Making Sense of NoSQL
Dan McCreary
Wednesday, Nov. 13th 2014
Agenda

• Why NoSQL?
• What are the key NoSQL architectures?
• How are they different from traditional RDBMS Systems?
• What types of problems do they solve?
• How to learn more
Background for Dan McCreary

• Co-founder of the NoSQL Now! conference
• Background
  – Bell Labs
  – NeXT Computer (Steve Jobs)
  – Owner of 75-person software consulting firm
  – US Federal data integration (National Information Exchange Model NIEM.gov)
  – Native XML/XQuery for metadata management since 2006
  – Advocate of web standards, NoSQL and XRX systems
  – As of Monday – Principal Engineer with MarkLogic

Copyright Kelly-McCreary & Associates, LLC
Making Sense of NoSQL

- Coauthor (with Ann Kelly) of "Making Sense of NoSQL"
- Guide for managers and architects
- Focus on NoSQL architectural tradeoff analysis
- Basis for **40 hour course** on database architectures
- http://manning.com/mccreary

Copyright Kelly-McCreary & Associates, LLC
The Story of Property Tax Forms

• How did I get into NoSQL?
• In 2006 a state agency in Minnesota wanted to standardize property tax records across 87 counties
• A story about standards
  – XML, XML Schema, XForms, XQuery, NEIM
• A story about agility
• A story about new technology adoption

Alex Bleasdale

Arun Batchu
2006 eCRV Case Study
<table>
<thead>
<tr>
<th>#</th>
<th>Data Element Name</th>
<th>Status</th>
<th>Primary Owner Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Activity</td>
<td>assigned-to-review-team</td>
<td>NIEM-universal</td>
</tr>
<tr>
<td>2</td>
<td>ActivityDate</td>
<td>assigned-to-review-team</td>
<td>NIEM-universal</td>
</tr>
<tr>
<td>3</td>
<td>ActivityEndDate</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>4</td>
<td>ActivityFederalFiscalYear</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>5</td>
<td>ActivityFiscalYear</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>6</td>
<td>ActivityStartDate</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>7</td>
<td>ActivityStateFiscalYearNumber</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>8</td>
<td>Address</td>
<td>assigned-to-review-team</td>
<td>NIEM-universal</td>
</tr>
<tr>
<td>9</td>
<td>AddressCityName</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>10</td>
<td>AddressLine1Text</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>11</td>
<td>AddressLine2Text</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>12</td>
<td>AddressPostalCodeID</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>13</td>
<td>AddressStateCode</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>14</td>
<td>AngularMinute</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>15</td>
<td>AngularSecond</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>16</td>
<td>Contact</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>17</td>
<td>ContactEmailID</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>18</td>
<td>ContactFAXText</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>19</td>
<td>ContactPhoneText</td>
<td>initial-draft</td>
<td>DataStandards</td>
</tr>
<tr>
<td>20</td>
<td>CRV</td>
<td>initial-draft</td>
<td>CRV</td>
</tr>
<tr>
<td>21</td>
<td>CRVAdjustmentCode</td>
<td>initial-draft</td>
<td>CRV</td>
</tr>
<tr>
<td>22</td>
<td>CRVCountyAuditorID</td>
<td>initial-draft</td>
<td>CRV</td>
</tr>
</tbody>
</table>
**Object:** CRV

**Definition:** A certificate of real estate value document that must be filed with the county auditor whenever real property valued over $1,000 is sold or conveyed in Minnesota.

**Complex:** true

**Subclass Of:** DocumentForm

**Abbreviation:** CRV

**Screen Label:** CRV

**Metadata Source:** Back of form PE20, Minnesota Department of Revenue glossary

The Minnesota Department of Revenue uses information on the CRV to determine if assessors through Minnesota are valuing property according to the same standards, and to determine how much state aid will go to all school districts and cities in the next year. The value of the real property in all school district and city affects the amount of financial aid the state will provide.

**Usage:**

**Referenced in:** DataStandards

**General Note:** Information reported on the CRV includes the sales price, the value of any personal property, if any, included in the sale, and the financial terms of the sale. The CRV is eventually filed with the Property Tax Division of the Department of Revenue. The deed types must be warranty deed, contract for deed, quit claim deed, trustee deed, executor deed or probate deed. If the value of the property is less than $1,000 the deed must have the following written on that back: The sale price or other consideration given for this property was $1,000 or less.

**Enumerated:** false
Legal Description of Minnesota Property Being Transferred

County Information:

- County: Dakota
- Sales Agreement County Allocation Amount: 20.0

Description of Minnesota Property Being Transferred:

- Legal description: A sample Legal description:
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris porttitor pellentesque leo. Suspendisse quam massa, sodales id, tempor et, semper eget, purus. Mauris sed lacus vitae lacus tempor iaculis. Ac eu scelerisque massa, dapibus varius, iaculis non, sagittis gravida.

Homestead Status:

- Will the buyer use this property as their principal residence?:
  - Yes
  - No

Primary Parcel ID:

- Primary Parcel ID: AEC-001

Additional Parcel IDs:

- Additional Parcel ID (leave first one blank, if none): AEC-001-A
- Additional Parcel ID (leave first one blank, if none): ABC-001-B

[XForms from XML Schema!]

## Minnesota Certificate of Real Estate Valuation Form - editing CRV# 19-08-49

### Current Workflow for this CRV Document

<table>
<thead>
<tr>
<th>ID</th>
<th>Date/Time</th>
<th>User</th>
<th>Org</th>
<th>Code</th>
<th>Activity</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2008-01-14, 08:42</td>
<td>anonymous</td>
<td>anonymous</td>
<td>original</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2008-01-14, 08:44</td>
<td>dakota</td>
<td>dakota</td>
<td>county-accepted</td>
<td>County Accepted CRV</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2008-01-14, 08:45</td>
<td>dakota</td>
<td>dakota</td>
<td>activity</td>
<td>CRV edited and saved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2008-01-14, 08:47</td>
<td>dakota-as</td>
<td>dakota</td>
<td>assessor-assigned</td>
<td>assigning to assessor</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2008-01-15, 10:33</td>
<td>dakota</td>
<td>dakota</td>
<td>activity</td>
<td>CRV edited and saved</td>
<td></td>
</tr>
</tbody>
</table>

### Add new Workflow event

- **CRV Workflow Code:**
- **County Code Text** (only relevant on county code selected):

### Tags

<table>
<thead>
<tr>
<th>ID</th>
<th>Org</th>
<th>Date/Time</th>
<th>Keyword</th>
<th>Comment</th>
<th>Work</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dakota</td>
<td>2008-01-14, 08:44</td>
<td>county-accepted</td>
<td>CRV has been accepted and automatically tagged by the accept process</td>
<td>2</td>
</tr>
</tbody>
</table>
Four Translations

- $T_1$ – HTML into Java Objects
- $T_2$ – Java Objects into SQL Tables
- $T_3$ – Tables into Objects
- $T_4$ – Objects into HTML
Kurt's Suggestion

Use a Native XML Database!

Web Browser

Web Form

Save

Kurt Cagle

eXist-db

store($collection, $file-name, $data)

Equivalent of 45 SQL inserts in 1 line of code!
Zero Translation

• XML lives in the web browser (XForms)
• REST interfaces
• XML in the database (Native XML, XQuery)
• XRX Web Application Architecture
• No "impedance mismatch", No translation!
• Department tried it and then went back to HTML, Java and SQL
• …but I was forever changed…I began to question everything I had been taught about databases
NoSQL on Google Trends

http://www.google.com/trends/explore?q=NoSQL%2C+RDBMS#q=NoSQL%2C%20RDBMS&cmpt=q
The NO-SQL Universe

Key-Value Stores
- riak
- redis
- EROSKIPE

Document Stores
- Couchbase
- MongoDB

Column-Family Stores
- MapR
- HBase
- Cassandra

Graph/Triple Stores
- neo4j
- AllegroGraph® RDFStore
- InfiniteGraph
- bigdata®

XML
- RAVENDB
- CouchDB
- existDB
- MarkLogic®
Sample of NoSQL Jargon

Document orientation
Schema free
MapReduce
Horizontal scaling
Sharding and auto-sharding
Brewer's CAP Theorem
Consistency
Reliability
Partition tolerance
Single-point-of-failure
Object-Relational mapping
Key-value stores
Column stores
Document-stores
Memcached

Indexing
B-Tree
Configurable durability
Documents for archives
Functional programming
Document Transformation
Document Indexing and Search
Alternate Query Languages
Aggregates
OLAP
XQuery
MDX
RDF
SPARQL
Architecture Tradeoff Modeling
ATAM
Erlang

Note that within the context of NoSQL many of these terms have different meanings!
Before NoSQL

Relational

Analytical (OLAP)
After NoSQL

Relational

Analytical (OLAP)

Key-Value

Column-Family

Graph

Document

Copyright Kelly-McCreary & Associates, LLC
Food for thought…

• What percentage of database transactions run on RDBMSs in the following organizations?

Google  
Yahoo!  
amazon.com  
facebook  
LinkedIn  
twitter

• What percentage of all transactions in Minnesota run on RDBMSs?
• Why is this number different?
• Is our data fundamentally different?
NoSQL – The Big Tent

- "NoSQL" – a label for a "meme" that now encompasses a large body of innovative ideas on data management
- "Not Only SQL"
- Focus on non-relational databases and hybrids
- A community where new ideas are quickly recombined to create innovative new business solutions

http://www.flickr.com/photos/morgennebel/2933723145/

Copyright Kelly-McCreary & Associates, LLC
Mainframe Era

- 1 CPU
- COBOL and FORTRAN
- Punchcards and flat files
- $10,000 per CPU hour

Commodity Processors

- 10,000 CPUs
- Functional programming
- MapReduce "farms"
- Pennies per CPU hour
Two Approaches to Computation

1930s and 40s

John von Neumann

Manage state with a program counter.

Alonzo Church

Make computations act like math functions.

Which is simpler? Which is cheaper? Which will scale to 10,000 CPUs?
### Standard vs. MapReduce Prices

<table>
<thead>
<tr>
<th>Standard On-Demand Instances</th>
<th>Amazon EC2 Price</th>
<th>Amazon Elastic MapReduce Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small (Default)</td>
<td>$0.06 per hour</td>
<td>$0.015 per hour</td>
</tr>
<tr>
<td>Medium</td>
<td>$0.12 per hour</td>
<td>$0.03 per hour</td>
</tr>
<tr>
<td>Large</td>
<td>$0.24 per hour</td>
<td>$0.06 per hour</td>
</tr>
<tr>
<td>Extra Large</td>
<td>$0.48 per hour</td>
<td>$0.12 per hour</td>
</tr>
</tbody>
</table>

Pricing for Amazon EC2 (On-Demand) and Amazon Elastic MapReduce

http://aws.amazon.com/elasticmapreduce/#pricing

Copyright Kelly-McCreary & Associates, LLC
MapReduce CPUs Cost Less!

Cut cost from 12 to 3 cents per CPU hour! Perhaps Alonzo was right!

Why? (hint: how "shareable" is this process)

http://aws.amazon.com/elasticmapreduce/#pricing
Pressures on Single Node RDBMS Architectures

- Large Data Sets
- OLAP/BI/Data Warehouse
- Document-Data
- Agile Schema Free
- Linked Data
- Social Networks
- Reliability
- Scalability
An evolving tree of data types

- RDBMS
- BI/DW
- Transactional
- Read/Write
- Structured
- Log Files
- Read Mostly
- Binary
- Graph
- Unstructured
- Web Crawlers
- Open Linked Data
- JSON
- XML
- Documents
Many Uses of Data

• Transactions (OLTP)
• Analysis (OLAP)
• Search and Findability
• Enterprise Agility
• Discovery and Insight
• Speed and Reliability
• Consistency and Availability
Three Eras of Enterprise Data

- NoSQL will not replace ERP or BI/DW systems – but they will complement them and also facilitate the integration of unstructured document data.
Simplicity is a Virtue

- Many modern systems derive their strength by dramatically limiting the features in their system and focus on a specific task.
- Simplicity allows database designer to focus on the primary business drivers.
- Simplicity promotes "separation of concerns".
Google MapReduce

- 2004 paper that had huge impact of functional programming on the entire community
- Copied by many organizations, including Yahoo
Google Bigtable Paper

Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber

{fay,jeff,sanjay,wilsonh,kerr,m3b,tushar,fikes,gruber}@google.com

Google, Inc.

- 2006 paper that gave focus to scaleable databases
- designed to reliably scale to petabytes of data and thousands of machines
Scale Up vs. Scale Out

Scale Up
• Make a single CPU as fast as possible
• Increase clock speed
• Add RAM
• Make disk I/O go faster

Scale Out
• Make Many CPUs work together
• Learn how to divide your problems into independent threads
Automatic Sharding

- When one node in a cluster has too much of a load the system should be able to automatically rebalance the data distribution.
- Note: Auto-sharding is not the same as replication!

Warning processor at 90% capacity!
Time to "Shard" – copy ½ data to a new processor

Each processor gets ½ the load

- When one node in a cluster has too much of a load the system should be able to automatically rebalance the data distribution.
- Note: Auto-sharding is not the same as replication!
"We can easily store the data that we actually get, not the data we thought we would get."
Horizontal Scalability

Linear scalable architectures provide a constant rate of additional performance as the number of processors increases.

Non-scalable systems reach a plateau of performance where adding new processors does not add incremental performance.

Figure 6.2
Shared Nothing Architecture

- Every node in the cluster has its own CPU, RAM and disk
The Master node may become a bottleneck in large clusters.
Many newer NoSQL architectures are moving toward a true peer-to-peer system.
• NoSQL systems that use many commodity processors can be precisely tuned to meet an organization's service level agreements.
Key-Value Stores

- Keys used to access opaque blobs of data
- Values can contain any type of data (images, video)

**Pros:** scalable, simple API (put, get, delete)

**Cons:** no way to query based on the content of the value

**Examples:** Berkley DB, Memcache, DynamoDB, S3, Redis, Riak
Column-Family

- Key includes a row, column family and column name
- Store versioned blobs in one large table
- Queries can be done on rows, column families and column names
- Pros: Good scale out, versioning
- Cons: Cannot query blob content, row and column designs are critical

Examples:
Cassandra, HBase, Hypertable, Apache Accumulo, Bigtable
The key is composed of:
- row id (string)
- Column family (grouping of columns)
- Column name (string)
- Timestamp (64-bit value)

Value
- any blob (byte stream)
Graph Store

• Data is stored in a series of nodes, relationships and properties
• Queries are really graph traversals
• Ideal when relationships between data is key:
  – e.g. social networks

Pros: fast network search, works with public linked data sets

Cons: Poor scalability when graphs don't fit into RAM, specialized query languages (RDF uses SPARQL)

Examples:
Neo4j, AllegroGraph, Bigdata triple store, InfiniteGraph, StarDog
Document Store

- Data stored in nested hierarchies
- Logical data remains stored together as a unit
- Any item in the document can be queried
- **Pros:** No object-relational mapping layer, ideal for search
- **Cons:** Complex to implement, incompatible with SQL

**Examples:**
MarkLogic, MongoDB, Couchbase, CouchDB, RavenDB, eXist-db
Two Models

"Bag of Words"

- All keywords in a single container
- Only count frequencies are stored with each word

"Retained Structure"

- Keywords associated with each sub-document component
Keywords and Node IDs

- Keywords in the reverse index are now associated with the **node-id** in every document
Using the Wrong Architecture

Credit: Isaac Homelund – MN Office of the Revisor
Find ways to remove barriers and empower the non programmers on your team.
Further Reading and Questions

Dan McCreary
dan@danmccreary.com

@dmccreary

Thank You!

http://manning.com/mccreary