A Data Management in a Private Cloud Storage Environment Utilizing High Performance Distributed File Systems

Tiago S. Soares*, M.A.R Dantas†, Douglas D.J. de Macedo‡, Michael A. Bauer§
*†*Informatics and Statistic Department (INE)*
‡*Post-Graduate Program of Knowledge Engineering and Management*
Federal University of Santa Catarina (UFSC)
Florianópolis, Brazil
§ *Computer Science Department (CSD)*
University of Western Ontario (UWO)
London, Canada
steinmetz@telemedicina.ufsc.br, {mario,macedo@inf.ufsc.br}, bauer@uwo.ca

***Abstract*—The new trend in the process of data-intensive management indicates the importance of a distributed file system for both Internet large scale services and cloud computing environments. I/O latency and application buffering sizes are two of a number of issues that are essential to be analysed on different class of distributed file systems. In this paper, it is presented a research work comparing four different high performance distributed file systems. Those systems were employed to support a medical image server application in a private storage environment. Experimental results highlight the importance of an appropriate distributed file system to provide a differential level of performance considering application specific characteristics.**

***Keywords*-Distributed File System; DICOM; HDF5; Telemedicine;**

I. Introduction

File systems for large distributed configurations (e.g. cluster, grid and cloud environments) can be broadly classified into two main different categories. In the first category, it is possible to find those designed for Internet services, examples are Google File System (GFS)[1], Amazon Simple Storage Service (S3)[2] and Hadoop File System (HDFS)[3]. On the other hand, examples of file systems targeting high performance computing are the Ceph[4], Fraunhofer File System (FhGFS)[5], Lustre[6], IBM General Parallel File System (GPFS)[7] and Parallel Virtual File System (PVFS)[8]. The second category of those files systems have the goal to support application intensive applications that usually execute in parallel on large configurations. In addition, two other characteristics of those applications are high level of scalability and necessary concurrent storage I/O.

It is recognized that areas candidate to utilize these distributed file systems approaches are applications from finance, oil & gas, large web projects, classic high performance computing challenges and medical applications. In the medicine segment an interesting problem to be tackled is the telemedicine applications, which appeared in early 60's. This paradigm is considered as an ancillary method to provide healthcare access to people that are geographically isolated or cannot have access to a specialized medical system. However, a number of challenges exist to be circumvented in order to allow the continuous growing of the telemedicine utilization. Due to the large amount of data involved in this area, new proposals should be able to provide both high levels of scalability and also high performance computing approaches.

Medical databases are normally composed by text, image and video, thus the order of magnitude of a real-world application - such as a Radiological Information System (RIS) or a Multimedia Telemedicine Network that integrates image and signal data with textual and structured patient information - can reach several terabytes. An example is the statewide telemedicine network *Rede Catarinense de Telemedicina* (RCTM)[9], from the Santa Catarina State, in Brazil. This project presently integrates healthcare facilities of 286 municipalities with a centralized Telemedicine Portal that offers services such as an Electronic Patient Record and a Picture Archiving and Communication Systems (PACS). The RCTM project provides a number of services. Local hospitals from different cities are able to provide exams such as: electrocardiograms, magnetic resonance, computed tomography and angiography. After performing theses examinations, the information is sent online as DICOM (Digital Imaging Communications in Medicine) images to a specially developed PACS, the *CyclopsDCMServer*.

In this paper we present experiments with four different distributed file systems, considering an enhancement of the *CyclopsDCMServer* server architecture, called as PH5WRAP[10], which was designed and implemented to improve the reading and writing functions in parallel (or sequential) of the binary part of the data. This contribution is differential in showing this component supported by distinct high performance distributed file systems.

The paper is organized as follows. In the section two it is

978-0-7695-5002-2/13 $26.00 © 2013 IEEE
DOI 10.1109/WETICE.2013.12

presented concepts related to telemedicine and the *CyclopsDCMServer*. Next section three is presented previous works which is the base to this study. Characteristics of the Ceph, FhGFS, Lustre and PVFS (distributed file systems utilized in this work) are illustrated in section four. The PH5WRAP parallel architecture is explained in section 5. Experimental results employing four distributed file systems are presented in section six. Finally, in section seven conclusions and future work of the present research are shown.

II. Image Standard and Communication System Server

A. Medical Images

In the 1970s, with the advent of the computed tomography (CT), new mechanisms for the storage and transmission of digital images inside hospitals were introduced. After the introduction of this new type of examination, other new diagnosis methods based on digital images appeared. Initially, each manufacturer created a different solution for storage, printing and visualization of digital images. Thus, each manufacturer had its own information for each type of exam[11].

The first version of this standard was published in 1985, and it was coined as ACR-NEMA Standards Publication No. 300-1985. Following this initial proposal, two revisions were published in 1986 and 1988. The latter was called as ACR-NEMA Standards Publication No. 300-1988. Both publications provided specifications related to hardware interface, a set of data format and also a set of commands for software packages.

In order to provide also a standard for networking, in 1992 a new proposal, called as ACR-NEMA Standards Publication PS3, was published. This standard was coined as DICOM3 (*Digital Imaging and Communications in Medicine ver.3*). The DICOM3 is accepted as a de facto standard for *Picture Archiving and Communications Systems* (PACSs). It is supported by the major medical digital image equipment manufacturers. Nowadays, as verified in many researches (e.g. [11]), any equipment that utilizes the DICOM standard indicates that can be easily integrated to an existing PACS. In the Figure 1 bellow is shown a example of DICOM image.

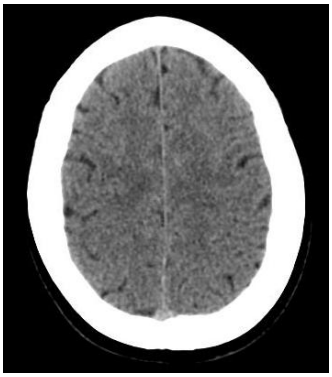

Figure 1. The DICOM image example

B. CyclopsDCMServer

The *CyclopsDCMServer* is a server that was conceived to provide a segmentation service for the incoming information of the statewide telemedicine network *Rede Catarinense de Telemedicina* (RCTM)[9], from the Santa Catarina State, in Brazil. In addition, the server also treats images and then stores all in a centralized database.

Afterwards, countryside physicians can remotely access findings results from exams, analyzed by a specialist. Results are available through a Telemedicine Portal[12]. One of the key services performed by the *CyclopsDCMServer* is a DICOM medical image facility that was designed to provide storage, considering wide area networks (WAN), and store all information in ordinary relational database. The actual DICOM medical image database hosted by the *CyclopsDCMServer* has an order of magnitude around twenty terabytes.

The *CyclopsDCMServer* can be considered as a special developed PACS. The major characteristic of the *CyclopsDCMServer* is to store and retrieve DICOM index files from a data repository managed by a relational DBMS such as PostgreSQL, MySQL or Oracle. All communication between the server and any medical equipment is performed through adopting the TCP/IP suite protocols. Nowadays, the server can support the following DICOM modalities:

- Computed Radiography (CR)
- Computed Tomography (CT)
- Magnetic Resonance (MR)
- Nuclear Medicine (NM)
- Ultrasound (US)
- X-Ray Angiography (XA)
- Electrocardiograms (DICOM Waveform)
- DICOM Structured Reporting (SR)

In [13] it is present a previous research work where the architecture of the *CyclopsDCMServer* is enhanced with PVFS for DICOM storage and retrieval. One of important feature from this architecture is the way which DICOM is organized in HDF5 structure as follow in Figure 2.

As it can be seen, the groups are represented in ellipses that contain one or more groups and may contain several or none dataset. Below the group root ("/"), there is the level of study which containing data about the exam and the patient. In the next series layer, contain metadata belong to the examination (e.g. Tomography). Finally, on the last level, linked to the series layer, there is an instance that is composed of various data about the image and the largest DICOM data: the pixel data.

Called *pixel data*, this contains the binary pixel of the image. Since the quality of the images is of great importance, this element contains the largest amount of data, resulting in

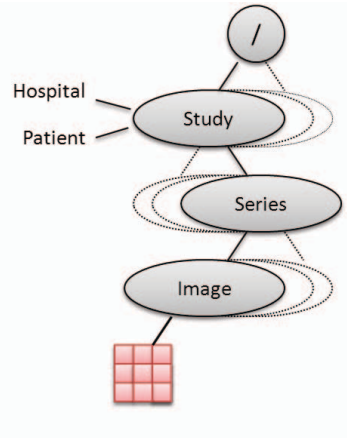

Figure 2. Hierarchical structure of a DICOM image in HDF5

a data set considerably large. Depending on the examination type, these images can reach up to 600Mbytes.

III. Previous Work

This work is part and is a continuation of previous work related to how to store and retrieve medical images in distributed scenarios. These works were started in 2008 where studies were replication techniques [14] for data integration Telemedicine Network. Still, we studied an approach for interoperability among distributed databases Internet [15].

However, due to these preliminary studies, this project encountered serious limitations on the use of relational databases for data persistence related to performance, flexibility, portability and maintenance of large volumes of data. Thus began the development of a persistence layer based on that open data formats, in which case the NetCDF and HDF5.

In [16] a study using the NetCDF data format was developed. Through it works where decides to follow using HDF5, once NetCDF presents lower performance, since the storage increases when the file grows. Thus in [17] and [13] it is presented a new approach which converts DICOM images into HDF5. This approach was called H5WL and after the H5Wrap is nominated. The experiments where performed on PVFS in comparative PostgreSQL DBMS with the ordinary and the result showed to perform great images in storage, but where decrease opened the retrieved image. In [18] there is a continuation of this work aimed at improving the performance of the approach.

IV. High Performance Distributed File Systems

Distributed file systems (DFS) can be classified by the target characteristics, which they have as a goal. As mentioned in [19] it is possible to broadly consider two categories.

In the first group, it is possible to find those distributed file systems that were designed to support applications executing in the Internet. Inside this class of DFS examples that could be found are: Google File System (GFS), Amazon Simple Storage Service (S3) and Hadoop File System (HDFS).

On the other hand, classic examples of the second category are Ceph, Fraunhofer File System (FhGFS), Lustre and Parallel Virtual File System (PVFS). This second category aims high performance computing aspects, in contrast to the Internet distributed file systems that are primarily designed and developed to execute under TCP stack protocols. However, each of the previous DFS, from the second category, has its own peculiarity with some disjoint characteristics as follows:

- *Ceph*: maximizes the separation between data and metadata management by replacing allocation tables with a pseudo-random data distribution function. This function, called as CRUSH, was designed for heterogeneous and dynamic clusters of unreliable object storage device;
- *FraunhoferFS* (FhGFS): it is the high-performance parallel file system from the Fraunhofer Competence Center for High Performance Computing. Its distributed metadata architecture has been designed to provide the scalability and flexibility that is required to run today's most demanding HPC applications. Unfortunately, it documentation is vague to academic community, but the experiments can show some similar structured to PVFS.
- *Lustre*: Authors from this DFS project claim that Lustre was conceived for the world's largest and most complex computing environments. The file system redefines high performance, scaling to tens of thousands of nodes and petabytes of storage with groundbreaking I/O and metadata throughput;
- *PVFS*: it is designed to provide high performance for parallel applications, where concurrent, large I/O and many file accesses are common. PVFS provides dynamic distribution of I/O and metadata, avoiding single points of contention, and allowing for scaling to high-end terascale and petascale systems. This DFS relaxes the POSIX consistency semantics where necessary to improve on stability and performance. Cluster file systems that enforce POSIX consistency require stateful clients with locking subsystems, reducing the stability of the system in the face of failures. These systems can be difficult to maintain due to overhead of lock management. PVFS clients are stateless allowing for effortless failure recovery and easy integration with industry standard high-availability systems.

V. Proposed Environment

In this section it is presented the PH5Wrap environment. This novel component - different from researches presented in [9], [11] and [12] represents an extension for the *Cyclops-*

DCMServer considering sequential and parallel readings and retrievals.

Figure 3 illustrates the general idea behind the proposed PH5Wrap environment. The main goal is to preserve the actual infrastructure working in a real day-by-day basis. This infrastructure nowadays is similar to H5Wrap layer, although the DICOM data is storage in an ordinary relational data base. In addition to H5Wrap layer has a parallel part which could be used with disjunctive distributed file systems. All of theses groups are called PH5Wrap.

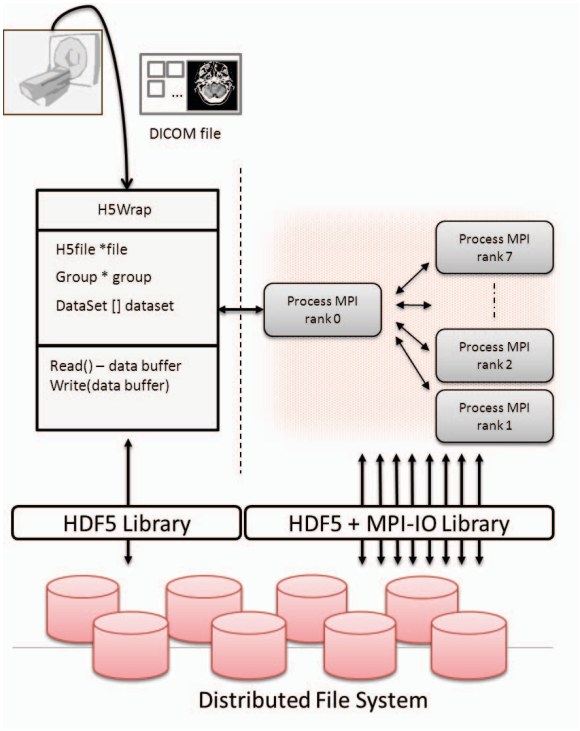

Figure 3. The PH5Wrap architecture

The PH5wrap is an abstract layer, which is responsible to convert DICOM image to HDF5 structure and perform both serial and parallel I/O in a DFS. To summary, all metadata from DICOM image it is parsing to HDF5 file by H5Wrap as serial I/O and the large data (pixel data) is transferred to MPI processes to realize high performance parallel I/O into HDF5 file.

Also, it is possible to configure PH5Wrap to only perform serial I/O on DFS, making the H5Wrap responsible to all operations such as writing the pixel data. Therefore, experiments shown in this paper only considered the left part of the Figure 3, which represents the serial I/O part. These experiments verify the I/O latency of *CyclopsDCMServer* as a single application between distinguished DFS and to analyse the buffering perform on each DFS.

VI. Experimental Results

In this section, it is presented empirical results considering four distributed file systems: Ceph, FhGFS, Lustre and PVFS. These experiments were performed considering DICOM images inside the private cloud of storage architecture. Thus, we focus on demonstrating the I/O latency according to increasing medical images volume and the scalability issue as performing experiments in four different volumes: 1000, 2500, 5000 and 10000 distinct DICOM images. These images come from RCTM and its average size is nearly to 512Kbytes.

The experiments were performing in an cluster seeking to use eight nodes as data servers, which one is a metadata server, as well. Each experiments was realized 25 iterations to ensure statistical significance in the results. For the figures below, the y-axis value is in seconds and the x-axis represents the medical images volume.

Figure 4 shows our first case study where we conceived the storage time for DICOM files using the four DFS. The value in seconds for this experiments, represents the time spent by the server to parse a DICOM image to HDF5 structure and write all data in the DFS.

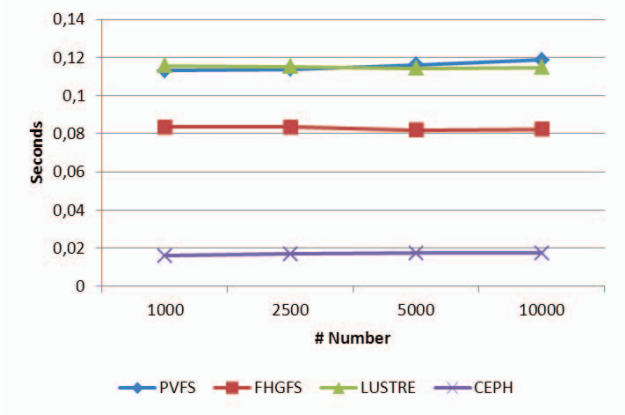

Figure 4. Storage time for DICOM files between the DFS.

The second set of experiments as shown in Figure 5 was characterized by the meantime of writing operation, only considering the large data on all DFS, despising the meantime of any other operation. This is aiming to collect and observer the DFS performs related to distributed and writes large data.

This graphs highlights an improved performance of the Ceph in comparison to the other DFS. While the Lustre obtained low performance. In storage time, systems PVFS, FHGFS, Lustre and CEPH had average performance of 0.11490, 0.08291, 0.11497 and 0.01714 seconds respectively. Thus the CEPH had a performance of 82.59% above the average of the whole system and around 92.11% more efficient in writing time of pixel data.

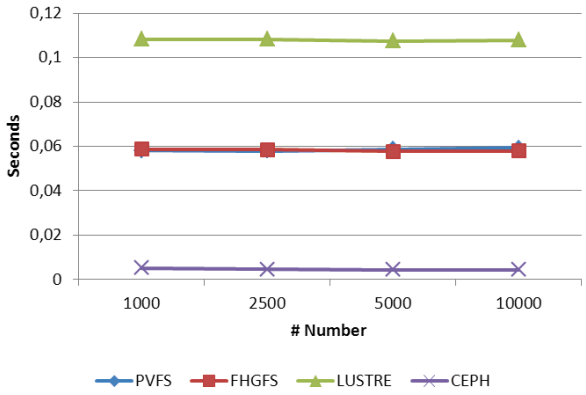

Figure 5. Writing time for only the pixel data between the DFS.

The next three experiments consider the retrieve meantime of HDF5 to DICOM from each hierarchical layer, considering the time spent since I/O latency to reconstructed the DICOM data. In Figure 6 it is presented the retrieve meantime of a specific study. On the other hand, Figure 7 shows the retrieve meantime of a specific series. Finally, in Figure 8 it is presented the retrieve meantime of an image.

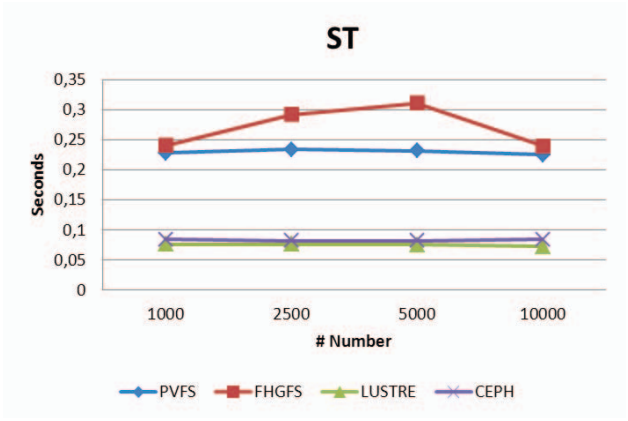

Figure 6. Retrieve meantime of a study layer.

In the results presented in these three graphs, it is possible to observe that both Lustre and CEPH achieved superior performance compared to parallel file systems FHGFS and PVFS. In the case of Lustre, it scores a top performance of 67.36% under the PVFS and 71.38% under FHGFS while CEPH achieved an average performance of 63.73% and 68.59%, respectively.

Furthermore, both file systems CEPH and Lustre showed to maintain the scalability with increasing volume of data. The average recovery of a complete study and series in seconds, are nearly to 0.075 seconds to 0.083 seconds, and the average recovery from image layer is 0.018 seconds in

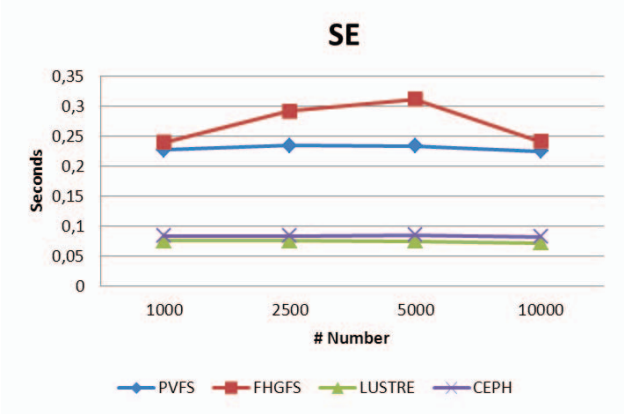

Figure 7. Retrieve meantime of a series layer.

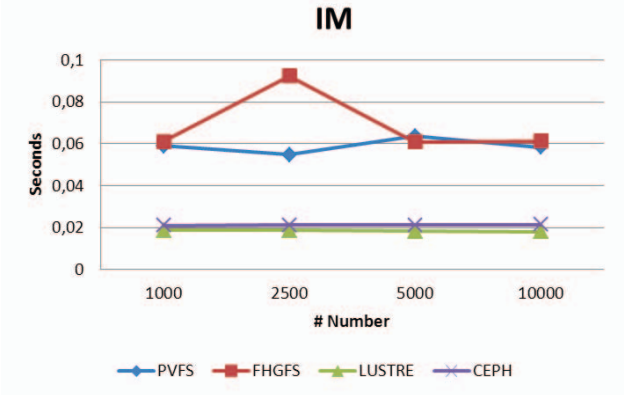

Figure 8. Retrieve meantime of an image layer.

Lustre, and 0.021 seconds in CEPH.

Its possible observe that Lustre and Ceph obtain a great percentage gain when compared with the others two DFS. This vantage can be specified mostly by the way which each DFS manages their files between the components, ensuring a better performance to locate the files on the environment.

VII. Conclusion

In this paper it was presented the PH5Wrap component, which is an extension of a Picture Archiving and Communication Systems (PACS), called as a *CyclopsDCMServer*. The aim of the proposal was to study the data management of DICOM medical images in a private cloud storage environment, utilizing disjoint high performance distributed file systems. Therefore, study case experiments were done to make a comparison of data management under the Ceph, FhGFS, Lustre and PVFS high performance distributed file systems.

Our results indicate that the Ceph distributed file systems has an interesting improvement to store and write of generic

files in contrast to the PVFS and others DFS. The former observation also occurs for retrieving experiments. These two experiments indicate that the day-by-day picture archiving and communication system (i.e. *CyclopsDCMServer*) is not employed the more appropriated DFS.

Future work for the present research considers experiments of the parallel part of the PH5Wrap and a mix of tests with both sequential and concurrent requests. The studies [20] and [10] show some previews results using the right part of PH5Wrap on PVFS. Even the experiments results are not applied to all processes (e.g. communication between H5Wrap and MPI process) and in [10] the experiments is performed in an environment networkless (virtual machines), the results demonstrated advantage in parallel side.

In addition, the research group is planning to extend the current private cloud storage environment to more institutions. Therefore, more studies of Internet distributed file systems will be required.

REFERENCES

[1] "Google (gfs)." [Online]. Available: http://research.google.com/archive/gfs.html

[2] "Amazon s3." [Online]. Available: http://aws.amazon.com/en/s3.

[3] "Hadoop file system (hdfs)." [Online]. Available: http://hadoop.apache.org/hdfs/

[4] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: a scalable, high-performance distributed file system," in *Proceedings of the 7th symposium on Operating systems design and implementation*, ser. OSDI '06. Berkeley, CA, USA: USENIX Association, 2006, pp. 307–320. [Online]. Available: http://dl.acm.org/citation.cfm?id=1298455.1298485

[5] F. FS. (2012) Fraunhoferfs - user guide. Fraunhofer Competence Center for High Performance Computing. [Online]. Available: http://www.fhgfs.com/wiki/wikka.php?wakka=UserGuide

[6] P. Schwan, "Lustre: Building a file system for 1000-node clusters," in *Proceedings of the 2003 Linux Symposium*, 2003, pp. 400–407.

[7] "Ibm gfs." [Online]. Available: http://www-03.ibm.com/systems/software/gpfs/

[8] P. H. Carns, W. B. Ligon, III, R. B. Ross, and R. Thakur, "Pvfs: a parallel file system for linux clusters," in *Proceedings of the 4th annual Linux Showcase & Conference - Volume 4*, ser. ALS'00. Berkeley, CA, USA: USENIX Association, 2000, pp. 28–28. [Online]. Available: http://dl.acm.org/citation.cfm?id=1268379.1268407

[9] R. Maia, A. von Wangenheim, and L. Nobre, "A statewide telemedicine network for public health in brazil," in *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, 0-0 2006, pp. 495 –500.

[10] T. S. Soares, T. C. Prado, M. A. R. Dantas, D. D. J. de Macedo, and M. A. Bauer, "An approach using parallel architecture to storage dicom images in distributed file system," *Journal of Physics: Conference Series*, vol. 341, no. 1, p. 012021, 2012. [Online]. Available: http://stacks.iop.org/1742-6596/341/i=1/a=012021

[11] P. R. Dellani, "Desenvolvimento de um servidor de imagens mdicas digitais no padro dicom," Master Thesis, Federal University of Santa Catarina, 2005.

[12] J. Wallauer, D. Macedo, R. Andrade, and A. von Wangenheim, "Building a national telemedicine network," *IT Professional*, vol. 10, no. 2, pp. 12 –17, march-april 2008.

[13] D. De Macedo, A. Von Wangenheim, M. A. R. Dantas, and H. Perantunes, "An architecture for dicom medical images storage and retrieval adopting distributed file systems," *Int. J. High Perform. Syst. Archit.*, vol. 2, no. 2, pp. 99–106, Mar. 2009. [Online]. Available: http://dx.doi.org/10.1504/IJHPSA.2009.032027

[14] D. de Macedo, H. Perantunes, R. Andrade, A. von Wangenheim, and M. Dantas, "Asynchronous data replication: A national integration strategy for databases on telemedicine network," in *Computer-Based Medical Systems, 2008. CBMS '08. 21st IEEE International Symposium on*, june 2008, pp. 638 –643.

[15] D. De Macedo, H. Perantunes, L. Maia, E. Comunello, A. von Wangenheim, and M. Dantas, "An interoperability approach based on asynchronous replication among distributed internet databases," in *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, 2008, pp. 658 –663.

[16] M. Magnus, T. Prado, A. von Wangenhein, D. de Macedo, and M. Dantas, "A study of netcdf as an approach for high performance medical image storage," in *Journal of Physics: Conference Series*, vol. 341. IOP Publishing, 2012, p. 012016.

[17] D. D. J. de Macedo, E. Comunello, A. von Wangenheim, and M. A. R. Dantas, "Armazenamento distribuído de imagens médicas dicom no formato de dados hdf5," in *Proceedings of the 14th Brazilian Symposium on Multimedia and the Web*, ser. WebMedia '08. New York, NY, USA: ACM, 2008, pp. 20–27. [Online]. Available: http://doi.acm.org/10.1145/1666091.1666097

[18] D. de Macedo, M. Capretz, T. Prado, A. von Wangenheim, and M. Dantas, "An improvement of a different approach for medical image storage," in *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2011 20th IEEE International Workshops on*. IEEE, 2011, pp. 140–142.

[19] G. G. W. Tantisiriroj, S. Patil, "Data-intensive
le systems for internet services: A rose by any other name ..." [Online]. Available: http://www.cs.cmu.edu/svp/2008tr-hdfspvfs.pdf

[20] T. Soares, D. de Macedo, M. Bauer, and M. Dantas, "A parallel architecture using hdf for storing dicom medical images on distributed file systems," *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'11)*, vol. part of WORLDCOMP'11, pp. 42–47, 2011, sunnyvale, California.