A Comparison of SQL and NoSQL Databases

Keith W. Hare
JCC Consulting, Inc.
Convenor, ISO/IEC JTC1 SC32 WG3
Abstract

NoSQL databases (either no-SQL or Not Only SQL) are currently a hot topic in some parts of computing. In fact, one website lists over a hundred different NoSQL databases.

This presentation reviews the features common to the NoSQL databases and compares those features to the features and capabilities of SQL databases.
Who Am I?

- Muskingum College, 1980, BS in Biology and Computer Science
- Senior Consultant with JCC Consulting, Inc. since 1985 – high performance database systems
- Ohio State – Masters in Computer & Information Science, 1985
- SQL Standards committees since 1988
- Vice Chair, INCITS H2 since 2003
- Convenor, ISO/IEC JTC1 SC32 WG3 since 2005
Topics

- SQL Databases
  - SQL Standard
  - SQL Characteristics
  - SQL Database Examples

- NoSQL Databases
  - NoSQL Definition
  - General Characteristics
  - NoSQL Database Types
  - NoSQL Database Examples
Standard SQL

The following is a short, incomplete history of the SQL Standards – ISO/IEC 9075

- 1987 – Initial ISO/IEC Standard
- 1989 – Referential Integrity
- 1992 – SQL2
  - 1995 SQL/CLI (ODBC)
  - 1996 SQL/PSM – Procedural Language extensions
- 1999 – User Defined Types
- 2003 – SQL/XML
- 2008 – Expansions and corrections
- 2011 (or 2012) System Versioned and Application Time Period Tables
SQL Characteristics

- Data stored in columns and tables
- Relationships represented by data
- Data Manipulation Language
- Data Definition Language
- Transactions
- Abstraction from physical layer
Applications specify what, not how

Query optimization engine

Physical layer can change without modifying applications

- Create indexes to support queries
- In Memory databases
Data Manipulation Language (DML)

- Data manipulated with Select, Insert, Update, & Delete statements
  - Select T1.Column1, T2.Column2 ...
    From Table1, Table2 ...
    Where T1.Column1 = T2.Column1 ...

- Data Aggregation
- Compound statements
- Functions and Procedures
- Explicit transaction control
Data Definition Language

- Schema defined at the start
- Create Table (Column1 Datatype1, Column2 Datatype2, ...)
- Constraints to define and enforce relationships
  - Primary Key
  - Foreign Key
  - Etc.
- Triggers to respond to Insert, Update, & Delete
- Stored Modules
- Alter ...
- Drop ...
- Security and Access Control
Transactions – ACID Properties

- **Atomic** – All of the work in a transaction completes (commit) or none of it completes.
- **Consistent** – A transaction transforms the database from one consistent state to another consistent state. Consistency is defined in terms of constraints.
- **Isolated** – The results of any changes made during a transaction are not visible until the transaction has committed.
- **Durable** – The results of a committed transaction survive failures.
SQL Database Examples

- Commercial
  - IBM DB2
  - Oracle RDMS
  - Microsoft SQL Server
  - Sybase SQL Anywhere

- Open Source (with commercial options)
  - MySQL
  - Ingres

Significant portions of the world’s economy use SQL databases!
NoSQL Definition

From www.nosql-database.org:

Next Generation Databases mostly addressing some of the points: being non-relational, distributed, open-source and horizontal scalable. The original intention has been modern web-scale databases. The movement began early 2009 and is growing rapidly. Often more characteristics apply as: schema-free, easy replication support, simple API, eventually consistent / BASE (not ACID), a huge data amount, and more.
NoSQL Products/Projects

http://www.nosql-database.org/ lists 122 NoSQL Databases

- Cassandra
- CouchDB
- Hadoop & Hbase
- MongoDB
- StupidDB
- Etc.
NoSQL Distinguishing Characteristics

- Large data volumes
  - Google’s “big data”
- Scalable replication and distribution
  - Potentially thousands of machines
  - Potentially distributed around the world
- Queries need to return answers quickly
- Mostly query, few updates
- Asynchronous Inserts & Updates
- Schema-less
- ACID transaction properties are not needed – BASE
- CAP Theorem
- Open source development
BASE Transactions

- Acronym contrived to be the opposite of ACID
  - **Basically Available,**
  - **Soft state,**
  - **Eventually Consistent**

- Characteristics
  - Weak consistency – stale data OK
  - Availability first
  - Best effort
  - Approximate answers OK
  - Aggressive (optimistic)
  - Simpler and faster
Brewer’s CAP Theorem

A distributed system can support only two of the following characteristics:

- Consistency
- Availability
- Partition tolerance

The slides from Brewer’s July 2000 talk do not define these characteristics.
Consistency

- all nodes see the same data at the same time – Wikipedia
- client perceives that a set of operations has occurred all at once – Pritchett
- More like Atomic in ACID transaction properties
Availability

- Node failures do not prevent survivors from continuing to operate – Wikipedia
- Every operation must terminate in an intended response – Pritchett
Partition Tolerance

- the system continues to operate despite arbitrary message loss – Wikipedia
- Operations will complete, even if individual components are unavailable – Pritchett
NoSQL Database Types

Discussing NoSQL databases is complicated because there are a variety of types:

- **Column Store** – Each storage block contains data from only one column
- **Document Store** – stores documents made up of tagged elements
- **Key-Value Store** – Hash table of keys
Other Non-SQL Databases

- XML Databases
- Graph Databases
- Codasyl Databases
- Object Oriented Databases
- Etc…
- Will not address these today
NoSQL Example: Column Store

- Each storage block contains data from only one column

- Example: Hadoop/Hbase
  - Yahoo, Facebook

- Example: Ingres VectorWise
  - Column Store integrated with an SQL database
  - http://www.ingres.com/products/vectorwise
Column Store Comments

- More efficient than row (or document) store if:
  - Multiple row/record/documents are inserted at the same time so updates of column blocks can be aggregated
  - Retrievals access only some of the columns in a row/record/document
NoSQL Example: Document Store

- Example: CouchDB
  - BBC

- Example: MongoDB
  - [http://www.mongodb.org/](http://www.mongodb.org/)
  - Foursquare, Shutterfly

- JSON – JavaScript Object Notation
CouchDB JSON Example

```json
{
    "_id": "guid goes here",
    "_rev": "314159",
    "type": "abstract",
    "author": "Keith W. Hare",
    "title": "SQL Standard and NoSQL Databases",
    "body": "NoSQL databases (either no-SQL or Not Only SQL) are currently a hot topic in some parts of computing.",
    "creation_timestamp": "2011/05/10 13:30:00 +0004"
}
```
CouchDB JSON Tags

- "_id"
  - GUID – Global Unique Identifier
  - Passed in or generated by CouchDB

- "_rev"
  - Revision number
  - Versioning mechanism

- "type", "author", "title", etc.
  - Arbitrary tags
  - Schema-less
  - Could be validated after the fact by user-written routine
NoSQL Examples: Key-Value Store

- Hash tables of Keys
- Values stored with Keys
- Fast access to small data values
- Example – Project-Voldemort
  - [http://www.project-voldemort.com/](http://www.project-voldemort.com/)
  - Linkedin
- Example – MemCacheDB
  - [http://memcachedb.org/](http://memcachedb.org/)
  - Backend storage is Berkeley-DB
Map Reduce

- Technique for indexing and searching large data volumes
- Two Phases, Map and Reduce
  - Map
    - Extract sets of Key-Value pairs from underlying data
    - Potentially in Parallel on multiple machines
  - Reduce
    - Merge and sort sets of Key-Value pairs
    - Results may be useful for other searches
Map Reduce

- Map Reduce techniques differ across products
- Implemented by application developers, not by underlying software
Map Reduce Patent

Google granted US Patent 7,650,331, January 2010
System and method for efficient large-scale data processing

A large-scale data processing system and method includes one or more application-independent map modules configured to read input data and to apply at least one application-specific map operation to the input data to produce intermediate data values, wherein the map operation is automatically parallelized across multiple processors in the parallel processing environment. A plurality of intermediate data structures are used to store the intermediate data values. One or more application-independent reduce modules are configured to retrieve the intermediate data values and to apply at least one application-specific reduce operation to the intermediate data values to provide output data.
Storing and Modifying Data

- Syntax varies
  - HTML
  - Java Script
  - Etc.
- Asynchronous – Inserts and updates do not wait for confirmation
- Versioned
- Optimistic Concurrency
Retrieving Data

- Syntax Varies
  - No set-based query language
  - Procedural program languages such as Java, C, etc.
- Application specifies retrieval path
- No query optimizer
- Quick answer is important
- May not be a single “right” answer
Open Source

- Small upfront software costs
- Suitable for large scale distribution on commodity hardware
NoSQL Summary

- NoSQL databases reject:
  - Overhead of ACID transactions
  - “Complexity” of SQL
  - Burden of up-front schema design
  - Declarative query expression
  - Yesterday’s technology

- Programmer responsible for
  - Step-by-step procedural language
  - Navigating access path
Summary

- **SQL Databases**
  - Predefined Schema
  - Standard definition and interface language
  - Tight consistency
  - Well defined semantics

- **NoSQL Database**
  - No predefined Schema
  - Per-product definition and interface language
  - Getting an answer quickly is more important than getting a correct answer
**HOW TO WRITE A CV**

1. **DO YOU HAVE ANY EXPERTISE IN SQL?**
   - **NO**

2. **geek & poke**

3. **DOESN'T MATTER. WRITE: "EXPERT IN NO SQL"**

---

**Leverage the NoSQL boom**
Questions?
Web References

- “NoSQL -- Your Ultimate Guide to the Non-Relational Universe!”
  http://nosql-database.org/links.html

- “NoSQL (RDBMS)”
  http://en.wikipedia.org/wiki/NoSQL

  Dr. Eric A. Brewer. Professor, UC Berkeley. Co-Founder & Chief Scientist, Inktomi.
  www.eecs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf

  http://www.julianbrowne.com/article/viewer/brewers-cap-theorem

- “How to write a CV” Geek & Poke Cartoon
Web References

- “Graph Databases, NOSQL and Neo4j” Posted by Peter Neubauer on May 12, 2010 at: http://www.infoq.com/articles/graph-nosql-neo4j
Web References


- “Scalable SQL”, ACM Queue, Michael Rys, April 19, 2011 [http://queue.acm.org/detail.cfm?id=1971597](http://queue.acm.org/detail.cfm?id=1971597)

Books