NoSQL Database Benchmarking

Prasoon Kumar

MongoDB
Agenda

- What is NoSQL?
- YCSB
- Benchmarking parameters
- Findings
- Wrap up
What is NoSQL?
What is NoSQL?

• There are many NoSQL systems
  Lists 150 currently, up from 120 in 2012

• How do they compare?
  – Feature trade-off
  – Performance trade-off
  – Not clear!
Scale-out vs Scale-up

Then: Scale Up

Now: Scale Out

![Graph showing price vs scale for then and now.]
Data Model

Key-Value Store
- Riak
- Memcache
- Project Voldemort
- Redis
- BerkeleyDB

Document Database
- MongoDB
- CouchDB
- OrientDB

Column-Family Stores
- Amazon SimpleDB
- Cassandra
- Hbase
- Hypertable

Graph Databases
- FlockDB
- Neo4J
- OrientDB
Yahoo! Cloud Serving Benchmark aka ycsb
Origin

Paper that started it all!
Benchmarking cloud serving systems with YCSB, 2010 (paper, pdf)
Authors: Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan and Russell Sears
Yahoo! Research

Goal
• Evaluate different systems on common workload
• Focus on performance, scaleout
• Extensible by parameters and even writing different workload
Install

- Download YCSB to client node
git clone https://github.com/achille/YCSB

- Install Java driver for particular database

- Run `ycsb` command:
  
  - `./bin/ycsb load mongodb -P workloads/workloada -p recordcount=10000`
  
  - `./bin/ycsb run mongodb -P workloads/workloada -p recordcount=10000`

* Original Brian Cooper’s client available on github. Google for it.
Steps

There are 6 steps to running a workload:

1. Set up the database system to test
2. Choose the appropriate DB interface layer
3. Choose the appropriate workload
4. Choose the appropriate runtime parameters (number of client threads, target throughput, etc.)
5. Load the data
6. Execute the workload
Tool

- Workload parameter file
  - R/W mix
  - Record size
  - Data set
  - ...

- Command-line parameters
  - DB to use
  - Target throughput
  - Number of threads
  - ...

- YCSB client
  - Workload executor
  - Client threads
  - Stats

- DB client

- Extensible: plug in new clients

- NOSQL DB

Extensible: define new workloads

mongoDB
Operations within workload

Operations against a data store were randomly selected and could be of the following types:

1. **Insert**: Inserts a new record.
2. **Update**: Updates a record by replacing the value of one field.
3. **Read**: Reads a record, either one randomly selected field, or all fields.
4. **Scan**: Scans records in order, starting at a randomly selected record key. The number of records to scan is also selected randomly from the range between 1 and 100.
Workloads

1. Workload A: Update heavily
2. Workload B: Read mostly
3. Workload C: Read only
4. Workload D: Read latest
5. Workload E: Scan short ranges
6. Workload F: Read-modify-write
7. Workload G: Write heavily

Each workload is defined by:

1. The number of records manipulated (read or written)
2. The number of columns per each record
3. The total size of a record or the size of each column
4. The number of threads used to load the system
Measure It!

- Availability: what is uptime requirement
- Throughput
  - average read/write/users
  - peak throughput?
  - OPS (operations per second)? per hour? per day
- Responsiveness
  - what is acceptable latency?
  - is higher during peak times acceptable?
Results

Throughput varies a lot depending on durability and consistency parameters, even for a single database.
Disadvantages

- *Secondary indexes* are not provided by all, so YCSB doesn’t test the same.
- It doesn’t test rich functionality exposed by specialized databases.
- It provides limited support for scaling across *multiple clients*.
- Poor documentation – Read the source code.
Wrapping Up
Summary

• YCSB is a versatile tool for testing many different workloads.
• It allows apple to apple comparison for NoSQL and cloud databases.
• It doesn’t test for correctness.
• Database can be configured for durability, consistency and other parameters. You should model your usecases and benchmark.
• Polyglot persistence is here to stay! Use different data-stores in a single application for work.
Questions?
Thank You

Prasoon Kumar
MongoDB
MMS: DB Stats

avg obj size

collections

data size

file size

index size

indexes

lock %

num extents

objects
Finding the bottleneck

Source: http://www.flickr.com/photos/laenulfhean/462715479/
What is performance tuning?

1. Assess the problem and establish acceptable behavior
2. Measure the current performance
3. Find the bottleneck*
4. Remove the bottleneck
5. Re-test to confirm
6. Lather, rinse, repeat

* - (This is often the hard part)

(Adapted from http://en.wikipedia.org/wiki/Performance_tuning)
Pro-Tip: know thyself

You have to recognize normal to know when it isn’t.

Source: http://www.flickr.com/photos/skippy/6853920/
MMS: Metrics to Watch

- Memory usage
- Opcounters
- Lock %
- Queues
- Background flush average
- Replication stats