NoSQL Movement - A Survey

Senthil Vaiyapuri

April 2010

About

- Cisco :
  - Currently working on supportforums.cisco.com
  - Web Directory, PAWS, Support Library, Labview, Web Analytics
- Elsewhere :
  - Security Event Information Management
  - Oracle Database Kernel development, porting, support
  - IBM DB2 Systems Programmer

Agenda

- Graze the NoSQL landscape, a learning exercise
- Overview of Different Technologies
- Solicit for real world trials and usages (please share in future brown-bags)

"We back up our data on sticky notes because sticky notes never crash."

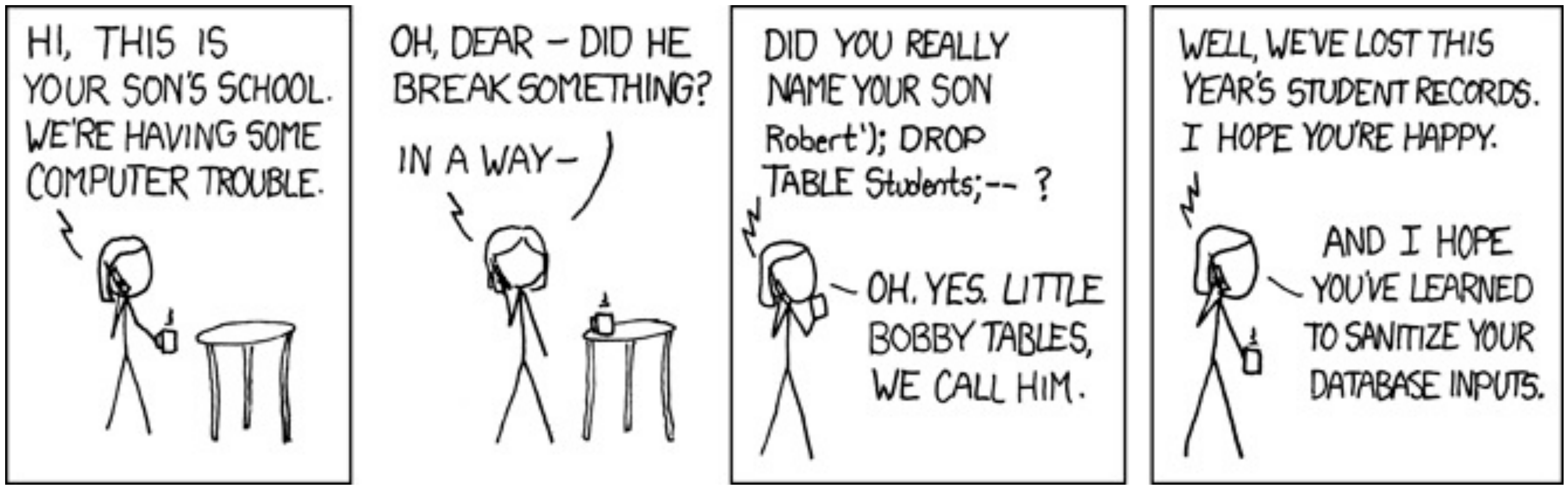
Pre SQL

- record oriented
- fixed/variable length records
- typically clients manage data layout
- access paths need to be coded by clients
- access paths are hard wired

Pre SQL

- ISAM/VSAM
- IMS
- IDMS
- DBM

SQL


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

SQL

- Data is modeled as entities and relationships
- ACID compliant
- Clients specify “What” and from “Where” not “How”
- SQL - Definition, Query, Manipulation
- Access paths are optimized and managed by Query Engine
- Physical data layout is managed by database engines
- Partitioning, Replication, HA, Objects, Rich Data - Features available
- Configuration heavy

SQL

- Postgres / MySQL
- Oracle / Informix / Sybase / Ingres
- IBM DB2
- DEC RDB
- Monet DB (Column Store)

NoSQL

So, how do I query the database?
It's not a database. It's a key-value store!
Ok, it's not a database. How do I query it?
You write a distributed map reduce function in Erlang!
Did you just tell me to go f[redacted]k myself?
I believe I did, Bob.

NoSQL

- Bad name choice, but catchy
- Very high data volumes
- Heavy and spikey read/write activities
- Few companies had proprietary infrastructure and software to deal with this kind of volume and workloads
- Availability of such infrastructure to public (cloud)
- Porting/Re-implementing and Availability of backing Software technologies as Open Source

No SQL - Features

- Performance/Scaling for internet scale workloads
- Sharding
- Replication (trivial to setup)
- High Availability
- ACID compromise
- Easy/Light Configuration
- Schema Free or Schema Light
- Excellent client language support (Python, Ruby etc.)
- Evolved from matching the use to data store organizations

Key-Value Data Store

- REDIS (Data Structure Server)
- Rich Values
  - Strings
  - Lists
  - Sets
  - Sorted Sets
- Rich and Atomic Operations
- Very Fast (>100K SETS/GETS per second)

REDIS

- SET *key value*
- GET *key value*
- INCR *key*
- INCRBY *key increment*
- LPUSH/LPOP *key value*
- RPUSH/RPOP *key*
- BLPOP/BRPOP *key*
- SADD *key member*
- ZADD *key score member*
- HSET *key field value*
- HGET *key field*

REDIS

- Whole datastore in memory
- Only keys in memory, values can spill to disk (Virtual memory)
- Semi / Full persistent mode
- Transactions (MULTI / EXEC / DISCARD)
- PubSub (PUBLISH / [P]SUBSCRIBE)

REDIS (links)

- *http://code.google.com/p/redis/ (project)*
- *http://code.google.com/p/redis/wiki/CommandReference (commands)*
- *http://simonwillison.net/static/2010/redis-tutorial/ (tutorial)*
- *http://try.redis-db.com/ (online redis shell)*

Document Oriented Data Store

- MongoDB
- Database, Collections, Nested, Reference
- Document Oriented, Binary format BSON (Binary JSON)
- B-Tree index on attributes
- Easy replication, writes on Master, Queries on slaves
- Data Sharding

MongoDB

- Parameters and results are Javascript literals
- Sample queries
    - db.users.find({})
    - db.users.find({‘last_name’ : ‘Smiths’})
    - db.users.find({‘last_name’ :‘Smiths’}, {‘ssn’ : 1})
    - db.users.find({}, {‘thumbnail’ : 0})
- Support for Cursors
- Update
    - x = { ‘name’ : ‘ John Doe’, ‘userid’ : ‘jdoe’ }
    - db.users.save(x)

MongoDB

- Support for Map/Reduce
- Support to store large binary objects
  - BSON object has size limit of 4MB
  - GridFS provides object size of any limit
- Links
  - *http://www.mongodb.org/*
  - *http://try.mongodb.org/*
  - *http://www.mongodb.org/display/DOCS/Tutorial*

Graph Data Store

- Neo4J
- ORM Impedence, Self Join performance degradation
- Graph data model
  - Nodes, relationships and properties
- Massive scalability, Speedy graph traversal API
- OO API
- Use as RDF store

Neo4J

- Links
  - *http://neo4j.org/*
  - *http://www.infoq.com/presentations/emil-eifrem-neo4j*

Temporal Data Store

- RRDTool (Round Robin Database)
- Time series data storage and graphing
- Fixed Database size (in terms of data points), set during creation time
- Data from different data stores, different types
    - Counter, Gauge, Derive, Absolute
- Collected at Primary data points
- Stored as different Archives (different granularity, by hour, by day etc.)
- Powerful graphing capability

RRDTool

- Working example
  - *http://tac-community:1234*
- Links
  - *http://oss.oetiker.ch/rrdtool/*
  - *http://oss.oetiker.ch/rrdtool/tut/rrdtutorial.en.html*

Version Data Store

- Git
- Distributed version control
- Simple Object Model
  - SHA (files are named by their SHA sum of the content)
  - blob (file content)
  - tree (points to other blobs and trees)
  - commit (points to A tree)
  - tag (mark a specific commit)

Git

- Powerful and Fast Operations
  - clone, push, pull, branch, merge, blame etc.
- Links
  - *http://git-scm.com/*
  - *http://book.git-scm.com/*

Constant DB

- Constant DB (cdb)
- Hash table (in a disk format) storing key, value pairs
- Fast Lookup
- No in place updates, database is rebuilt on writes
- 4G limitation
- Links
  - *http://cr.yp.to/cdb.html*
  - *http://cr.yp.to/cdb/cdb.txt*

Observations

- Right tool for the right job
- Use a data store technology meets application needs
- Evolve, move pertinent data objects out of Relational DBs to matching technologies
- Use one of the client language bindings
- Infrastructure challenges
- Maturity of these technologies
- Interfaces are somewhat raw/low-level. Opportunity for layered applications, interfaces, frameworks over this.

Finally

WHERE THE HECK IS MY DATA?
ITS THERE, UP IN THE CLOUDS.
Brainstuck.com

Q&A

- Thanks!