CSCI235 Database Systems

NoSQL, NewSQL, and Other Database Systems

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

NoSQL, NewSQL, and Other Database Systems

Outline

XML Database Systems

The Extensible Markup Language (XML) was defined by the WWW Consortium (W3C) as a document markup language in late 1990s

XML has become a major format for data exchange, flexibly structured documents, or configuration files

XML document consists of nested XML elements

XML element consists of start tag, data, and end tag

XML element may contain other XML elements

XML element can be described by the attributes located within start tag

Attributes provide a meta description or identification of XML element

XML document my contain additional information like version declarations, entity declarations, comments, and processing instructions

XML namespaces (XMLNS)are useful to have unique naming of elements

Namespace is usually Uniform Resource Identifier (URI) which denotes scope and origin of the document

XML Database Systems

Sample XML document

XML
```xml
<?xml version="1.0" encoding ="UTF-8"?>
<!DOCTYPE booking SYSTEM "/xml-resources/dtd/booking.dtd">
<booking>
  <hotel ID="Hongkong Sheraton">
    <rooms>
      <room>
        <category>suite</category>
        <price>1234</price>
      </room>
      <room>
        <category IDREF="Sydney Intercontinental">standard</category>
      </room>
    </rooms>
  </hotel>
</booking>
```

XML Database Systems

Document Type Definition (DTD) is the simplest way to specifiy a schema for XML documents

XML Schema is more sophisticated schema language that DTD

XML Schema allows for data typing withing a system defined types, createing user defined types, and inheritance

XML Schema suppoprts constraints on the number of occurences of sublements, on maximu, minimum values, and an adanced ID referencing

XML has 3 query/transformation languages:

- XPath: A langauge for navigation in XML document
- XQuery: A complex query and transformation language that uses XPath as navigational tool
- XSLT: A query and transformation language

XML Database Systems

Native XML database systems: XML document management systems

- eXistDB: Java-based open source XML database; supports indexing, XPath, XQuery, XML:DBI (Java API)
- BaseX: Java-based system that stores XML documents in a tabular representation; support indexing, conversion to JSON, XML:DB API

XML support in relational a database systems: columns of type XMLType, e.g. ANSI SQL definition (SQL/XML) and Oracle, Postgres, DB2, MySQL, and SQL Server

NoSQL, NewSQL, and Other Database Systems

Outline

Key-value Stores and Document Database Systems

Key-value pair is a tuple of two strings `("key","value")`

A key-value store stores key-value pairs

A value is retrieved by specifiying a key

A key-value store is a protottype of a schemaless database system

It is possible to put any key-value into a store and no retrictions are enforced on a format or structure of a value

A key-value store offers three operations: put, get, and delete

A value in key-value pair main contain other key-value pairs or sequences of keys

Key-value Stores and Document Database Systems

Dynamo DB is a key-value database implemented by Amazon

Architectural characteristics:

- The system uses a hash value of a key to determine the nodes in a cluster responsible for the key; nodes can be added and removed with minimal rebalancing overhead
- To determine one of n nodes in cluster where a key-value is saved hash(key) is computed modulo n
- Addition of a new node is performed by re-hashing
- An application can determine trade-offs between consistency, read performance and write performance
- It is possible to determine strong consistency, eventual consistency, and weak consistency
- Since write operations are never blocked it is possible that there will be multiple versions of an object in the system
- Versions can be merged by a data store itself or sometimes need to be resolved by the application or user
- For instance, if the buyer updates his or her shopping cart from two computers, the resulting cart may have duplicate items that he or she may need to remove

Key-value Stores and Document Database Systems

A document database stores and retrieves documents, which can be in JSON, BSON, etc format

Documents are self-describing, hierchical data structures

The structures of documents are similar to each other but not exactly the same

Documents are stored as the value part of key-value store

JSON
```
{
 "firstname": "James",
 "likes": ["biking", "hiking", "viking"],
 "city": "Boston",
 "friend": null,
 "addresses":[{"state": "NSW", "city": "Sydney"},
              {"state": "Vic", "city": "Melbourne"} ]
}
```

Key-value Stores and Document Database Systems

Other key-value stores and document database systems:

- Riak: it groups key-value pairs called as Riak objects into logical units called buckets; buckets can be configured by defining bucket type (specific configuration for a bucket); the system comes with several options to configure a storage backend
- Redis: it is an adavnced in-memory key-value store with command line interface; it supports several data types and data structures like strings, linked lists, unsorted sets, sorted sets, hash maps, bit arrays
- MongoDB: it is a document database with BSON as its storage format
- CouchDB: it is an Erlang based document database; it stores JSON documents in so called databasesand it supports multi-version concurrency control; it exposes conflict handling to te users: a user has to submit the most recent revision number of a document when writing data into it
- Couchbase: it is a document database that merges the features of CouchDB and Membase; it stores JSON documents and other formats like binary and serialized data in buckets; each document is tored under a unique key; it implements multiversion concurrency control
- Other other systems: ArangoDB, OrientDB, RavenDB, RethinkDB

NoSQL, NewSQL, and Other Database Systems

Outline

Graph databases

Graph databases store entities and relationships between the entities

Nodes are instances of objects in the applications

Edges have properties and have directional significance

Graph data model

```
Node james = graphDb.createNode();
james.setProperty("name", "James");
Node harry = graphDb.createNode();
harry.setProperty("name", "Harry");
james.createRelationahip(harry, FRIEND);
harry.createRelationahip(james, FRIEND);
```

Graph databases ensure consistency through ACID-compliant transactions

Graph databases use domain specific languages for traversing graph structures

Properties of nodes can be indexed

A commonly used technique for horizontal scaling is sharding

Any link-rich domain is well suited for graph databases

Graph databases

Implementation of logical graph data model in a relational database system requires indexing to speed up processing of join operations (index lookups)

Efficient real-time graph processing requires a way to move through the graph structure without index lookups (index-free adjacency)

In index-free adjacency implementation of graph data model each node knows the physical location of all adjacent nodes

There is no need to use indexes to efficiently navigate the graph, because each node “points” to all adjacent nodes

It creates the problems with distribution of data among many servers

Graph compute engines enable graph processing that needs to work across distributed datasets

Graph databases

Neo4j is a Java-based graph database based on Property Graph data model

Property Graph data model provides a richer model for representing complex data than RDF (presented later) by associating both nodes and relationships with attributes

Neo4j supports billions of nodes, ACID compliant transactions, and multiversion consistency

Neo4j implements a declarative graph query language Cypher comparable to SQL or SPARQL (query language for RDF)

Graph databases

The following sequence of statements creates the data nodes and edges in Neo4j database

```
CREATE (TheMatrix:Movie {title:'The Matrix',                                Neo4j
                         released:1999,
                         tagline:'Welcome to the Real World'})
```

```
CREATE (JohnWick:Movie {title:'John Wick',                                  Neo4j
                        released:2014,
                        tagline:'Silliest Keanu movie ever'})
```

```
CREATE (Keanu:Person {name:'Keanu Reeves', born:1964})                      Neo4j
CREATE (AndyW:Person {name:'Andy Wachowski', born:1967})
```

```
CREATE                                                                      Neo4j
(Keanu)-[:ACTED_IN {roles:['Neo']}]->(TheMatrix),
(Keanu)-[:ACTED_IN {roles:['John Wick']}]->(JohnWick),
(AndyW)-[:DIRECTED]->(TheMatrix)
```

Graph databases

The following queries retrieve information from Neo4j database

```
MATCH (kenau:Person {name:"Keanu Reeves"})                Neo4j
RETURN kenau;
```

```
MATCH (kenau:Person {name:"Keanu Reeves"})                Neo4j
-[:ACTED_IN]-
(movie)<-[:ACTED_IN]-(coStar)
RETURN coStar.name;
```

```
MATCH (kenau:Person {name:"Keanu Reeves"})                Neo4j
-[*1..2]-
(related)
RETURN distinct related;
```

Graph databases

Resource Description Framework (RDF late 1990) is a Web standard developed for modeling of Web resources and the relationships between them

Information in RDF is expressed as triples subject:predicate:object

```RDF
TheMatrix:Is:Movie
Kenau:Is:Person
Kenau:StarredIn:TheMatrix
```

RDF was intended as the means of creating a formal database of resources on the web together with the dependencies that these resources relied on

RDF graphs can be stored in a variety of formats including XML, or even within tables inside a relational database

A widely used query language forRFD graphs is SPARQL that has an SQL-like syntax

Graph databases

The other most popular systems include:

- InfiniteGraph: it is an enterprise distributed graph database implemented in Java and build on former object-orented database ObjectivityDB
- OrientDB: it is the first multi-model NoSQL database system with a graph database engine; it provides graph, document, key-value, and object view of data
- FlockDB: it is is an open source distributed, fault-tolerant graph database for managing wide but shallow network graphs; it is not designed for multi-egde graph traversal but rather for rapid set operations on graph connections
- AllegroGraph: it is a closed source triple store which is designed to store RDF triples; it supports transactions, automatic indexing, querying with SPARQL and reasoning with RDFS++ and Prolog
- Ontotext: it provides RDF view of data for content management, knowledge discovery and semantic search
- GraphDB: it is a semantic graph database, compliant with W3C Standards; it provides a view of a database as RDF triplestores

Graph databases

The other most popular systems include:

- StarDog: it allows to query, search, and analyze enterprise data using knowledge graph technology; it supports unification with relational database systems, ACID transaction processing, query answering for location aware (geospacial) graphs
- Oracle Spatial and Graph: it includes high performance, enterprise-scale, commercial spatial and graph database and analytics for Oracle Database 12 it supports large-scale Geographic Information Systems (GIS)
- Apache Giraph: it is an iterative graph processing framework, built on top of Apache Hadoop;
- GraphX: it unifies Extract Transform and Load (ETL), performs exploratory analysis, and iterative graph computation within a single system; it provides a view of the same data as both graphs and collections

NoSQL, NewSQL, and Other Database Systems

Outline

Column stores

Column store (or columnar database management system) is a database management system (DBMS) that stores data tables by column rather than by row

Practical use of a column store versus a row store differs little in the relational DBMS world

To reduce overhead imposed DML operations columnar databases generally implement some form of write-optimized delta store

Delta store is optimized for frequent writes like tables stored in row-by-row format

Delta store is merged with the main columnar-oriented store

The merge will occur periodically, or whenever the amount of data in the delta store exceeds a threshold

Prior to the merge, queries access both the delta store and the column store in order to return complete and accurate results

Column stores

KDB was the first commercially available column-oriented database developed in 1993 followed in 1995 by Sybase IQ

Sybase IQ (mid 1990) ("Intelligent Query") data warehousing platform implemented relational tables in column-by-column mode

C-Store and later on Vertica (2005-2011) implemented a native columnar database

MonetDB was released under an open-source license in 2004

Vertica was eventually developed out of C-Store, while the MonetDB related project evolved into VectorWise

Druid is a column-oriented data store that was open-sourced in late 2012 and now it is used by numerous organizations

NoSQL, NewSQL, and Other Database Systems

Outline

Extensible Record Stores

Extensible Record Store organizes data into extensible tables

Extensible table consists of rows

Each row is uniquely identified by a row key

Data within a row is grouped by a column family

Column families have an important impact on the physical implementation of extensible table

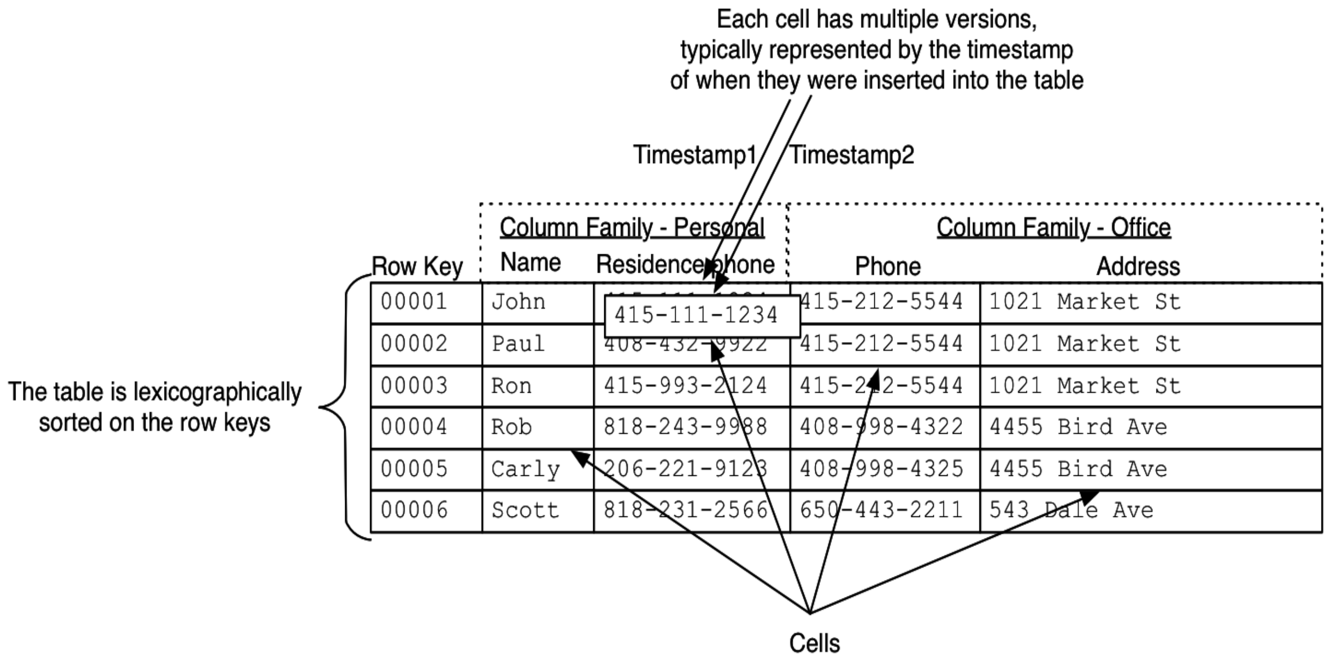
Every row has the same column families although some of them can be empty

Data within a column family is addressed via its column qualifier, or simply, column name

Hence, a combination of row key, column family, and column qualifier uniquely identifies a cell

Values in cells do not have a data type and are always treated as sequences of bytes

Extensible Record Stores


Each cell has multiple versions, typically represented by the timestamp of when they were inserted into the table
Timestamp1
Timestamp2
Column Family - Personal
Column Family - Office
The table is lexicographically sorted on the row keys
Cells

Row Key	Name	Residencephone	Phone	Address
00001	John	[illegible] / 415-111-1234	415-212-5544	1021 Market St
00002	Paul	408-432-9922	415-212-5544	1021 Market St
00003	Ron	415-993-2124	415-212-5544	1021 Market St
00004	Rob	818-243-9988	408-998-4322	4455 Bird Ave
00005	Carly	206-221-9123	408-998-4325	4455 Bird Ave
00006	Scott	818-231-2566	650-443-2211	543 Dale Ave

Values within a cell have multiple versions

Versions are identified by their version number, which by default is a timestamp when the cell was written

Extensible Record Stores

If a timestamp is not determined at write time, then the current timestamp is used

If a timestamp is not determined during a read, the latest one is returned

The maximum allowed number of cell value versions is determined for each column family

Foundation of Extensible Record Stores is Google's system called Big Table

The other systems include

- Apache Cassandra: it stores column families in so called "keyspace"; cretae table command creates a new column family in the keyspace; indexes can be creted on columns
- Apache HBase: it stores tables in namespaces; it offers command line interface, SQL interface called Phoenix; and Java API
- Hypertable:it stores tables in namespaces;it provides a a query language HQL similar to SQL
- Apache Accumulo: it also stores tables in namespaces and it also provides a comman line interface and Java API

NoSQL, NewSQL, and Other Database Systems

Outline

Object-Oriented and Object-Relational Database Systems

Object-Oriented database systems have its roots in a concept of persistent programming language that distinguishes between transient (RAM memory) data structures and persistent (HDD, SSD memory) data structures

Object-Oriented database systems allow to store and to operate on both persistent and transient objects

Object Definition Language (ODL) can be used to define classes of objects, properties, associations, generalization and aggregation hierarchies, and methods

Object Query Language (OQL) provides SQL-like syntax to express queries on instances of objects in a database

One of the approaches to implementation of persistently stored objects was to use conventional relational database systems as an underlying storage engine

Such approach required Object-Relational mapping of object data model into tabular data model

Object-Oriented and Object-Relational Database Systems

Object-Relational mapping provides a view of a database as a collection of tables filled with objects instead of rows and it is called as Object-Relational data model

Object-Relational database system a database is a collection of tables (containers) filled with objects that contains references to other objects in the same or in te other tables (containers)

Object-Relational Mapping (ORM) tools are avaible for a wide range of Object-Oriented programming languages

The standards for Java object persistence are Java Persistence API (JPA) and Java Data Objects (JDO) API

Pure Object-Oriented database systems are almost non existent any more unless we consider JSON data model based systems as partial implementation of Object-Oriented database systems

NoSQL, NewSQL, and Other Database Systems

Outline

In-Memory Database Systems

Solid state disk (SSD) technology has allowed for implementation of in-memory database systems

SSDs store bits of information in cells

A single-level cell (SLC) contains one bit of information per cell, while a multi-level cell (MLC) SSD contains usually only two but sometimes three bits per each cell

Cells are arranged in pages of about 4K and pages are arranged in blocks that typically contain 256 pages

Read operations, and initial write operations, require only a single-page IO

Changing the contents of a page requires an erase and overwrite of a complete block, i.e. it is two times slower than read

Many traditional relational databases perform relatively poorly on SSDs

This is because critical IO has been isolated to sequential write operations for which hard disk drives perform well, but that represent the worst possible workload for SSD

In-Memory Database Systems

When a transaction commits in an ACID-compliant relational database, the transaction record is usually written immediately to a sequential transaction log (called a redo log in some databases)

When the log is moved to SSD performance does not improve because of lesser SSD write performance

SSD based systems (Aerospike) use main memory to store indexes to the data while keeping the data always on flash

A more paradigm-shifting trend has been the increasing practicality of storing complete databases in main memory (RAM)

Architectural problems of main memory databases:

- Cache-less architecture: there is no point caching in memory what is already stored in memory
- Alternative persistence model: data in memory disappears when the power is turned off

In-Memory Database Systems

In-memory databases either

- replicate data to other members of a cluster or
- write complete database images to disk files or
- write out out transaction/operation records to an append-only disk file (called a transaction log or journal)

The most popular in-memory systems include:

- TimesTen: (owned by Oracle since 2005); it is entirely based on transactional relational database architecture; it can be used as a standalone in-memory database or as a caching database supplementing the traditional disk-based Oracle RDBMS; when the database is started, all data is loaded from checkpoint files into main memory; periodically or when required database data is written to checkpoint files
- Redis: (owned by VMware since 2013); it is an in-memory key-value store; it uses a snapshot files to store the copies of the entire system at a given point in time; append only file keeps a journal of changes that can be used to “roll forward” the database from a snapshot in the event of a failure; configuration options allow the users to configure writes to the append files after every operation, at one-second intervals, or based on operating-system-determined flush intervals

In-Memory Database Systems

The most popular in-memory systems include:

- SAP HANA: (owned by SAP, 2010) it is a relational database that combines in-memory technology with a columnar storage option, installed on an optimized hardware configuration; relational tables in HANA can be configured for row-oriented or columnar storage; on start-up, row-based tables and selected column tables are loaded from the database files and other column tables can be loaded on demand; reads and writes to row-oriented tables are applied directly; updates to column-based tables are buffered in the delta store; queries against column tables must read from both the delta store and the main column store
- VoltDB: it is a hybrid (in-memory/HDD) ACID-compliant relational database based on shared nothing (sharding) architecture; it supports SQL access from within pre-compiled Java stored procedures; it relies on horizontal partitioning down to the individual hardware thread to scale
- DB2 with BLU acceleration: (owned by IBM) it is a hybrid relational database that stores frequently used data on SSD; it allows for better performance at a price closer to HDD based systems than purely SDD based systems, e.g. Oracle's Exadata database machine

NoSQL, NewSQL, and Other Database Systems

Outline

Data Stream Processing Systems

A data stream is an infinite sequence of transient values and when recorded persistent values

Data streams can be produced by sensor networks or by network traffic monitoring, stock exchange monitoring, etc

Data Stream Management System (DSMS) processes data streams by interminably running continuous queries over and over again

DSMS must handle the queries that might be running for days,months, or even years

DSMS must aggregate data from the entire stream, e.g. overall average of all values, or from a subset of data called as sliding window, e.g. an average of values in the last 30 days

Majority of DSMS are based on the relational data model and their continuus query languages are similar to SQL

In such a case data items from the streams is represented as a pair (timestamp,tuple) where tuple corresponds to a row in an infinite relational table

NoSQL, NewSQL, and Other Database Systems

Outline

Array Database Systems

Array Database Systems organize data alonmg multiple dimensions and can be used to store and to manipulate data with complex structures, e.g. astronomical data obtained from satellite observations

In the Array Data Model data are stored in multidimensional arrays

Each cell in an array contains a tuple of certain length

The elements of such tuple can be either atomic values or arrays by themselves

Array Data Model can express an arbitrary number of array nestings

Tuples are addressed by specifying the corresponding dimensions

In an Array Database System SciDB individual scalar values can be addresses by named attributes

Array Database Systems offer specialized functions to manipulate array data, like different types of aggregations, operation on matrices, and even joins

NoSQL, NewSQL, and Other Database Systems

Outline

NewSQL Database Systems

A term NewSQL describes adoption of the design principles underlying NoSQL systems to build new distributed relational database systems with SQL interface

Such re-design of conventional relational DBMSs would make then scalable, i.e. it would be possible to store relational data in a number of server while efficiently processing SQL queries

Additionally, NewSQL systems suppose to transparently handle hardware failures in a network while still maintaining ACID compliant transction protocols

The first type of NewSQL systems are completely new database platforms designed to operate in a distributed cluster of shared-nothing nodes, in which each node owns a subset of the data, e.g. Google Spanner, CockroachDB, Clustrix, VoltDB, MemSQL, NuoDBand, and Trafodion

The second type are highly optimized storage engines for SQL that provide the same programming interface as SQL, but scale better than built-in engines, e.g. MySQL Cluster, Infobright, TokuDB, and MyRocks

References

Wiese L. Advanced Data Management for SQL, NoSQL, Cloud and Distributed Databases, De Gruyter Oldenburg, 2015

Harrison G., Next Generation Databases NoSQL NewSQL Big Data, Apress, 2015