

NoSQL

John Paul Ashenfelter
CTO/Transitionpoint

Why do you use SQL?

- ACID
- Reliable
- Strong ecosystem

SQL #fail?

- Scaling horizontally
- Performance
- Flexibility (fixed schema)
- Doesn't model domain

NOSQL

What is NoSQL?

- **Not Only SQL => NOSQL**
- web workloads, not OLTP or DW
- ecosystem that addresses #fail of sql

Why NoSQL?

- schema-free documents
- fast key lookups
- horizontal scaling
- REST APIs

Who's Using NoSQL?

- Digg
- Gilt
- Twitter
- Foursquare
- Amazon
- Google
- Yahoo
- LinkedIn
- Facebook

NoSQL Ecosystem

- Column stores
- Key value stores
- Document stores
- Graph databases

Column Stores

Column Store 4 | 1

- Column-oriented database
 - tuple of name, value, timestamp
 - supercolumn is a map of columns
 - columnfamily \sim table
- Stored on (custom) filesystem
- No joins, no sorting after insert

Why use a Column Store?

- very large (TB => PB) data set
- distributed (local => global) data set
- high volume of reads
- need for MapReduce-style processing

Column Store Implementations

- Google BigTable
- Hadoop
- Cassandra

Column Store Examples

- Google is canonical example
- Digg with Cassandra
- Twitter for analytics

Graph Databases

Graph Database 4 | 1

- Stores data in graphs (networks)
 - nodes
 - relationships
 - properties
- Traverse the graph to find the data
- Semi-structured

Why Use a Graph Database

- Problem domain maps to graph or network
 - social networks
 - collaborative filtering
- Sparse data

Graph Database Implementations

- neo4j

Graph Database Examples

Document Stores

Document Store 4 | 1

- document-oriented
 - tree of objects
 - objects have attribute values and lists
- similar to XML or JSON storage
- directly accesses structure
- similar to column store with N columns/

Why Use a Document Store

- schema-less (!!!!!)
- need data in JSON (or XML)
- read/write as a unit
- partitioning and replication for scaling

Document Store Implementations

- **Mongodb**
- **Couchdb**
- **Riak**

Document Store Examples

Key-Value Stores

Key-Value Store 4 | 1

- big hash table -- keys and values
- transparent partition and replication
- lower emphasis on transactions
- “eventual consistency”

Why Use A Key-Value Store

- scaling, scaling, scaling
- low-latency optimized
- high-throughput optimized

Key-Value Implementations

- Amazon SimpleDB
- Redis
- Memcachedb
- Tokyo Cabinet/Tyrant
- Amazon Dynamo
- Project Voldemort
- (Microsoft) Dynamite
- Mnesia
- Lawnchair for JS

Key-Value Store Example

CODE!

Where Can I Learn More

- MyNoSQL Blog
- Key Papers
- Conferences, particularly dynamic languages

ColdFusion and NoSQL

NoSQL Conferences

- no:sql(east)
- meetings (particularly Mongo)
- emerging tech conferences -- OSCON, Ruby/Python, JS

MyNoSQL Blog

- <http://nosql.mypopescu.com/>
- Lots of activity
- Not really CF-oriented, but Java pops up

Key papers

- Dynamo: [http://
www.allthingsdistributed.com/2007/10/
amazons_dynamo.html](http://www.allthingsdistributed.com/2007/10/amazons_dynamo.html)
- CAP Theorem: [http://
camelcase.blogspot.com/2007/08/cap-
theorem.html](http://camelcase.blogspot.com/2007/08/cap-theorem.html)
- Map/Reduce: [http://labs.google.com/papers/
mapreduce.html](http://labs.google.com/papers/mapreduce.html)

Thanks!

- Contact
 - johnpaul@transitionpoint.com
- Follow
 - @johnpaul
 - @transitionpoint
- Read
 - <http://www.ashenfelter.com>
 - <http://www.transitionpoint.com>