is employing the MapReduce paradigm [9]. The MapReduce is a scheme of distributed computations, which is widely applicable for parallel calculations with big (up to hundreds of Petabytes) data amounts in computing clusters (see Figure 1).

The MapReduce workflow involves two steps: Map and Reduce (called so in reference to high-level functions with the same names, map and reduce). The preprocessing of input data takes place at the Map stage. In order to do this, one of computers in the cluster (so-called NameNode) fetches task input, then divides it into parts and passes to other computers in the network (so-called DataNodes) for preliminary processing. At the Reduce stage the preprocessed data are converged. Then the NameNode receives callbacks from the DataNodes and formulates output on their base. This output is considered to be the solution of the original problem.

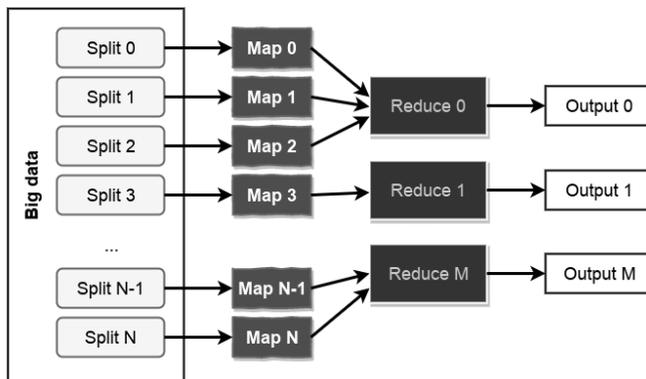


Figure 1. Scheme of MapReduce workflow.

The advantage of MapReduce is based on the fact that it supports distributed execution of preprocessing and data convergence. Although this process happens to be less effective in comparison with more sequential algorithms, the good feature is that MapReduce could be applied to analyze very large amounts of data at several servers simultaneously. Parallel processing also provides us with several ways to recover after node failures and makes our system more fault-tolerant.

Hadoop is a project by the Apache Software Foundation. It involves a set of freely-distributed utilities and libraries, and a framework for development and implementation of distributed programs within the scope of the MapReduce paradigm. Hadoop was developed for running on computing clusters consisting of up to hundreds and thousands of nodes.

So, the implementation of a distributed PACS employing Hadoop framework helps us to benefit from system scalability, cost effectiveness, data replication and a good fault-tolerance level. But, on the other hand, for small clusters image retrieval time grows significantly, so using such Hadoop-based PACS should be discussed for any specific case.

An interesting Hadoop-based PACS implementation was proposed at [10]. Employing MapReduce in a cloud-computing environment authors has implemented scalable CT image reconstruction for cone-beam geometry in high-speed way.

The implementation accelerates the cone-beam FDK reconstruction algorithm by porting it to a MapReduce model. Here authors apply the map functions for filtering and back-projection. Then reduced functions are used to gather parts of back-projected images in full volume. Due to that, the reconstruction time speeded up drastically, so it was found to be in linear correlation with the number of DataNodes employed.

In the study [11], the authors propose an architecture of Medical Image File Access System (MI-FAS) integrating several levels of software platforms: from virtualization infrastructure (OpenNebula+Xen) to virtualization of the fault-tolerance system, then to the Hadoop distributed file system (HDFS), and, finally, several DICOM and Hadoop APIs with user-end web interface. One of the main benefits of this system is its focus on the fault tolerance.

A natural approach to solve the problem of full volume reconstruction is to decompose the main problem in a sequence of slice reconstruction tasks. But in the case of cone-beam geometry such tasks are not completely data-independent, so decomposition at this level does not provide cost-effective results within the framework of the MapReduce paradigm.

The more convenient approach in order to gain the benefits of MapReduce-based distributed calculations is employing a lower-level decomposition with smaller tasks like processing a readout of a single pixel string of the detector at a particular angle, consisting of its Fourier transform, filtering and back-projection.

Thus, a MapReduce approach would be effective only when the amount of data exceeds a particular threshold, depending on overheads for creating a subtask in a particular system implementation, and is expected to become more effective while the amount of data grows.

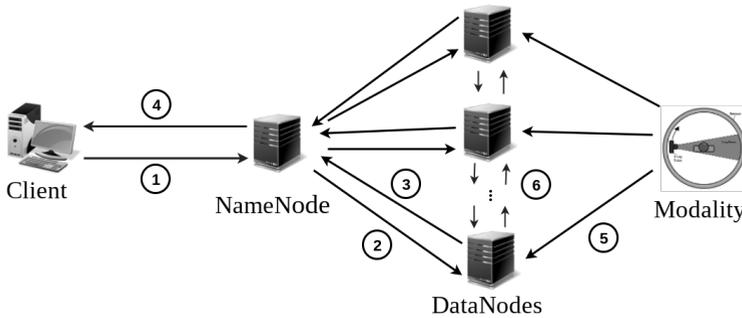


Figure 2. Flows of information in a distributed PACS server. Image reconstruction 1–4: 1) Initial request; 2) Requests to perform the Map step; 3) Gathering data packages for the Reduce step; 4) Providing a final result. Raw data information flow 5–6: 5) Raw data flow from modality; 6) Data exchange between server nodes.

Observing the investigations given before, we propose for our distributed PACS the architecture, shown in Figure 2. Information flow for a reconstruction or data-obtaining request can be described in the following way. A client sends a request to a NameNode of the distributed PACS server. After processing the request, the NameNode sends pieces of code that would process data on DataNodes. The DataNodes send results of data processing back to the NameNode, where the whole reconstruction is gathered using the Reduce algorithm. Optionally, a reconstructed image can be saved in a dedicated

file storage connected to the NameNode. Finally, NameNode provides a final result to the client via user API.

The information flow of the raw data could be given as follows. A PACS modality (e.g. a CT scanner) sends the obtained raw data to nodes of the distributed PACS server. Server nodes exchange data according to built-in algorithms of the distributed file system, e.g. HDFS [12].

5 Conclusion

Big data is going to play an essential role in future medicine. The availability of medical devices and their resolution are growing rapidly, resulting in a fast increase of the size of medical data, which requests innovative solutions to process these data in reasonable timescales. One of these solutions could be new-generation PACS using distributed data storage and processing.

The MapReduce programming model could be especially useful for medical imaging, since this paradigm implies moving code to data instead of vice versa, and thus eliminates the necessity to transfer large data volumes through the storage system. Besides, it was designed for being deployed on inexpensive clusters using commodity hardware. The practical implementation of distributed PACS based on the MapReduce model could be done using recent developments in open-source software, namely, the Hadoop project and the associated frameworks. Such systems could provide means for storing Terabytes and Petabytes of medical imaging data and for parallel fault-tolerant processing of these data. These tools will form the core of future medical informatics software as the quantity and quality of medical data are continuously increasing.

References

- [1] R. Zainon, A.P.H. Butler *et al.*, *Internetworking Indonesia Journal* **2(1)**, 3–10 (2010).
- [2] DICOM standard official site, available at <http://dicom.nema.org/standard.html>.
- [3] H. Turbell, *Cone-beam reconstruction using filtered back-projection*, Linköping University Electronic Press, 189 (2001).
- [4] I. Scholl, T. Aach, T.M. Deserno *et al.*, *Comput. Sci. Res. Dev.* **26**, 5–13 (2011).
- [5] C.A. Roobottom, G. Mitchell, G. Morgan-Hughes, *Clin. Radiol.* **65 (11)**, 859–67 (2010).
- [6] T. Hiroyasu, Y. Minamitani, M. Yoshimi, M. Miki, *WORLD COMP'11 2011-MRS-84 (5)* (2011).
- [7] S. Xiao, Y. Bresler, D.C. Munson, *Proceedings 2003 International Conference on Image Processing* **2**, II–819–22 (2003).
- [8] L. Feldkamp, L. Davis, J. Kress, *Journal of the Optical Society of America A* **1(6)**, 612–619 (1984).
- [9] A. Belle, *BioMed Research International* **2015**, 16 (2015).
- [10] B. Meng, G. Prax, L. Xing, *Med. Phys.* **38(12)** 6603–6609 (2011).
- [11] C.-T. Yang, C.-T. Kuo, W.-H. Hsu, W.-C. Shih, *Proceedings of the 7th international conference on Advances in Grid and Pervasive Computing* **7296**, 338–349 (2012).
- [12] HDFS documentation, available at <http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>.