

Medical data management in the SYSEO project

Yahia Chabane, Laurent d'Orazio, Le Gruenwald, Baraa Mohamad, Christophe Rey

▶ **To cite this version:**

Yahia Chabane, Laurent d'Orazio, Le Gruenwald, Baraa Mohamad, Christophe Rey. Medical data management in the SYSEO project. SIGMOD record, 2013, 42 (3), pp.48-53. 10.1145/2536669.2536678 . hal-02062373

HAL Id: hal-02062373
https://hal.science/hal-02062373

Submitted on 8 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Medical Data Management in the SYSEO Project

Yahia Chabane*, Laurent d'Orazio*, Le Gruenwald**, Baraa Mohamad* and Christophe Rey*
* Blaise Pascal University, LIMOS UMR 6158, CNRS, Clermont-Ferrand, France
** School of Computer Science, University of Oklahoma, Oklahoma, United States
*firstname.lastname@univ-bpclermont.fr, **ggruenwald@ou.edu

ABSTRACT

The SYSEO project aims at producing a software solution suitable for endoscopic imaging in order to enable physicians to manage, manipulate and share medical images. This paper presents our two main components for data management in this system: (1) a novel hybrid row-column database for medical data storage within the cloud and (2) a system for semantic image annotation and retrieval relying on an ontology for polyps.

1. INTRODUCTION

Medical data management has become an important challenge. The emergence of new medical imaging techniques and the necessity to access medical data at any time have led to a need to find new solutions for managing these data. Our work focuses on Digital Imaging and Communications in Medicine (DICOM) [1], one of the most important medical standards. DICOM aims to achieve interoperability between medical imaging systems. The requirement in storing DICOM-based images is that all the study related information be stored within the image file so that the image can never be separated from its information by mistake. The wide use of DICOM standard has led to the development of some management systems, such as the Picture Archiving and Communication System (PACS) [2] and the ORDICOM (object-oriented) data type in Oracle 11G [3]. Unfortunately, such systems are highly expensive, IT experts dependent and not scalable. Particularly, in these systems the crash of a server may prevent accessing the required images. Moreover they do not include facilities to efficiently and intuitively annotate and retrieve images. Particularly, no semantic retrieval technique has been developed yet in this field, in spite of efforts in medical ontology building (see section 2).

To fill this gap, we have developed SYSEO [4] that includes facilities to deal with medical data effectively and efficiently. Indeed, it aims at producing a comprehensive software solution that enables physicians to acquire, enrich, store, retrieve, manipulate, share and export medical images easily.

Data management in SYSEO addresses the following key challenge: how to best store and query huge quantities of DICOM images and videos? Three dimensions have been investigated: (i) performance (allowing huge data to be stored and quickly accessed), (ii) relevance (allowing query results to be the most precise possible) and (iii) completeness (allowing the use of many query mechanisms and their associated advantages).

Our first proposal considers performance through a cloud-enabled and hybrid medical data management system. Such a system first takes advantage of the cloud features to provide a highly available and cost-effective solution; then it provides an appropriate storage model that overcomes the intense heterogeneity, complexity and huge size of DICOM files and, at the same time, provides high expressiveness. To deal with relevance, our second proposal is a semantic system that allows images to be intuitively annotated and retrieved via semantic web techniques. This semantic system implements a complete semantic approach for endoscopic polyp images annotation and retrieval. It is based on a polyp ontology we have developed. Our goal is to provide more relevance compared to classical syntactic search. Moreover, this ontology is the first step towards a reference library of annotated polyp images that physicians may use in their everyday practice. This article aims at showing how completeness is reached in SYSEO using our proposals and the various querying modalities it supplies.

This paper is organized as follows. Section 2 presents related works. Sections 3 and 4 present the hybrid cloud-based data store and the ontology-based system, respectively. Section 5 presents the system implementation. Finally, Section 6 concludes the paper with a discussion of future research.

2. RELATED WORK

2.1 Medical Data Stores

Picture Archiving Systems (PACS)[2] systems are used in many medical centers. They are very expensive and low-expressive (pre-defined queries). Additionally they do not cope with heterogeneity since they mostly use a relational database that stores all heterogeneous attributes in a blob-like datatype without any ability to interrogate them.

Oracle 11g introduces a DICOM support feature [3], consisting of a new data type ORDicom that allow any column of this type in a table to hold DICOM content. Even though Oracle provides indexing and compression techniques, each DICOM file is stored in a separate object, leading to significant data redundancy, and as a consequence, increasing the storage space and reducing the performance, especially when using certain DICOM-specific methods.

eDiaMoND [5] is a grid-enabled medical imaging database, that relies on an object-relational approach to store DICOM files. It supports only three modalities (secondary capture images, mammography x-Ray images and structured reports) and restricts users to a set of pre-determined queries. This system is designed over a grid (a structure with a limited number of dedicated severs); therefore it is not suitable for a huge infrastructure of unreliable machines (such as the cloud).

Commercial cloud-based medical systems exist, such as DicomGrid [6], but without any documentation or research papers about them.

2.2 Reasoning and Ontologies to manage Medical Images

The first purpose of medical ontologies [7] is to gather existing taxonomies so as to link together concepts having the same meaning but a different name [8, 9]. We refer to [10] for a more complete discussion of existing ontologies in medicine.

A concrete usage of medical ontologies is image annotation, especially in the case of syntactic keyword-based image retrieval system. The Medico scenario in the Theseus project [11] aims at setting up standards for the syntax and semantics in medical image annotation from ontologies. Our approach is quite similar in that we handle the annotation and retrieval problems using description logics. However, our aim is less oriented towards diagnosis than towards giving physicians a semantic infrastructure to manage their medical images. The AIM project [12] aims at setting up an ontology-based standard for the annotation and the markup of medical images. Our approach differs in that we put the semantic capabilities at the heart of the system since we use a true ontology (not a lexicon) based on a Description Logic (DL) [13] and associated with precisely defined reasoning. The semantic features seem not to be a main objective in the AIM project. Other works handle the issue of semantic image annotation [14, 15]. Our proposal is close to these works, but is different in the used retrieval reasoning.

Concerning DL image retrieval reasoning, what differs from one approach to another is the proximity notion that is used to qualify the good answer images to a query. We can find two classical approaches [16, 17, 14] which correspond to our R1 (see Table 1), which is the classical individuals retrieval, and the composition of R2 followed by R1, which amounts to finding images associated with concepts that have the same properties as the query (and maybe others properties). Other approaches are based on non-standard DL reasoning (abduction and contraction) [18, 19], which imply, however, the use of a less expressive DL. This reasoning enables a better ranking of answers than the previous one.

3. HYBRID CLOUD-BASED DICOM DATA MANAGEMENT SYSTEM

The architecture of SYSEO's DICOM Management System is illustrated in Figure 1. The implemented system shows interesting results to store and query DICOM files. We present in the following sections the main components in our solution: the data storage and the query execution.

3.1 Data Storage

The DICOM standard defines more than 3000 attributes. Only some of them are mandatory to be inserted in a given DICOM file, whereas the others are optional. Therefore, each DICOM file could contain a different subset of these attributes. Modifications/additions can be proposed by do-main experts resulting in a new version of the standard every year, so the schema is changing over time. We propose a hybrid (row-column) two cloud-based

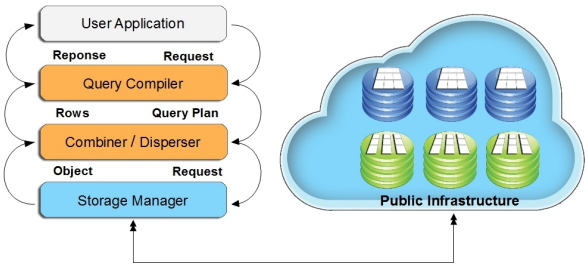


Figure 1: Data Management Architecture

layers data storage structure. Each of these layers is designed to store a special set of DICOM attributes. For that, we decompose the attributes into three categories: (1) Mandatory/frequently used attributes, (2) frequently accessed together attributes; and (3) optional/private attributes. Then, we propose the most appropriate layer to store each of them. We link these layers together by a unique identifier that allows us to reconstruct our DICOM files. Both (row-column) layers are cloud-based, which ensures the elasticity and fault tolerance for each of them (e.g. with GFS [20] storing automatically several copies of our data in geographically separated areas, a server crash is not a problem).

We propose to store mandatory/frequently used attributes (e.g. PatientID and Pixel Representation) and frequently accessed together attributes (e.g. Patient Name and Birth-date) in a traditional row-oriented database layer. We store frequenlty accessed together attributes in the same table in order to reduce join costs (tuple reconstruction time) by applying appropriate vertical partitioning. The advantage of this layer is its write-optimized feature (each tuple insertion in row-oriented databases needs one disk block I/O for insertion alone). Thus, having a lot of insertions over this layer will not be challenging. A sharded database, like Azure [21] or RDS [22], is a candidate solution for such a layer. However, in order to reduce the storage cost and have a more scalable solution, we focus on shared nothing MapReduce based approaches like Pig[23] or Hive [24].

Optional/private attributes (e.g., Smoking Status and Chemical Shift) vary enormously from one medical file/center to another. For these heterogeneous attributes, we propose storing them in column-oriented databases. Only non-null values will be inserted into their corresponding columns which improves significantly the query performance. Therefore, this model copes perfectly with our heterogeneous data. Columnar databases are OLAP-optimized, so this layer offers the ability to perform efficiently ad-hoc/statistical queries which are very selective. Additionally, this storage model provides a good solution for the evolutive schema issue since each column is stored in a separate disk block, so adding new columns is not challenging. On the other hand, the attributes stored in this layer are less frequently accessed together, so we minimize the tuple reconstruction time. A number of cloud columnar systems can be possible solutions for this layer. Examples are BigTable [25], Vertica [26] and HBase [27]. The high cost and proprietary features of Vertica and BigTable (GFS dependent) lead us to focusing on the other systems (e.g. HBase).

To improve the dynamicity of this storage, we plan to implement a column mover, which is a process that moves (when necessary) some attributes from one layer to the other when needed. A similar idea has been implemented by SAP database [28].

3.2 Query Execution

The query execution engine contains two main components: the Query Compiler and the Combiner/Disperser.

The Query Compiler is responsible for compiling user requests and translating them into a cloud query language. Actual systems (e.g. Pig, Hive) - under heavy development - have some limitations such as the absence of metadata/schema in some of them, and/or the lack of some functionalities (e.g. join). Therefore, new operations and optimizations will be added to adapt the used system to our hybrid storage.

The Combiner/Disperser is responsible for partitioning the coming queries according to the layers (row-oriented, column-oriented). After the query execution, the Combiner/Disperser is in charge of combining the results coming from both storage layers and sending the final results to the user.

In order to provide a good compromise between storage cost and query response time, we propose a query optimizer. It is responsible for choosing the best query plan/execution order (i.e. column layer first, or row layer first, or parallel execution of both layers) for executing the query over our hybrid storage model.

The query optimizer applies a Cost/Rule Based Optimization [29]. Yet the existing CBO/RBO solutions should be rethought for the cloud by taking into account the pay-per-use and elasticity features. In this context, we distinguish two query types. The first is the real time search where the physician may need certain images rapidly. In this case, the response time is crucial; so we may increase the number of resources used from the cloud according to the Service Level Agreement (SLA) [30].The second is the data analysis where the response time is not crucial; so we can reduce the used resources. Hence we maintain a good correlation between response time and resource cost.

4. USING SEMANTICS

The architecture presented in the previous section provides storage and querying capabilities on the DICOM attributes. Yet, there is not any standard set of attributes for a practitioner to store his/her observations of an image. In the best case, he/she

stores them as full text in a private DICOM attribute. The consequence is that query relevance may be low on these observations since physicians do not always use a standard vocabulary. That is why we propose to address the querying problem using a new ontology following a semantic web approach, namely a description logic approach.

Description Logic (DL) [13] is a well-known knowledge representation and reasoning formalism [13]. The OWL language [31], one of the main standards in semantic technologies, is based on DL. We now present the content of the polyp ontology, the annotation and query mechanisms.

4.1 The Polyp Ontology

The ontology is divided into three main parts related to the observable properties of the image content (colors, shapes, textures, etc.), its anatomical properties and the medical diagnosis comments on it. An image annotation is then defined as a set of information coming from these three parts. The base gastroenterological concepts come from four standard classifications that have been integrated in the ontology: (Paris, PitPattern, Vienne and MST which describe polyp shapes, polyp surfaces, polyp pathological states and many gastroenterological concepts respectively). Each concept coming from a classification and denoting a special set of polyps is called a class.

The language we choose to build our ontology is $\mathcal{SHOIQ}^+$ [32]. It is a very expressive DL for which the powerful HermiT reasoner is built [33].

4.2 Annotating and Querying

The process of our semantic image retrieval approach is illustrated in Figure 2.

First, DICOM images are stored in a cloud database (1). The ontology (2) is linked to this database via a keyword database (3). In the keyword database are stored image identifiers linked with keywords which are concepts taken from the ontology. Moreover image identifiers are also stored in the ontology as individuals that are instances of their associated image annotations. Two modules (4) and (5) ensure the coherency among (1), (2) and (3). Upon this knowledge infrastructure, the semantic image retrieval process runs as follows. First the system displays the concept hierarchy computed from the ontology (6). Then the user can browse it (7) and select a set of keywords which are concepts of the ontology (8). This set is then mapped to the generic definition of an image to obtain an image annotation (9). So, such an annotation is an instance of the generic definition of an image. Afterwards, this annotation can either be stored in the ontology (this is the annotation scenario)(10), or this annotation can be viewed as a query (this is the semantic retrieval scenario)(11).

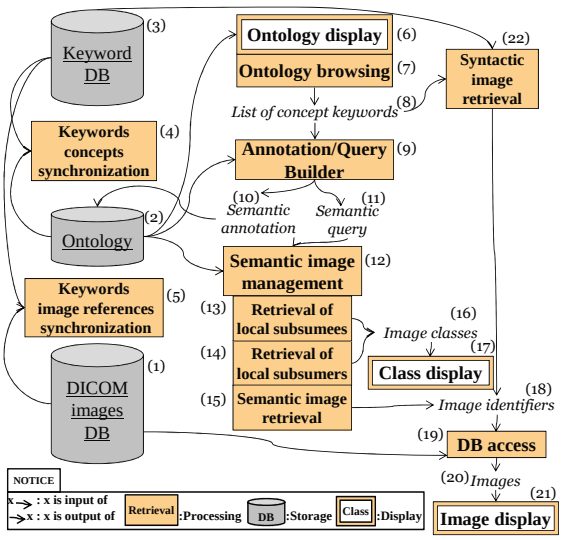


Figure 2: Semantic Image Retrieval Approach

#	Scenario	#	Reasoning	Fig. 2
S1	Semantic images retrieval	R1	Individual retrieval	(15)
S2	Exact classes retrieval	R2	Retrieval of local subsumees	(13)
S3	Approximated classes retrieval	R3	Retrieval of local subsumers	(14)

Table 1: Scenarios and corresponding reasonings.

We define three semantic retrieval cases (12): S1 for semantic images retrieval (15), S2 for exact class retrieval (13) and S3 for approximated class retrieval (14). We propose three kinds of DL reasoning (R1, R2 and R3) to implement them (see Table 1). Reasoning R1 is well-known in the DL litterature: we want to find all individuals (image identifiers) that belong to a given concept description (the annotation). Reasoning R2 and R3 are mainly based on subsumption (i.e. the subclass/superclass relation among concepts). In R2, we find all the subclasses of a given classification (e.g. Paris or PitPattern) that are also subclasses of the query. In R3, we find all the superclasses of the query that are subclasses of a given classification. That is why it is an approximation reasoning. Once image identifiers have been obtained (18), the system looks for (19) their associated DICOM images (20). Then the images can be displayed (21). Once the image classes have been inferred (16), they can be displayed to the user (17). A very interesting feature in this process is that the semantic part can be inserted within a

classical syntactic retrieval. Indeed, once the list of keywords is known (8), a keyword-based search engine can be run (22) to retrieve image identifiers (18) from the keyword database (3).

To conclude this part, we point out how the cloud-based and the semantic query mechanisms complement each other. The former allows efficient image retrieval with a classical syntactical approach on DICOM attributes. The latter allows less efficient but more relevant image retrieval with a semantic approach on semantic descriptions of the images that are not already stored as DICOM attributes. Moreover, classical syntactic retrieval can be achieved over concepts from these semantic descriptions. That is why we claim that the Syséo query mechanism is complete.

5. IMPLEMENTATION

5.1 Implementation Details

For the development of our application, we have used Struts 2 framework and servlets for the implementation of the MVC pattern, and JSP for the user-interface creation.

For the Storage system we have built the row-oriented layer using Pig.We have simulated the columnar storage by the use of ZEBRA library over Pig. We have developed dedicated user-defined functions for the parsing and decomposition of DICOM files for the corresponding layers. The user-interface is dynamic to allow the user create easily his/her query. This query is then translated into the corresponding Pig query language. Our current work is about assuring efficient query execution over our storage structure by proposing new optimizers.

For the semantic part, we have used the OWL API for ontology manipulating, the HermiT reasoner [33] for reasoning on the ontology and Prefuse [34] for creating user interfaces of image annotations and query generation.

5.2 Example

We show in Figure 3 an example illustrated within a general schema of our system.

Adding Images: When the physician wants to save a new image, the DICOM parser reads and decomposes the image into three categories: 1) attributes should be sent to the private infrastructure (health care data center) (e.g.Patient Name), 2) attributes to be stored in the row-oriented layer (e.g. Patient ID) and 3) attributes will be stored in the column storage layer (e.g. Pregnancy Status).

Retrieving Images/Statistics: The user uses the user-interface to create her query. The query is then written in the PigLatin query language. The Disperser rewrites the query according to the location of each of the required tables/attributes (e.g. Patient ID, Sex and Birth date attributes belong to the patient table on the row-oriented layer whereas SmokingStatus resides on the columnar layer).

Semantic Annotation and Retrieval: Image annotation and queries are generated manually using an interactive user-interface. This interface allows navigation in the ontology. According to the physician observation/need, she selects the most appropriate concepts and individuals for the representation of images.

The annotation (query) mechanism building is illustrated in Figure 3. The user selects three concepts of ontology: stomach, orange and haemorrhaic. The subsumers (belonging to the annotation concept definition) of these concepts will be determined in order to select the most appropriate roles for each concept. Thereafter, a concept description is built from these subsumers and roles. The result is the user annotation (or the user query).

6. CONCLUSION

In this paper, we presented the data management designs in the SYSEO project. We introduced a cloud-enabled hybrid database and semantic approach for medical data management. The challenges in this context are due to the high heterogeneity and huge volumes of DICOM files. For that we propose a new architecture providing: (1) ease of use, high performance and ad-hoc queries over DICOM files, (2) the capacity to exploit the cloud elasticity, billing-by-use and scalability and (3) give a complete and flexible semantic infrastructure to manage medical images, diagnosis and education.

The next objective of our project is to validate our prototype for real medical applications. We are about to integrate different solutions and install them in hospitals in order to validate the solutions and the Ontology. We plan to achieve a high level of QoS that allows querying large amounts of data via different types of computing devices. Additionally, some optimization (e.g. materialized views, cache manager, semantic reasoning) should be rethought for our particular structure. In the near future, we will provide more details about the prototypes and results on the project Web site [4].

7. ACKNOWLEDGEMENT

This work is supported by Yansys, the Agence Nationale de la Recherche (under grant SYSEO ANR-10-TECSAN-005-01), and the Conseil Régional d'Auvergne.

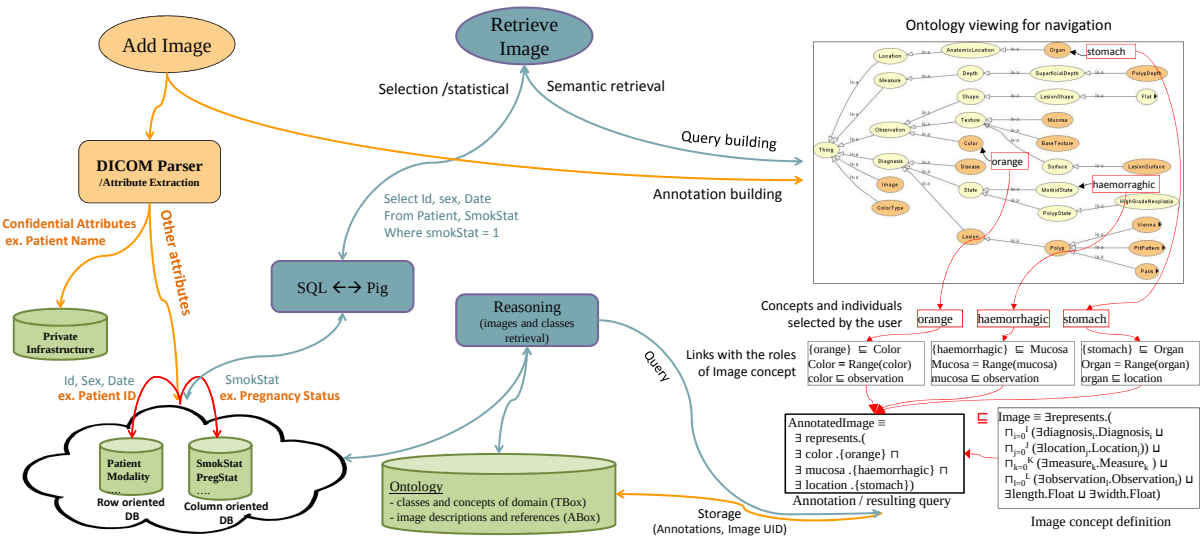


Figure 3: Image Adding and Retrieval

8. REFERENCES

[1] "Dicom," 2013. http://medical.nema.org/.

[2] M. Tieche, J. Gump, M. E. Rieck, and A. Schneider, "PACS: Picture Archiving and Communication Systems," *Springer*, 2010.

[3] Oracle, "Oracle Database 11g DICOM Medical Image Support," Tech. Rep. September, 2009.

[4] "Syseo," 2013. http://www.syseo-anr.fr/.

[5] D. J. Power, E. A. Politou, M. Slaymaker, S. Harris, and A. C. Simpson, "A relational approach to the capture of dicom files for grid-enabled medical imaging databases," in *SAC*, pp. 272–279, 2004.

[6] "Dicomgrid," 2013. http://www.dicomgrid.com.

[7] "Openclinical: knowledge management for medical care," 2011. http://www.openclinical.org/ontologies.html/.

[8] "Galen and the galen-core high-level ontology for medicine.," 2011. http://www.opengalen.org/.

[9] U. N. L. of Medicine, "Systematized nomenclature of medicine - clinical terms," 2011. http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html.

[10] "Bioportal," 2011. http://bioportal.bioontology.org/.

[11] G. F. M. of Economics and Technology, "Theseus project, medico scenario," 2010. http://theseus.pt-dlr.de/en/920.php.

[12] S. C. for Biomedical Informatics Research., "Aim project.," 2010. http://cabig.cancer.gov/solutions/applications/aim/.

[13] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, eds., *The Description Logic Handbook: Theory, Implementation, and Applications (2nd Edition)*, Cambridge University Press, 2007.

[14] D. L. Rubin, P. Mongkolwat, V. Kleper, K. Supekar, and D. S. Channin, "Medical imaging on the semantic web: Annotation and image markup.," in *AAAI Spring Symposium: SSKI*, pp. 93–98, AAAI, 2008.

[15] P. Wennerberg, K. Schulz, and P. Buitelaar, "Ontology modularization to improve semantic medical image annotation.," *JBIOBL*, vol. 44, no. 1, pp. 155–162, 2011.

[16] E. D. Sciascio, F. M. Donini, and M. Mongiello, "Semantic indexing in image retrieval using description logic," in *ITI*, 2000.

[17] B. Hu, S. Dasmahapatra, P. H. Lewis, and N. Shadbolt, "Ontology-based medical image annotation with description logics," in *ICTAI*, pp. 77–, 2003.

[18] T. Di Noia, E. Di Sciascio, F. M. Donini, F. di Cugno, and E. Tinelli, "Non-standard inferences for knowledge-based image retrieval," in *EWIMT 2005, IEE press*, pp. 191–197, IEE, 2005.

[19] S. Colucci, T. D. Noia, E. D. Sciascio, F. M. Donini, and M. Mongiello., *Description Logic-Based Resource Retrieval.*, pp. 185–197. Encyclopedia of Knowledge Management, 2011.

[20] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The google file system," in *SOSP*, pp. 29–43, 2003.

[21] "Windows azure," 2011. http://www.microsoft.com/windowsazure/.

[22] "Amazon rds," 2012. http://aws.amazon.com/rds/.

[23] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins, "Pig latin: a not-so-foreign language for data processing," in *SIGMOD*, pp. 1099–1110, 2008.

[24] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "Hive - a warehousing solution over a map-reduce framework," *PVLDB*, vol. 2, no. 2, pp. 1626–1629, 2009.

[25] F. Chang, J. Dean, S. Ghemawat, W. Hsieh, D. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber, "Bigtable: A distributed storage system for structured data," *ACM TOCS*, vol. 26, 2008.

[26] "Hp vertica," 2011. http://www.vertica.com/.

[27] "Apache hbase," 2011. http://hbase.apache.org/.

[28] P. Rösch, L. Dannecker, F. Färber, and G. Hackenbroich, "A storage advisor for hybrid-store databases," *Proc. VLDB*, vol. 5, no. 12, 2012.

[29] H. Pirahesh, J. M. Hellerstein, and W. Hasan, "Extensible/rule based query rewrite optimization in starburst," in *SIGMOD*, pp. 39–48, 1992.

[30] S. A. Baset, "Cloud slas: present and future," *SIGOPS*, vol. 46, July 2012.

[31] "Owl, the web ontology language," 2007. http://www.w3.org/2007/OWL.

[32] B. Motik, R. Shearer, and I. Horrocks, "Hypertableau reasoning for description logics," *JAIR*, vol. 36, pp. 165–228, 2009.

[33] "Hermit," 2012. http://www.hermit-reasoner.com/.

[34] J. Heer, S. K. Card, and J. A. Landay, "prefuse: a toolkit for interactive information visualization," in *CHI*, pp. 421–430, 2005.