INCORPORATING NoSQL INTO A DATABASE COURSE

Abdul Sattar, Torben Lorenzen and Keerthi Nallamaddi

This article introduces the concepts of Big Data and NoSQL and describes a semester long web-based project that uses both a relational database (Oracle 11g) and a NoSQL (MongoDB) database for an undergraduate database course. The relational database stores the record information of an online sales system and the NoSQL database stores three manuals. The semester long assignment and its implementation are available as a download.

1 Introduction

For the past several decades, commercial web applications have used relational database management systems (RDBMS) as the backend for their web sites. In recent years, web applications have begun processing staggering amounts of data. For example, Walmart handles more than 1 million customer transactions every hour. These transactions are imported into a database estimated to contain more than 2.5 peta (1015) bytes of data - the equivalent of 167 times the information contained in all the books in the US Library of Congress [1].The type of data that must be stored has also changed. Now it is necessary to store both traditional structured data and unstructured data such as pictures and product manuals. When data is enormous in size and/or unstructured in type, it is called "big data." The authors describe a semester long senior level undergraduate database project in which a traditional relational database system is complemented by a NoSQL database to handle unstructured data

2 A RDBMS Doesn't Work Well with Big Data

"The RDBMS data model assumes a centralized computing model – shared CPU, memory and disk. If the data for an application will not fit on a single server, an application will implement some form of data partitioning (sharding) to manually spread data across servers. There are three undesirable consequences to this approach. When you fill a shard, it is highly disruptive to re-shard because you have to change the sharding strategy in the application itself. To do joins across shards, you must run a query on every server and piece the results together in application software. Third, you can't do cross-node locking when making updates. So one must ensure all data that could need to be atomically operated on is resident on a single server, unless using an external TP monitor system or complex logic in the application itself. You have to create and maintain a schema on every server. If you have new information you want to collect, you must modify the database schema on every server, then normalize, retune and rebuild the tables. What was hard with one server is a nightmare across many [2]."

3 NoSQL, ACID, and CAP

NoSQL (an acronym for not only SQL) is a term coined by Carlo Strozzi in 1998 to describe a new database model where "the responsibility for modeling and data relationship management was shifted from the RDBMS into the application space [3]." Today NoSQL is an umbrella term for a host of database technologies including document store, key/value store, column oriented store, the graph database, and the object data base [4].

Some relational databases such as Oracle 11g store unstructured data as a special type. Oracle, a long time leader in RDBMS development, has signed on to the new technology and recently released its own NoSQL database. Marie-Anne Neimat, vice president of database development at Oracle, states that their NoSQL database "would be of interest to customers who are acquiring massive amounts of data who are unsure about the schema, who want more fluid capture of the data [5]." Craigslist has moved their data from a RDBMS to MongoDB [6]. Companies like Facebook, Twitter, Digg, Amazon, LinkedIn and Google all use NoSQL in one way or another [7].

This paper concentrates on MongoDB , a NoSQL technology based on the concept of a Key/Value pair. Specify a key and the associated value is returned by the NoSQL database. The key is paired with a data "blob" that can hold any data, such as blog posts, document files, songs etc. The search for the appropriate value is implemented via a hash table.

NoSQL does not adhere to the (RDBMS) model.

NoSQL does not use SQL as its query language.

It has a distributed, fault-tolerant architecture.

It may not give the full ACID guarantees of atomicity, consistency, isolation, and durability [8].

In Atomicity an entire transaction must be completed or the transaction fails—"All or Nothing." Consistency means a guarantee of database consistency. It ensures that the transaction will bring the database from one valid state to another based on all the rules—constraints, triggers, etc… Isolation certifies that each transaction happens independently of each other and unknown to each other. Durability signifies that once a transaction has been committed it will remain so [9].

- *Column:* these NoSQL databases are a hybrid between the NoSQL key-value stores and the classical row/column structure. They do not implement the strict rules that relational databases have [12].

4 MongoDB

MongoDB is an open source NoSQL database system. It is a document oriented database management system with dynamic schemas designed for high performance, ease of development and scalability. It can store rich data structures used with ecommerce content (categories, product types and their various attributes).

MongoDB is an open source NoSQL database system. It is a document oriented database management system with dynamic schemas designed for high performance, ease of development and scalability. It can store rich data structures used with ecommerce content (categories, product types and their various attributes).

ACID guarantees do not hold in a distributed computer system.

The CAP Theorem - proposed by Eric Brewer in 2000 and proven by Seth Gilbert and Nancy Lynch in 2002 - states that a distributed computer system can only simultaneously provide two out of the three guarantees of Consistency, Availability and Partition Tolerance at the same time. "In the Document-Oriented storage (NoSQL) realm, like MongoDB and Berkeley DB, we'd see Consistency and Partition Tolerance being applied [10]."

This CAP theorem really comes to life as an application scales. Smaller scale systems with low volume transactions won't see much of a difference. Allowing tiny latencies in a database for the sake of consistency has no noticeable effect on the overall performance of the system. But, as your system scales upward with increased activity, these consistency requirements will fail with errors and limit growth. Internet commerce would suffer greatly if user experience was impacted by long waits or errors, especially if errors were encountered during payment processing. Amazon stated that one tenth of a second longer response on their site will cost them one percent in sales. Even Google noticed that just a half a second increase in latency caused their traffic to drop by one fifth. This theorem is more than just a better way of doing things, it means money [11].

NoSQL databases store data in one of the following ways:

- *Key-Value Store:* a hash table is used where there is a unique key with a pointer to a specific data object. This is a very simple and easy implementation.
- *Document Store:* similar to the key-value store, but requires that the data document object be in a specific format, such as JSON, BSON, XML, etc. The document objects are usually other key-value collections. This allows for nested values to be associated with each key. MongoDB is a document store database.
- *Graph:* data is stored in tuples that have multiple attributes. Data that is best represented as a graph such as road maps, network topologies, and social networks, are prime candidates for graph NoSQL databases.

- A Mongo **system** holds a set of databases.
  A **database** holds a set of collections. A collection is analogous to a table in a RDBMS but it does not enforce any schema.
- A **collection** holds a set of documents. The documents within a collection may not need the exact same set of fields, but typically all documents in a collection have a similar or related purpose for an application.
- A **document** is a set of fields. All data are documents in MongoDB. Documents are the default representation of most user accessible data structures in the database.
- A **field** is a key-value pair.
- A **key** is a name (a string).
- A **value** is a basic type like string, integer, float, timestamp, binary, etc., a document, or an array of values.

MongoDB provides client drivers for many languages such as C, C++, Java, .NET, JavaScript, and PHP. There are also many third party frameworks and libraries available to interface with MongoDB from these languages. Many large corporations and organizations have implemented a NoSQL MongoDB database. It is being utilized in Craigslist, SAP (PAAS), Disney, MTV Networks, SourceForge, Open Sky, Shopwiki, Intuit, Athena Capital, bit.ly, IGN, EA, Justin.tv, Shutterfly, Forbes and the New York Times to just name a few.

5 Will Relational Databases Become Obsolete?

MongoDB's chief developer and 10gen CEO said "NoSQL is very agile and scalable but is no substitute for SQL when doing complex transactions. Enterprise customers will probably adopt a mixed bag of database technologies — both SQL and NoSQL — for different workloads [13]."

6 Inclusion of NoSQL into a Database Course:

Here is the outline of our current 15 week long course:

Introduction - Basic Concepts	[week 1]
Modeling - Design – Implementation	[week 2]
Entity-Relationship Model	[week 3, 4]
Relational Data Model	[week 4, 5]
Relational Algebra	[week 6, 7]
SQL	[week 8]
Normalization	[week 9]
Oracle Basics	[one class in week 10]
New database model-NoSQL(MongoDB)	[week 10, 11]
Web Interfaces: the front end Code is passed to the class	[week 12]
Final Project	[week 13, 14, 15]

In weeks 10 and 11, we introduce NoSQL and MongoDB. This material replaces the previous study of the internals of a DBMS (such as storage, query processing and query optimization).

7 Final Project: The SalesSystem [SQL + NoSQL]

The class is assigned to build a hybrid system for a hypothetical firm, ABC Electronics. The company sells electronic products to customers who submit online orders. Data regarding customers and products are kept in a relational database and the product manuals are stored in a NoSQL database.
Oracle 11g (personal edition) stores the record data of the Sales System using five tables (indicated by a bold underlined primary key and italicized foreign keys):

PRODUCT_CATEGORY	(**Code**, Description)
PRODUCTS	(**ProductId**, *Code*, Description, AvailableQuantity, Cost, ListPrice)
CUSTOMERS	(**CustomerNo**, Fname, Lname, Address, Bdate, Email, HomePhone, Mobile)
ORDERS	(**OrderNo**, *CustomerNo*, OrderDate, ShipDate, CancelDate)
ORDER_ITEMS	(**OrderNo**, **ProductId**, Quantity, Price)

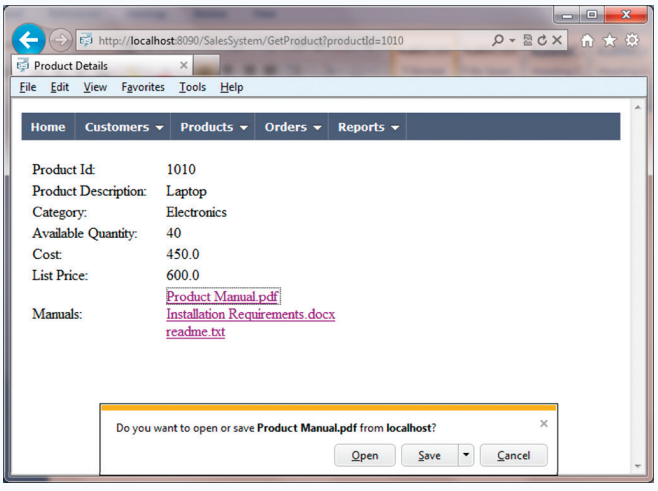
This system provides an option to cancel an order if it has not yet been shipped. Sales reports can be generated by Product Category and by Customer between any two dates. Since the user specifies keys to retrieve various product manuals, a noSQL technology based on the concept of a Key/Value store would be the most appropriate implementation choice for this feature of our application. Since MongoDB is used in real world applications, is free and is installed by simply downloading a zip file and double clicking the executable, a pedagogical choice to use that software was made.

Install and Run MongoDB on Windows

To install MongoDB, download the appropriate binary zip file from www.mongodb.org/downloads and extract to your chosen directory. Download the MongoDB java driver from www.mongodb.org and add to the classpath. Then connect to MongoDB using one of the following:

- Connect to a local host and to a default port
  Mongo m = new Mongo();
- Connect to a remote host and port 25125.
  Mongo m = new Mongo("mongodbhost", 25125);
- To connect to "mydb" database
  DB db = m.getDB("mydb");
- To authenticate:
  bool auth = db.authenticate(username, password);
- Get a collection named "employees":
  DBCollection coll = db.getCollection("employees")
- Inserts are the basic method to add data to a database:
  Insert a document with a firstName of John and a lastName of Smith into the 'employees' collection:
  BasicDBObject doc = new BasicDBObject();
  doc.put("firstName", "John");
  doc.put("lastName", "Smith");
  coll.insert(doc);
- The find() method is used to query a database:
  Find all documents with a lastname of Smith:
  BasicDBObject query = new BasicDBObject();
  query.put("lastName", "Smith");
  DBCursor cur = coll.find(query);
  while(cur.hasNext()) {
  System.out.println(cur.next()); }

SalesSystem Screenshot: Data from both databases



Software and Tools
Database: Oracle 11g Express Edition, MongoDB
Java SE 6, Java EE 6
JDBC Driver: ojdbc6.jar
IDE(s): Eclipse 3.7, Oracle SQL Developer 3.0
Application Server: JBoss AS7 Community Edition

Outcomes and Conclusions
"NoSQL data stores will not be a 'passing fad.' The simplicity, flexibility, and scalability of these systems fills a market niche, e.g. for web sites with millions of read/write users and relatively simple data schemas. Even with improved relational scalability, NoSQL systems maintain advantages for some applications [14]." Students get exposure to the principles and concepts of database design that are essential for building well-designed database applications and also become familiar with the new NoSQL technology.

Download
Download the source code and installation guide from http://webhost.bridgew.edu/sattar/DB.zip **lr**

References

[1] http://en.wikipedia.org/wiki/Big_data#cite_note-0
[2] http://www.couchbase.com/sites/default/files/uploads/all/whitepapers/NoSQL-Whitepaper.pdf
[3] http://www.versant.com/docs/default-document-library/nosql--beyond-the-key-value-store-for-the-enterprise_wp_low_res.pdf?sfvrsn=2
[4] http://www.slideshare.net/benoitperroud/nosql-overview-implementation-free
[5] http://news.techworld.com/applications/3313166/oracle-launches-nosql-database-for-unstructured-data/
[6] http://www.mongodb.org/about/production-deployments/
[7] http://newtech.about.com/od/databasemanagement/a/Nosql.htm
[8] http://en.wikipedia.org/wiki/ACID
[9] http://en.wikipedia.org/wiki/ACID
[10] http://lpd.epfl.ch/sgilbert/pubs/BrewersConjecture-SigAct.pdf
[11] http://www.julianbrowne.com/article/viewer/brewers-cap-theorem
[12] http://www.mongodb.org/display/DOCS/Introduction
[13] http://www.zdnet.com/blog/open-source/mongodb-chief-it-will-be-mixed-sql-nosql-world/9297
[14] Cattell, R. "Scalable SQL and NoSQL Data Stores," SIGMOD Record, (39,4, 2010).

ABDUL SATTAR, TORBEN LORENZEN, AND KEERTHI NALLAMADDI
Department of Mathematics and Computer Science
Bridgewater State College
Bridgewater, Massachusetts 02325 USA
Sattar@bridgew.edu
Lorenzen@bridgew.edu
knallamaddil@student.bridgew.edu

Categories and Subject Descriptors: H. [Information Systems]; H.2 [Database Management]
General Terms: RDBMS, Query languages, Languages
Keywords: Database design, SQL, NoSQL, Java, Oracle, MongoDB

DOI: 10.1145/2465085.2465100

SLI: A TOOL FOR EASING THE UNDERSTANDING OF AUTOMATED PROOF CONSTRUCTION

José A. Maestro-Prieto and M. Aránzazu Simon-Hurtado

This article describes the experience obtained from teaching computational logic in an introductory course for undergraduates. The proposed teaching approach helps students to reach a deeper level of understanding in first order logic representation, computational logic procedures and automated theorem proving. The article includes the description of SLI, a theorem prover with a graphical output, a feature appreciated by students.

1 Introduction

Logic, as a core topic in several studies (e.g., philosophy, mathematics, computer science), continues to attract the attention of research focused on learning. Automating the process for obtaining a proof for first order logic (FOL) was a landmark in the development of artificial intelligence (AI). This signaled the achievement of the automation of a reasoning mechanism, something that was considered a proof of genuine intelligence, no longer reserved to humans. Automated theorem proving appears in the standard