RadioGraphics

Informatics in Radiology

Use of CouchDB for Document-based Storage of DICOM Objects[1]

TEACHING POINTS
See last page

Simón J. Rascovsky, MD, MSc • Jorge A. Delgado, MD • Alexander Sanz, BS • Víctor D. Calvo, BS • Gabriel Castrillón, BS

Picture archiving and communication systems traditionally have depended on schema-based Structured Query Language (SQL) databases for imaging data management. To optimize database size and performance, many such systems store a reduced set of Digital Imaging and Communications in Medicine (DICOM) metadata, discarding informational content that might be needed in the future. As an alternative to traditional database systems, document-based key-value stores recently have gained popularity. These systems store documents containing key-value pairs that facilitate data searches without predefined schemas. Document-based key-value stores are especially suited to archive DICOM objects because DICOM metadata are highly heterogeneous collections of tag-value pairs conveying specific information about imaging modalities, acquisition protocols, and vendor-supported postprocessing options. The authors used an open-source document-based database management system (Apache CouchDB) to create and test two such databases; CouchDB was selected for its overall ease of use, capability for managing attachments, and reliance on HTTP and Representational State Transfer standards for accessing and retrieving data. A large database was created first in which the DICOM metadata from 5880 anonymized magnetic resonance imaging studies (1,949,753 images) were loaded by using a Ruby script. To provide the usual DICOM query functionality, several predefined "views" (standard queries) were created by using JavaScript. For performance comparison, the same queries were executed in both the CouchDB database and a SQL-based DICOM archive. The capabilities of CouchDB for attachment management and database replication were separately assessed in tests of a similar, smaller database. Results showed that CouchDB allowed efficient storage and interrogation of all DICOM objects; with the use of information retrieval algorithms such as map-reduce, all the DICOM metadata stored in the large database were searchable with only a minimal increase in retrieval time over that with the traditional database management system. Results also indicated possible uses for document-based databases in data mining applications such as dose monitoring, quality assurance, and protocol optimization.

Abbreviations: BLOB = binary large object, DICOM = Digital Imaging and Communications in Medicine, ID = identification, JPEG = Joint Photographic Experts Group, JSON = JavaScript Object Notation, PACS = picture archiving and communication systems, RDBMS = relational database management systems, REST = Representational State Transfer, SQL = Structured Query Language, UID = unique identifier, WADO = Web Access to DICOM Objects

RadioGraphics 2012; 32:913–927 • **Published online** 10.1148/rg.323115049 • **Content Code:** IN

[1]From the Department of Research, Instituto de Alta Tecnología Médica de Antioquia, Cra 50 #63-95, Medellín, Colombia. Presented as an education exhibit at the 2010 RSNA Annual Meeting. Received March 11, 2011; revision requested July 21 and received October 29; accepted January 11, 2012. All authors have no financial relationships to disclose. **Address correspondence to** S.J.R. (e-mail: *investigacion@iatm.com.co*).

Introduction

Medical imaging has been evolving in much the same way as the World Wide Web. A side effect of this evolutionary transformation is the need for efficient handling of increasing amounts of medical imaging data. In addition to the well-established trend toward larger data volumes, there seems to be a more recent trend toward increased collaboration and a resultant broader distribution of medical imaging data facilitated by cloud-based storage and applications.

When one looks at the way medical imaging systems have been designed over the years, it becomes apparent that the traditional picture archiving and communication systems (PACS) cannot accommodate this transformation. Recently, a new breed of database management systems has appeared that is described as "nonrelational" (1). Most of these new systems have their origins in technologies developed by Google and Amazon, two undisputed champions of "big data" management. Nonrelational databases are designed to be highly scalable and to excel at handling large datasets in near-real time.

The article explains why nonrelational document-based key-value stores may be more efficient than the relational database management systems (RDBMS) that PACS traditionally have relied on for the storage and distribution of medical imaging data. The authors summarize their experience in creating a document-based database at their institution, testing its performance against that of a traditional database, and assessing its additional use for data mining, dose monitoring, quality assurance, and teleradiology.

Increasing Challenges for Imaging Data Storage and Retrieval

PACS have been adapted incrementally over time to accommodate the evolution of medical imaging: Storage capacity has been added to accommodate ever-increasing volumes of data, and user-requested functionality has been implemented to allow simultaneous viewing at multiple radiology sites, both locally and remotely (teleradiology). However, the overall architecture of imaging information archival has not changed substantially over the years. PACS traditionally have relied on relational, schema-based Structured Query Language (SQL) databases for imaging data management. RDBMS have a fixed design in which a schema of tables and relations between those tables ("joins") is defined at the outset and any data inserted thereafter must conform to the predefined schema. To improve the speed of data retrieval operations, an "index" (a list of values extracted from a particular column in a table and stored in a format that allows faster access) is usually created during database design and implemented at the expense of storage space. Although the RDBMS supporting most PACS are mature and robust, they may no longer be the best platform for medical imaging data archival and distribution.

Most radiologic imaging data that are archived on PACS consist of Digital Imaging and Communications in Medicine (DICOM)-compliant objects. DICOM is an international standard that defines file formats, network services, and requirements for the transmission and storage of digital medical images and reports. The basic structure of a DICOM object includes a header (which usually identifies the file as a DICOM file), a metadata portion (containing information about the image), and a pixel data portion (containing the actual image pixels). The purpose of this structure was to create "self-contained" units of imaging information. The DICOM metadata portion consists of data elements or "tags" that encode the patient-related attributes, study-related attributes (eg, accession number), and image-related attributes (eg, image size) of the DICOM object. The attributes of a particular DICOM file might require the inclusion of hundreds of such tags, but even that number is a small subset of the more than 3000 DICOM tags currently available for use.

Although the DICOM standard includes a dictionary of permissible tags, it makes no attempt to classify the objects described and thus imposes no schema. Classification, if desired,

must be performed by the system processing the metadata for storage and retrieval. DICOM objects, as tag-value documents, are inherently heterogeneous and cannot be represented adequately by RDBMS unless compromises are made. When DICOM objects are loaded into RDBMS, much of the metadata must be stripped out so that only information that is supported by the predefined database schema remains.

Forcing freely structured DICOM objects into a predefined schema poses a difficult problem: Which tags should be stored? The obvious answer is "all of them"; however, the necessary result of using this approach would be database collapse, because RDBMS are not usually designed to accommodate tables with more than 1024 columns. Even if a particular database were capable of storing more extensive data tables, indexing and retrieval would be impracticable.

A common solution to this problem is to store as columns in the database tables only the DICOM tags that are most frequently used in indexing and retrieval, relegating most other tags to nonindexable Binary Large Object (BLOB) fields, the contents of which are defined by the storage application. Any tags not stored in one of these ways can be directly retrieved by reading the DICOM file during runtime or ignored if they are considered irrelevant to the PACS operation. Although this solution may seem practical, it could result in the waste of informational content deemed unimportant at present that might be needed in the future.

One example of such content is radiation exposure information. Radiation exposure monitoring has gained renewed attention in recent years, since the publication of study results highlighting the potential health risks of medical imaging–related radiation exposure (2). In the past, the risks from most such exposures were considered insignificant. However, it has been shown that repeated imaging examinations over a patient's lifetime—especially those performed in childhood—can contribute to an increased cumulative risk for malignancies. Many current digital radiography and computed tomography (CT) systems provide capabilities for estimating radiation exposures and dose levels, and the resultant radiation exposure information is storable as DICOM metadata. However, image data management systems such as PACS must be substantially restructured to allow meaningful tracking of radiation dose–specific DICOM tags that previously were unavailable or were considered irrelevant. Such restructuring necessarily involves the creation of separate modules for registering this information during initial data archival, retrospective alteration of the PACS database tables to permit indexing of these parameters, or both. Thus, a substantial effort must be made to accommodate the information.

Radiation exposure data is an important example of information now considered necessary that previously was either captured and discarded or was never captured because it was unavailable when the PACS database was designed. Other, similar instances could occur. For example, if the specific absorption rate in magnetic resonance (MR) imaging examinations were found in future to have clinically significant cumulative health effects, the predefined schemas of RDBMS would have to be restructured again to allow these data to be archived in a searchable format in PACS.

Document-based Databases

The shortcomings of RDBMS for applications that model real-world data generation have not gone unnoticed. Over the past few years, an ecosystem of nonrelational or "noSQL" database management systems has emerged, evolving from pioneering efforts such as Google's BigTable, Amazon's Dynamo, and Apache's Cassandra. More than 50 nonrelational database management systems are currently available. These may be loosely categorized as shown in Table 1. Document stores or document-based databases mirror real-world imaging data generation. They share the schema-free nonrelational design of standard key-value stores but offer greater functionality (eg, secondary indexing, complex querying and sorting).

Table 1
Nonrelational (NoSQL) Database Management Systems

System Class and Type	System Name
Core noSQL systems	
Wide-column stores and column families	Hadoop, Cassandra, Hypertable, Cloudera, Amazon SimpleDB, SciDB, OpenNeptune, Qbase, KDI
Document stores*	CouchDB, MongoDB, Terrastore, ThruDB, OrientDB, RavenDB, Citrusleaf, CloudKit, Persevere, Jackrabbit
Key-value stores and tuple stores	Azure Table Storage, MEMBASE, Riak, Redis, Chordless, Scalaris, Tokyo Cabinet/Tyrant, GT.M API, Keyspace, Berkeley DB, MemcacheD, HamsterDB, Faircom C-Tree, Mnesia, LightCloud, Pincaster, GenieDB, Scality, KaTree, TomP2P, Kumofs, NMDB, luxio, actord, flare, schema-free, RAMCloud
Eventually consistent key-value stores	Amazon Dynamo, Voldemort, Dynomite, KAI, SubRecord, Mo8onDb, Dovetaildb
Graph databases	Neo4J, Sones, InfoGrid, HyperGraphDB, AllegroGraph, Bigdata, DEX, Infinite Graph, OpenLink Virtuoso, VertexDB, Java Universal Network, Graph Framework, Sesame, Filament, OWLim, NetworkX, iGraph
Soft noSQL systems	
Object databases	db4o, Versant, Objectivity, Gemstone, Progress, Perst, ZODB, NEO, StupidDB, KiokuDB, Durus
Grid database solutions	GigaSpaces, Hazelcast, Joafip, GridGain, Infinispan, Coherence, eXtreme Scale
XML databases	Mark Logic Server, EMC Documentum xDB, Tamino, eXist, Sedna, BaseX, Xindice, Qizx, Berkeley DB XML
Multivalue databases	U2, OpenInsight, OpenQM
Other noSQL-related databases	IBM Lotus/Domino, eXtremeDB, ISIS Family, Prevayler, Yserial

*All systems in this category are document-based database management systems.

As a result of their nonrelational design, most document-based systems require a new method of searching the database, a method that is completely different from the SQL-based queries used in RDBMS. Information extraction from most document-oriented databases and key-value stores is performed by using "map-reduce," an algorithm devised by Google to efficiently analyze enormous amounts of data. Map-reduce, which has since been incorporated into many document-based data management systems, improves the speed of the database search and retrieval of the information requested. The map-reduce routine runs through the complete database only once, the first time it is invoked. After that, if a document is created or changed, map-reduce only recomputes the functions on the basis of the altered data.

A detailed description of the complexities of the map-reduce algorithm is available elsewhere (3).

Briefly summarized, map-reduce is a batch analysis routine that performs two functions: The first function ("map") computes a set of intermediate key-value pairs based on the search filter criteria; the second function ("reduce") aggregates all the values that share the same key (Fig 1).

Advantages of Map-Reduce over SQL for Queries

Queries in RDBMS and in document-based systems have the same goal: to filter information on the basis of search criteria and aggregate that information according to the context. SQL allows the creation of database schemas and index structures and the management of informational content by performing "create," "update," and "delete" operations. Map-reduce is designed for efficient processing and retrieval of large amounts of data. The map function processes an input key-value pair, converting it to an intermediate key-value pair that then becomes the input for the reduce function. The reduce function merges

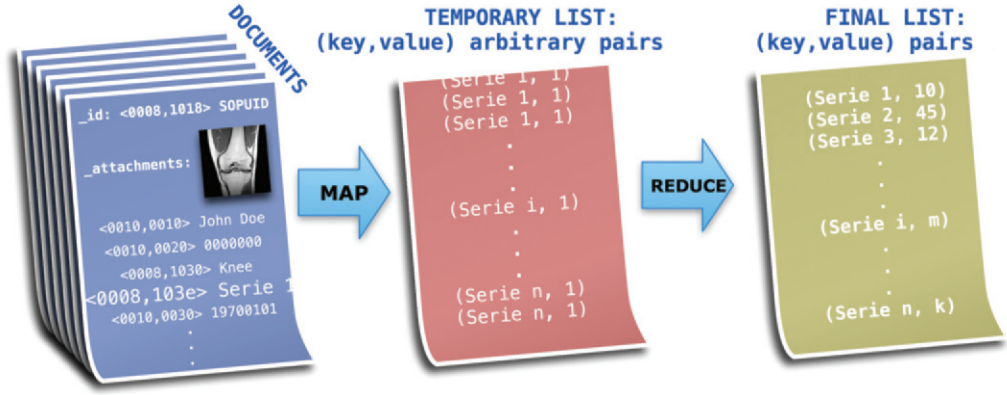

Figure 1. Schema shows a typical map-reduce operation for *n* documents representing DICOM objects stored in a CouchDB database. To identify the image series belonging to a specific study (eg, a contrast material–enhanced series and an unenhanced series belonging to a chest CT examination), a "map" routine is executed first; this produces a temporary key-value list in which the series descriptor is the key and the number 1 is the value. Next, a "reduce" routine sums all the values with the same series descriptor and computes a list of all series in the study.

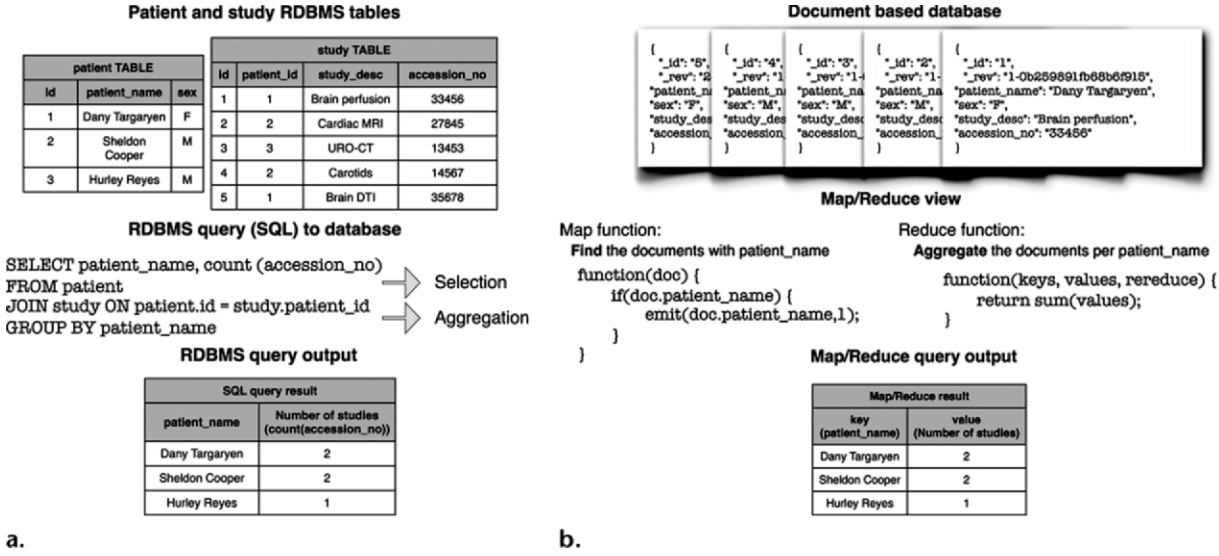

Figure 2. Schemas show the different operations that are performed to count the number of imaging studies per patient with RDBMS **(a)** and with document-based database systems such as CouchDB **(b)**. With RDBMS, a SQL query is sent to the database to select and aggregate the necessary DICOM metadata; with document-based management systems, the same DICOM metadata are retrieved by using a map-reduce query.

the intermediate key-value pairs to produce useful output that corresponds to the query (Fig 2).

Both algorithms allow data searches and analyses from various perspectives and with different use cases. However, because map-reduce operations are performed incrementally, results do not have to be recomputed for the entire dataset each time a function is run. By contrast, a SQL query always results in recalculation based on all current data stored in the database. Furthermore, the index structure in SQL must be formalized when the database schema is created; in contrast, map-reduce uses as its indexing mechanism the operation key, which is a dynamic variable that can be defined by any data field within a document.

Advantages of Document-based Systems for Imaging Data Storage and Retrieval

Teaching Point

Document-based database management systems are naturally more suitable than RDBMS for storing and retrieving DICOM objects, and being schema-less, they are free of the limitations of RDBMS. DICOM metadata can be stored in document-based databases without any modification or stripping of tags. If it is desirable to add query functionality to a tag not previously considered relevant, such a change is easily made. Equally important is the capability for future storage of tags that have not yet been defined; this capability of document-based databases makes them ideal for use in an environment of continuously evolving standards such as DICOM and Health Level 7 (HL7). The unique algorithms available for querying these databases allow efficient handling of large amounts of data.

Advantages of CouchDB

Teaching Point

CouchDB (developed by the Apache Software Foundation, Los Angeles, Calif; available at *couchdb.apache.org*) is an open-source document-oriented database management system in which querying and indexing are performed by using JavaScript (4). We selected it as a model for testing because of its overall usability and practicality (5,6). The features of CouchDB dovetail with the current data management needs of medical imaging practices. Among the characteristics that make this system particularly appropriate for medical imaging data management are its operational reliance on common Web standards (HTTP, Representational State Transfer [REST]); its capability for managing attachments to documents; its functionality in regard to data replication; and its potential utility for data mining applications such as dose monitoring, quality assurance, and protocol optimization.

Web Standards–based Communication.—CouchDB provides a RESTful JavaScript Object Notation (JSON) Application Programming Interface (API) that can be accessed from any environment that supports HTTP requests. REST is an architecture style for designing networked applications. HTTP is used to make communication calls between RESTful applications by using standard requests to submit, edit, read, query, and delete data. REST relies on HTTP protocols for database operations such as create, read, update, and delete. The HTTP- and REST-based architecture of CouchDB uses and builds on the entire existing body of knowledge that supports Web-based operations (eg, access, security, proxies, load balancing), which makes CouchDB extensible with a minimal learning curve. Because CouchDB uses standard HTTP requests for all its operations, it is instantly portable to HTTP implementations, which are available for every programming language. In addition, CouchDB acts as its own high-performance Web server, providing front-end functionality with direct coupling to the database, with no middleware.

Ability to Manage Attachments.—CouchDB has the ability to associate attachments with documents and store them in much the same way as attachments can be associated with e-mail. Attachments can stand alone if referenced in a related "parent" document, or they may be embedded within the database. Any type of file can be stored as an attachment. The storage of DICOM pixel data as an attachment facilitates the retrieval of the radiologic image data along with the DICOM metadata. This feature sets CouchDB apart from other document-based database management systems (Fig 3).

Data Distribution and Database Replication to Multiple Sites.—Any number of CouchDB hosts can replicate the same database with full functionality: Database changes from the host (source) can be copied to another database (target) or replicated bidirectionally on request or automatically at connection. In this respect, CouchDB provides out-of-the-box distributed radiology functionality. Document queries can specifically reference the desired DICOM objects, and pixel data stored as attachments can be replicated automatically or on demand. The ACID properties (*a*tomicity, *c*onsistency, *i*sola-

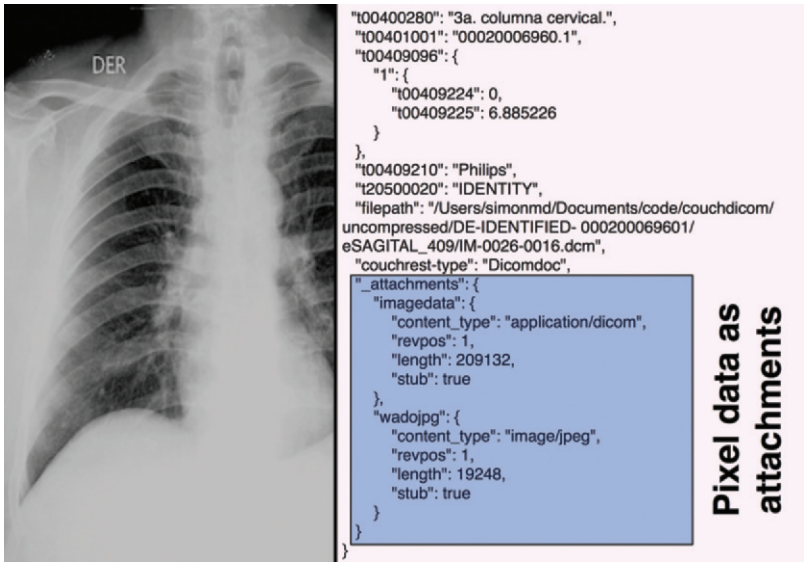

Figure 3. Anteroposterior radiograph stored as an embedded attachment in a document in the CouchDB database. The pixel data were uploaded by executing a standard HTTP POST request. The DICOM metadata (right) include references to the attachment in both DICOM and JPEG formats (blue shading). Binary data of all kinds can be stored as stand-alone or embedded attachments in a document database. *DER* = right.

tion, *d*urability) of CouchDB ensure that all data retain their integrity throughout replication, synchronization, and other database operations even if the operation is unexpectedly interrupted (eg, by a power failure); the interrupted operation resumes automatically after connectivity is restored. CouchDB also allows filtered replication, which means that a subset of objects in the database (eg, data relating to all the patients of a particular referring physician) can be designated for selective replication to the appropriate remote client or database.

Possible Use Cases for CouchDB

Many of the scenarios described in this section would be possible also with other document-based database systems. However, several (eg, multisite functionality) are predicated on replication and attachment management features specific to CouchDB.

Private Tag Storage

DICOM private tags are sets of attributes that are reserved for storing vendor-specific information about an image within the DICOM metadata. Private tags are commonly used to designate additional postprocessing functionality available on the vendor's workstations or with use of the vendor's software. If RDBMS-based PACS are configured to process private tags, these tags can be stored and referenced (usually as BLOB fields); however, the storage of private tags often has to be specifically enabled. Because every single DICOM tag can be natively included in a document-based database without special configuration, all of the DICOM metadata are retained, including private tags, which are sometimes lost during transfer to RDBMS-based PACS.

Archival of All Information from the Radiology Work Flow

This might be the area in which document-based database management systems most obviously excel, since they are designed to store information in a manner similar to the way real-life information is created. CouchDB is able to store both structured and unstructured data, information of an almost unlimited variety, from any step in the radiology work flow (Fig 4). The types of documents that might be stored in a local document database for a radiology department might include an order entry and scheduling information, patient demographic information, paper-based referring physician orders, scans of the patient's

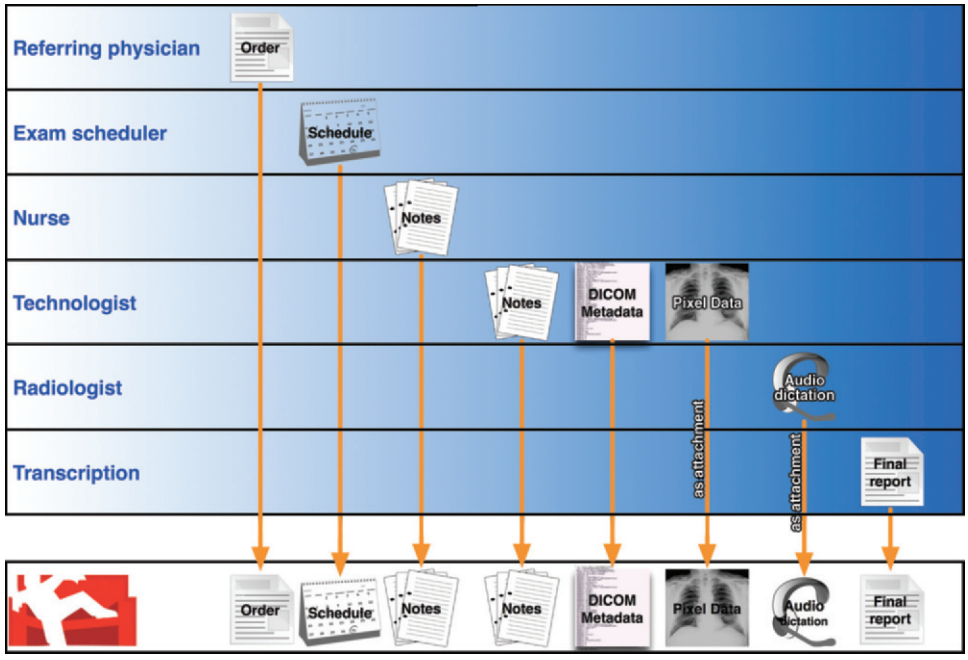

Figure 4. Schema shows that all data generated at any step in the fulfillment of a radiologic imaging order can be stored in the CouchDB database, regardless of whether the data are structured or unstructured, textual or digital (binary).

proof of identification (ID) and insurance, notes from the nurse or technologist, DICOM metadata and pixel data, audio notes from the radiologist, and the textual radiology report. All these documents can be associated with a patient's ID number (eg, a medical record number or, for larger deployments, a national ID number) and indexed by using map-reduce views with the patient's ID number used as the key.

Replication among Multiple Radiology Sites

The replication functionality of CouchDB allows robust distribution of stored information by using standard HTTP protocols. The storage of DICOM metadata with related pixel data in the database allows effective image distribution across departments, sites, and institutions.

Strategies for healthcare image and document distribution across institutions (collectively known as health information exchange) have been studied extensively, and several models have been proposed (7). The traditional, "federated" model for health information exchange relies on a centralized record locator, such as an Enterprise Master Patient Index, that manages "pointers" to the source information, which stays within the generating entity. CouchDB replication can synchronize all copies of a database across all entities so that there is no single point of failure. In case of a network outage that isolates a generating entity, the latest version of a patient's record, from just before the outage, is still available to all users within the network.

Centralized databases that allow the sharing of information across multiple institutions have been around for many years, but CouchDB provides distributed replication that synchronizes changes automatically in all database instances, with practically no need for special configuration. Because the same database version is distributed simultaneously to all users, there is no need for a central registry such as that created for Integrating the Healthcare Enterprise (IHE®) Cross-Enterprise Document Sharing.

Because the CouchDB database is a "master" copy that retains full functionality and can replicate even under conditions of inconsistent network access, multiple replication scenarios are possible. This flexibility in replication may prove particularly useful for the distribution of medical images in rural or underserved areas with poor Internet connectivity. The ability to replicate only selected information from the database allows complex multi-institutional use (eg, in teleradiology scenarios) (Fig 5).

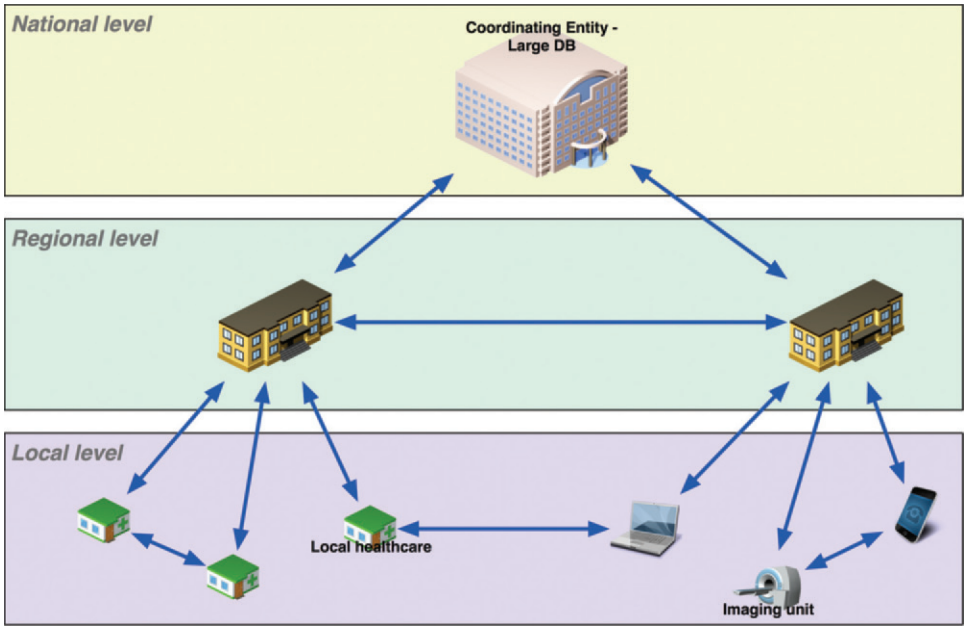

Figure 5. Schema shows an image distribution scenario for which document-based databases are well suited. Local imaging services can store DICOM objects and replicate them to a local database or a regional hospital database. From regional repositories, the DICOM objects may be replicated in turn to a national coordinating entity. Every CouchDB instance is a "master" database that functions independently and replicates horizontally and vertically with ease. *DB* = database.

Imaging Research Repositories

Imaging is used with increasing frequency in research, not only within the field of radiology but also in clinical trials of interventions and medications. The images generated during such trials often must strictly adhere to technical parameters to maintain homogeneity across subjects. The recording of all DICOM information in document-based databases allows easy comparison of technical parameters and storage of the research data, which can then be mined against the DICOM data to gain new insights. The anonymization of patients can be performed by editing the relevant attributes directly, as in RDBMS-based PACS, or data replication can be tailored so that only attributes not containing protected health information are synchronized and only anonymized datasets are replicated for specific users. For multisite imaging research studies, the flexible replication capability of CouchDB would facilitate the creation of data repositories with the necessary architecture for central institutional or governmental review.

Quality Assurance and Protocol Optimization

Adherence to acquisition protocols is an important part of quality assurance in any radiology department. DICOM metadata include information about the acquisition parameters of a particular imaging study or image series, which may vary from those of the established protocols. Great variability in a particular parameter over several studies may indicate poor compliance with institutional protocols. The storage of all DICOM metadata in a document-based database allows queries based on individual acquisition parameters, and, thus, the generation of plots showing parameter variability, which could provide insight into the adequacy of established protocols. Many other parameters also could be investigated, such as section thickness at CT or pulse sequence, echo time, repetition time, and flip angle at MR imaging.

Radiology Department Management

DICOM metadata stored in document-based databases also could be used to answer other, more general management questions within a radiology department. Possible data queries for management purposes might include "computed radiography imaging plate ID number" (to obtain data about the frequency of use of a particular imaging plate); "Modality Performed Procedure Step beginning and ending times" (to obtain data about departmental asset management); and radiology information system–type queries such as "studies by day/month/year," "study types by equipment," and "study counts by referring physician."

Radiation Exposure Monitoring

As mentioned earlier, monitoring of radiation exposures has become increasingly important, particularly among patients in vulnerable age groups, as data have emerged about the potential mutagenic effects of cumulative radiation doses over time. Document-based databases that store full DICOM metadata allow comprehensive monitoring with the use of specific queries based on well-defined tags such as peak voltage (0018,0060), exposure time (0018,1150), tube current (0018,1151), CT exposure sequence group (0018,9321), volume CT dose index (0018,9345), and relative x-ray exposure (0018,1405).

CouchDB Implementations

What follows is a brief summary of our initial experience in creating two CouchDB-based medical imaging databases and testing their functionality. More rigorous and in-depth experimentation is still needed to assess particular features. A more comprehensive experiment to evaluate the feasibility of this technology for clinical use would probably involve a multisite institutional trial in which two parallel RDBMS-based PACS at different institutions interface with each other and with a radiology information system to receive both radiologic images and radiology reports. Transmission times for standard DICOM transmissions with C-Move and CouchDB transmissions with HTTP would be compared, disk storage requirements under identical conditions would be determined, replication in real-world and artificially degraded network conditions would be tested, and filtered replication methods based on authentication would be explored.

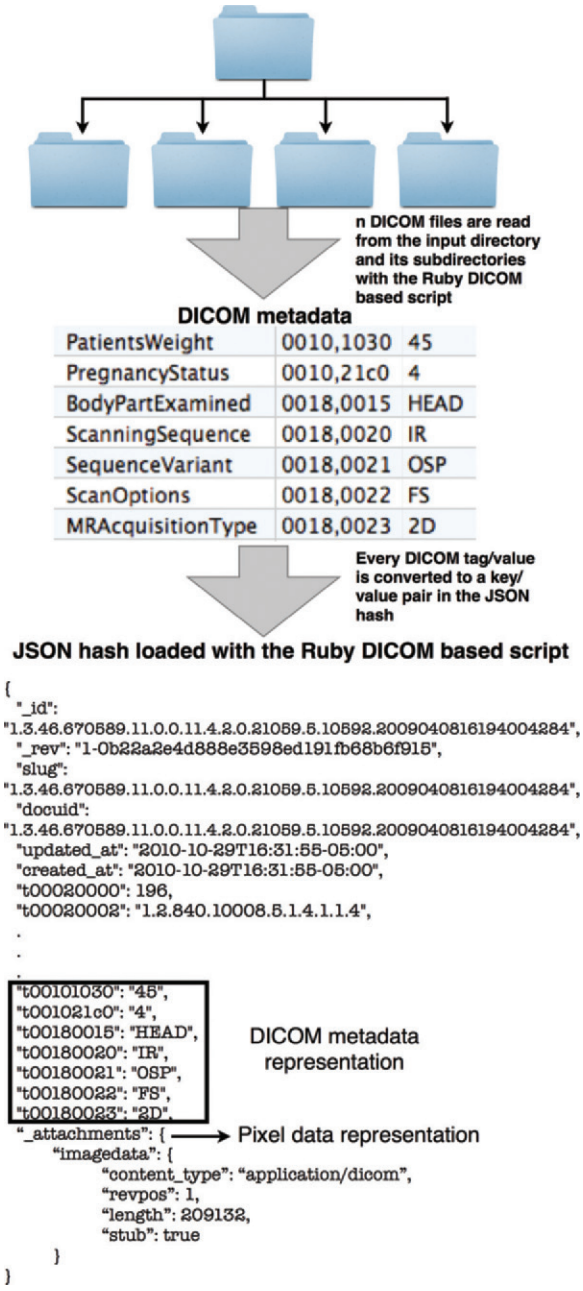

Figure 6. Schema shows the process used to generate JSON files for insertion into the CouchDB database. First, the Ruby loading script read the DICOM metadata in every document file and converted each DICOM tag and value to a key-value pair, producing a "JSON hash." Next, the pixel data were saved as attachments that were referenced in the relevant documents.

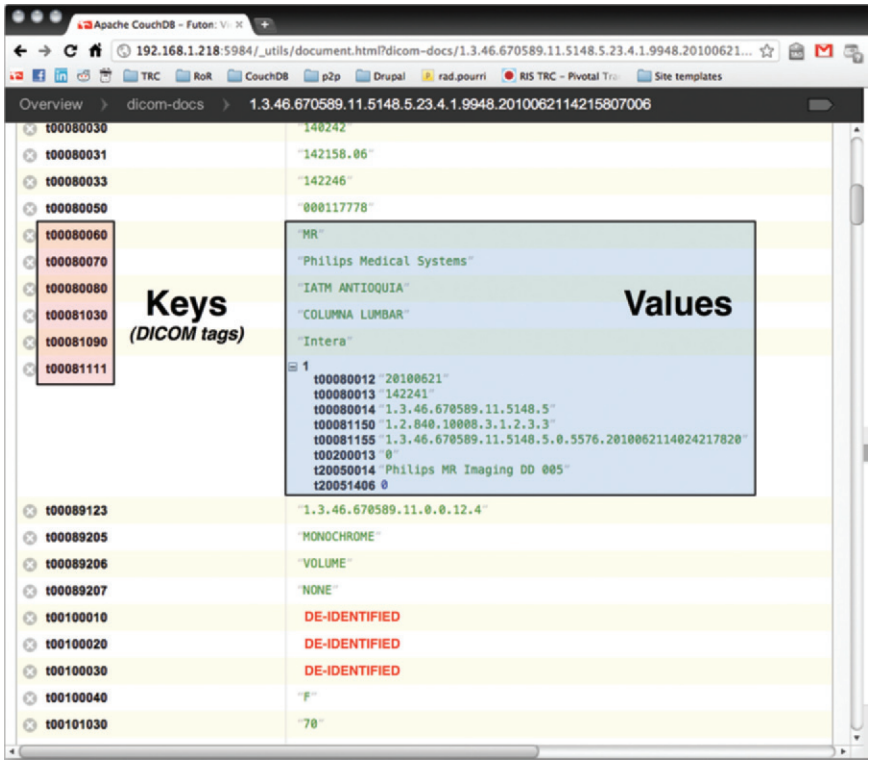

Figure 7. Screenshot shows a DICOM instance displayed as a document by the CouchDB Futon administration tool. The DICOM tags (keys) appear in the red shaded box, and the values of each tag appear to the right, on the same line as the tag, in the blue shaded box. (Courtesy of the Apache Software Foundation.)

Creating DICOM Documents in CouchDB

DICOM metadata can be loaded into a CouchDB instance in any language or tool that supports HTTP requests. We created a short script in the Ruby programming language that leverages the functionality of the Ruby DICOM library (8) for reading each file in a DICOM directory and converting it to a standard JSON object, as well as that of the CouchREST HTTP wrapper in the CouchDB application program interface (9) for formatting the JSON hash to be used in a CouchDB request. The loading script reads each tag–data element value in the DICOM metadata and creates a JSON object with the key-value pairs suitable for CouchDB insertion (Fig 6). The DICOM tags are passed to the database as nested JSON hashes by executing an HTTP PUT request to the CouchDB server. The script also labels each document for loading into the database with an ID number based on the DICOM tag that encodes the unique identifier (UID) for the service-object pair (ie, SOP) instance, a procedure designed to prevent the duplication of document ID numbers. The final DICOM structure represented in the document can be easily visualized by using Futon, the CouchDB administration tool (Fig 7).

The DICOM file itself is saved as an attachment that is referenced in an "_attachments" attribute of the document. The attachment is a JSON structure consisting of the name of the attachment, the content type (MIME type), and the attachment data in Base64-encoded string.

The code for our loading script is available at *https://github.com/simonmd/couchdicom*. At the time of this writing, there is to our knowledge only one other implementation of CouchDB for DICOM tag loading; it is based on a script written in Python (10).

Populating the Databases

A total of 5880 anonymized studies generated between January and October 2010 were extracted from one of our imaging centers' PACS as uncompressed DICOM files (for compatibility) and exported for loading into CouchDB. Two databases were created in CouchDB by using the

Table 2
CouchDB View Benchmarks

Query Type	Retrieval Time (msec)*	
	CouchDB	MySQL
Patients filtered by study date	56	23
Studies filtered by patient ID number	60	17
Series filtered by study UID	64	37
Instances filtered by series UID†	15	45

*Retrieval times are for the first 1000 results.
†This view is produced by using the map function without the reduce function.

Ruby loading script: One database was generated to explore the ability of CouchDB to handle attachments to documents. For this database, 906 documents were created to which the related DICOM file pixel data were saved as attachments in both DICOM and JPEG formats. This database contained 1.4 GB of data. A second, larger database (1,949,753 documents; 40.6 GB) was generated by using only the DICOM information, without attachments, for exploring metadata queries and aggregations.

Testing the Databases

All tests were performed on an Apple 2-GHz Core2duo iMac with 4 GB of RAM. We designed multiple views representing some of the most common queries sent to PACS from imaging system workstations: studies by date (day/month/year), studies by patient ID number, studies by patient name, and—to simulate WADO (*W*eb *A*ccess to *D*ICOM *O*bjects) queries—studies by modality and study UID, by series UID, and by instance UID.

The performance of the larger CouchDB database was measured against that of the open-source SQL-based DICOM archive DCM4CHEE (11). To allow a direct comparison of database performance, we designed and executed the same four queries in both SQL and map-reduce formats: For the DCM4CHEE archive, the MySQL command line tool was used to author queries; for the CouchDB database, the cURL command line tool was used to generate HTTP requests (Table 2).

Using CouchDB Views to Query the Database.—CouchDB uses "views" to query information based on user input. CouchDB views are stored sets of results from map-reduce functions written in JavaScript and are similar to views in standard RDBMS, which consist of stored queries that are accessible as virtual tables. The main difference between CouchDB views and RDBMS views (apart from the query language) is that CouchDB views are auto-incremental, which obviates a search of the entire dataset every time the same query is executed and thus drastically improves data retrieval times. CouchDB views can be tailored to retrieve any information queried, including but not limited to common queries used in the radiology work flow, and are equivalent to the views executed in a DICOM query-retrieve operation. CouchDB views are executed by implementing an HTTP GET request (Fig 8). After a view is run for the first time, the index is updated incrementally as new documents arrive. To obtain a particular record in subsequent map-reduce queries, the view can be further refined by including in the URL a "key" parameter that acts as an additional filter.

In our database tests, we compared the time needed for view generation with CouchDB against that required for an equivalent MySQL query to a SQL database containing the same imaging data in the RDBMS-based PACS. CouchDB views took up to 6 hours to run while

Figure 8. Screenshot shows the CouchDB view defined in JavaScript for a query of all studies by patient name. The two lines of text above the screen display show the HTTP syntax used to retrieve views for all patients (first line) and for an individual patient by appending the patient-specific key (second line). (Courtesy of the Apache Software Foundation.)

the index was being created the first time they were executed for the large database. However, subsequent queries were almost instantaneous, as shown in Table 2. When interpreting these benchmarks, it is important to bear in mind that the SQL database schema accommodated only a few DICOM tags, whereas the CouchDB database handled all the DICOM metadata for almost 2 million documents.

Testing Database Replication.—Replication of DICOM objects in CouchDB was shown to be reliable and predictable. Documents containing DICOM metadata and files containing pixel data were effortlessly replicated to other databases as attachments with a simple HTTP request, even with network interruptions (when "continuous replication" was selected, replication was seamlessly resumed when the network was restored). Replication in CouchDB is always unidirectional, but we achieved bidirectional synchronization (with changes mirrored to all copies of the database) by initiating a replication request in the opposite direction.

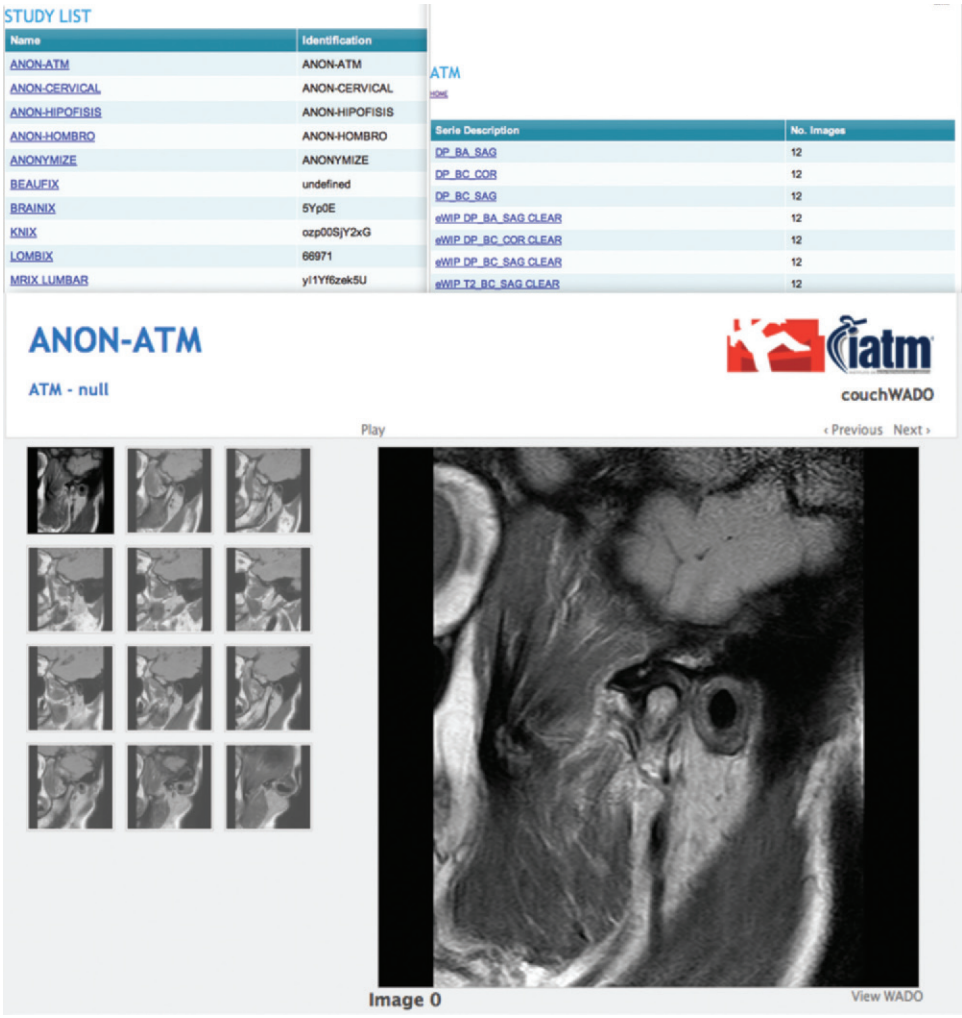

Figure 9. Screenshot demonstrates the use of a WADO application embedded in CouchDB for querying and retrieving DICOM images from the document database by using standard HTTP or WADO-type requests. The application allows queries based on study UID, series UID, or instance UID and retrieval of a single image saved as an attachment to a document.

Replication of a subset of the data (filtered replication) is also possible, although it is not described in detail here. Filtered replication functionality refers to the possibility of transferring a subset of documents selected on the basis of filtering criteria. CouchDB allows the inclusion of a filter function in the source database, allowing the replication of specific documents only. This function is useful when documents' availability must be restricted according to site, user, or another criterion.

Testing a WADO Application for Managing Document Attachments.—WADO is a method, defined in the DICOM standard, for accessing DICOM objects via the Web either as DICOM documents or JPEG files. To test the attachment management and Web access capabilities of CouchDB, we created CouchWADO, a Web-based application that combines WADO functionality with features of CouchDB. CouchWADO provided an intuitive interface with which we explored and navigated through the studies in the CouchDB DICOM database while the database was responding to standard WADO requests. CouchWADO allowed us

to transform views (queries) and retrieve selected documents by using the embedded functions "list" and "show" (4–6,12).

Although WADO requires that the URL contain the study UID, series UID, and instance UID and returns only one DICOM object at a time, WADO functionality was easily extended in CouchDB to show an HTML page with the list of patients, series, and studies that were available for selection. By using the CouchDB operations "view," "list," and "show," we navigated through multiple layers of information to locate and select a particular image (Fig 9). The CouchWADO application showcases the close relationship between Web and database functionality that is particular to CouchDB. The code for this simple application can be found at *https://github.com/simonmd/couchWADO*.

Conclusions and Future Directions

RDBMS are robust and reliable, and they have long been the backbone of PACS data archival. However, document-based databases may be better suited to store current medical imaging data volumes. Although these systems lack the maturity of RDBMS and their implementation demands a paradigm shift, they offer unique features that may prove useful in the healthcare imaging domain.

Larger multisite pilot studies of the use of document-based PACS would address the question of scalability. However, nonrelational databases are specifically designed for high-performance handling of large datasets, and it is reasonable to expect that document-based databases would gracefully accommodate much larger data volumes than those we tested.

Teaching Point

Document-based databases such as CouchDB offer the possibility of efficient storage of all DICOM metadata in a query-enabled environment that may be useful for data mining applications such as dose monitoring, quality assurance, and protocol optimization. By using efficient information search algorithms such as map-reduce, all DICOM metadata in a document-based database can be made available for examination with only a minimal difference in performance compared with that achieved by using traditional SQL-based stores.

Teaching Point

The use of document-based databases could substantially improve image management and distribution and should be seriously considered in the development of next-generation PACS. CouchDB, with its tightly coupled Web functionality, attachment handling, and replication abilities, provides an especially intriguing preview of the features we should aim for in future PACS storage solutions.

Acknowledgment.—We thank Christoffer Lervåg, author of the Ruby DICOM library, for his invaluable assistance in creating the loading script.

References

1. NoSQL. List of noSQL databases. http://nosql-database.org/. Accessed September 24, 2010.
2. Brink JA, Amis ES Jr. Image Wisely: a campaign to increase awareness about adult radiation protection. Radiology 2010;257(3):601–602.
3. Ghemawat DJ. MapReduce: simplified data processing on large clusters. OSDI 2004 [online serial]. Available at: research.google.com/archive/mapreduce.html. Accessed January 31, 2012.
4. Apache Software Foundation. CouchDB. http://wiki.apache.org/couchdb/. Version 1.1.0. Accessed September 30, 2010.
5. Anderson JC, Lehnardt J, Slater N. CouchDB: the definitive guide. Sebastopol, Calif: O'Reilly Media, 2009.
6. Lennon J. Beginning CouchDB. Berkeley, Calif: Apress, 2009.
7. Just BH, Durkin S. Clinical data exchange models: matching HIE goals with IT foundations. J AHIMA 2008;79(2):48–52.
8. Lervåg C. Ruby DICOM. https://github.com/dicom/ruby-dicom. Version 0.9.2. Accessed October 27, 2011.
9. CouchREST. https://github.com/couchrest/couchrest. Version 1.1.2. Accessed October 27, 2011.
10. Wallace M. DICOM and CouchDB. http://mikewallace.posterous.com/dicom-and-couchdb-0. Accessed March 2, 2010.
11. DCM4CHEE archive modules. http://www.dcm4che.org/confluence/display/ee2/Home. Version 2.16.1. Accessed October 27, 2011.
12. Strom C. Collating (not reducing) with CouchDB List Functions. http://japhr.blogspot.com/2010/02/collating-not-reducing-with-couchdb.html. Published February 10, 2010. Accessed September 30, 2010.

RadioGraphics

Informatics in Radiology
Use of CouchDB for Document-based Storage of DICOM Objects

Simón J. Rascovsky, MD, MSc • Jorge A. Delgado, MD • Alexander Sanz, BS • Víctor D. Calvo, BS • Gabriel Castrillón, BS

RadioGraphics 2012; 32:913–927 • **Published online** 10.1148/rg.323115049 • **Content Code:** IN

Page 915
DICOM objects, as tag-value documents, are inherently heterogeneous and cannot be represented adequately by RDBMS unless compromises are made. When DICOM objects are loaded into RDBMS, much of the metadata must be stripped out so that only information that is supported by the predefined database schema remains.

Page 918
Document-based database management systems are naturally more suitable than RDBMS for storing and retrieving DICOM objects, and being schema-less, they are free of the limitations of RDBMS.

Page 918
CouchDB (developed by the Apache Software Foundation, Los Angeles, Calif; available at *couchdb.apache.org*) is an open-source document-oriented database management system in which querying and indexing are performed by using JavaScript (4). We selected it as a model for testing because of its overall usability and practicality (5,6). The features of CouchDB dovetail with the current data management needs of medical imaging practices. Among the characteristics that make this system particularly appropriate for medical imaging data management are its operational reliance on common Web standards (HTTP, Representational State Transfer [REST]); its capability for managing attachments to documents; its functionality in regard to data replication; and its potential utility for data mining applications such as dose monitoring, quality assurance, and protocol optimization.

Page 927
Document-based databases such as CouchDB offer the possibility of efficient storage of all DICOM metadata in a query-enabled environment that may be useful for data mining applications such as dose monitoring, quality assurance, and protocol optimization.

Page 927
The use of document-based databases could substantially improve image management and distribution and should be seriously considered in the development of next-generation PACS.