Membase + CouchOne = COUCHBASE

Is it us, or did databases just get a lot more awesome?
What is Membase?
Before: Application scales linearly, data hits wall

- Application Scales Out
  Just add more commodity web servers

- Database Scales Up
  Get a bigger, more complex server
Membase is a distributed database

In the data center

On the administrator console
Membase is Simple, Fast, Elastic

- Five minutes or less to a working cluster
  - Downloads for Linux and Windows
  - Start with a single node
  - One button press joins nodes to a cluster

- Easy to develop against
  - Just SET and GET – no schema required
  - Drop it in. 10,000+ existing applications already “speak membase” (via memcached)
  - Practically every language and application framework is supported, out of the box

- Easy to manage
  - One-click failover and cluster rebalancing
  - Graphical and programmatic interfaces
  - Configurable alerting
Membase is Simple, Fast, Elastic

- **Predictable**
  - “Never keep an application waiting”
  - Quasi-deterministic latency and throughput

- **Low latency**
  - Built-in Memcached technology

- **High throughput**
  - Multi-threaded
  - Low lock contention
  - Asynchronous wherever possible
  - Automatic write de-duplication
Membase is Simple, Fast, Elastic

- **Zero-downtime elasticity**
  - Spread I/O and data across commodity servers (or VMs)
  - Consistent performance with linear cost
  - Dynamic rebalancing of a live cluster

- **All nodes are created equal**
  - No special case nodes
  - Any node can replace any other node, online
  - Clone to grow

- **Extensible**
  - Filtered TAP interface provides hook points for external systems (e.g. full-text search, backup, warehouse)
  - Data bucket – engine API for specialized container types
Fact: Membase development team has also contributed over half of the code to the Memcached project.
Use Cases
Ad targeting

40 milliseconds to come up with an answer.
Search and Gaming Portal

Database

NAVER

Hangame

NAVER 뉴스

membase

Flags of South Korea, Japan, and China.
Membase Architecture
Clustering

• Underlying cluster functionality based on erlang OTP
• Have a custom, vector clock based way of storing and propagating...
  – Cluster topology
  – vBucket mapping
• Collect statistics from many nodes of the cluster
  – Identify hot keys, resource utilization
A Membase Node

memcached operations
(any client)  
memcached smart client operations  
REST Interface

Membase Node
A Membase Node: Component View

moxi

ns_server

membase
(memcached + membase engine)

TAP

vbucketmigrator
Clients, nodes and other nodes

- ns_server
- membase (memcached + membase engine)
- moxi
- vbucketmigrator

- memcached operations with tap commands
- port 11211 memcached operations
- port 11210 memcached operations
- REST/comet cluster topology and vbucket map

Client
moxi + Client

Client port 11211
moxi + Client port 11210

memcached operations

moxi
ns_server

membase (memcached + membase engine)

TAP
Data buckets are secure Membase “slices”

In the data center

On the administrator console
Dataset may have many items infrequently accessed. However, memcached has different behavior (LRU) than wanted with membase.

Still, traditional (most) RDBMS implementations are not 100% correct for us either. The speed of a miss is very, very important.
Membase Demo
The Future
But first, an intro...
What is Couchbase?

1+1 really does equal 3
Where will the merger take us

- Sub-document manipulation
- New approaches to indexing
- Simplified development
Attributions

Membase + CouchOne = COUCHBASE

Is it us, or did databases just get a lot more awesome?