Membase Announcements
Hadoop World 2010
Membase – Company Background

- Database software company
- Founded: January 2009
- Proven Leadership
  - Management team combines 50+ years experience from Informix, VMware, IBM, Intel, Netscape, Hotmail, Yahoo!
  - Founders have previously founded companies worth over $250 million at exit (Kiva, Akimbi, Fifth Generation Systems)
- Customer and Partner Advisory Board
  - Cadir Lee (Zynga), Bobby Johnson (Facebook), Mike Olson (Cloudera), Alan Kasindorf (Six Apart)
- Strong balance sheet with top-tier investors
  - $15 million funding (Series A and B)
  - Accel Partners, Mayfield Fund, North Bridge Venture Partners, Zynga
Hadoop World 2010 Announcements

🌟 Membase is generally available
- Membase Server - Enterprise Edition
- Membase Server - Community Edition
- Membase source code - Apache 2.0 license

🌟 NorthScale, Inc. is now Membase, Inc.

🌟 Membase and Cloudera announce partnership, integration and joint customers
- Membase Server: low-latency access to data
- Cloudera Distribution for Hadoop: batch data analytics
- Bidirectional data integration via Flume and Sqoop
- AOL Advertising and ShareThis joint customers
Membase Generally Available
Where to get Membase

membase.org

- Source code.
- Community binary.
- Developer community.

membase.com

- Certified product.
- Membase Network.
- Services and support.
## Membase Enterprise Edition - Pricing

<table>
<thead>
<tr>
<th>Pricing and Licensing</th>
<th>Membase Basic</th>
<th>Membase Standard</th>
<th>Membase Premium</th>
<th>Memcached Basic</th>
<th>Memcached Standard</th>
<th>Memcached Premium</th>
<th>Membase Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>$999</td>
<td>$1,999</td>
<td>$2,999</td>
<td>$599</td>
<td>$999</td>
<td>$1,499</td>
<td>$0</td>
</tr>
<tr>
<td>Licensing</td>
<td>1 Node</td>
<td>1 Node</td>
<td>1 Node</td>
<td>1 Node</td>
<td>1 Node</td>
<td>1 Node</td>
<td>1 Node, Max 2 Node Cluster</td>
</tr>
<tr>
<td>Licensing Restrictions</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>Max 2 Node Cluster for Production; Unlimited for Eval</td>
</tr>
</tbody>
</table>

### Support Entitlements

<table>
<thead>
<tr>
<th>Hours of Operation</th>
<th>Support Channel</th>
<th>Number of Cases</th>
<th>Response Time - P1</th>
<th>Response Time - P2</th>
<th>Product Update Entitlements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 x 5</td>
<td>Forums, Email, Web</td>
<td>5</td>
<td>1 Business Day</td>
<td>1 Business Day</td>
<td>Major Releases (1.0)</td>
</tr>
<tr>
<td></td>
<td>7am - 5pm PT</td>
<td></td>
<td>5 hour (within business hours)</td>
<td>5 hour (within business hours)</td>
<td>X</td>
</tr>
<tr>
<td>10 x 5</td>
<td>Forums, Email, Web, Phone</td>
<td>Unlimited</td>
<td>2 hours</td>
<td>2 hours</td>
<td>Minor Releases (1.x)</td>
</tr>
<tr>
<td></td>
<td>7am - 5pm PT</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>24 x 7</td>
<td>Forums, Email, Web, Phone</td>
<td>Unlimited</td>
<td>1 Business Day</td>
<td>1 Business Day</td>
<td>Maintenance Releases (1.0.x)</td>
</tr>
<tr>
<td></td>
<td>n/a</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>10 x 5</td>
<td>Forums, Email, Web, Phone</td>
<td>Unlimited</td>
<td>1 Business Day</td>
<td>1 Business Day</td>
<td>Hotfixes</td>
</tr>
<tr>
<td></td>
<td>7am - 5pm PT</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>24 x 7</td>
<td>Forums, Email, Web, Phone</td>
<td>Unlimited</td>
<td>1 Business Day</td>
<td>1 Business Day</td>
<td>Membase Network</td>
</tr>
<tr>
<td></td>
<td>n/a</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Update Notification Service</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Technical Alerts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
“AOL serves more than 5 billion impressions per day from our ad serving platforms, and any incremental improvement in processing time translates to huge benefits in our ability to more effectively serve the ads to needed meet our contractual commitments. Traditional databases like MySQL lack the scalability required to support our goal of five milliseconds per read/write. Creating user profiles with Hadoop, then serving them from Membase, reduces profile read and write access to under a millisecond, leaving the bulk of the processing time budget for improved targeting and customization.”

Pero Subasic
Chief Architect, AOL
Membase-Cloudera Partnership

- Joint development of bi-directional software integration between Membase and Hadoop
  - Membase NodeCode Module streaming interface to Cloudera Distribution for Hadoop via Flume interface
  - Sqoop-derived command line utility for bi-directional batch movement of data between Membase and Cloudera Distribution for Hadoop

- Joint marketing and sales of integrated distributed OLTP-OLAP solution
  - Membase – the distributed OLTP solution
  - Cloudera – the distributed OLAP solution

- Cloudera to distribute integration components
Customer use case – Ad targeting

40 milliseconds to come up with an answer.

1. Events
2. Profiles, campaigns
3. Profiles, real time campaign statistics
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

Graphs show system cost and application response time as a function of users, with a point indicating that the database won't scale beyond this point.
Membase is a distributed database

In the data center

On the administrator console
Fact: Membase development team has also contributed over half of the code to the Memcached project.
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
Membase is Simple, Fast, Elastic

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

- Low latency
  - Built-in Memcached technology
  - Auto-migration of hot data to lowest latency storage technology (RAM, SSD, Disk)
  - Selectable write behavior – asynchronous, synchronous (on replication, persistence)

- 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
  - 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
  - Membase NodeCode [FUTURE]
Leading cloud service (PAAS) provider
Over 65,000 hosted applications
Over 2,000 users to date
**Membase Server** serving over 3,000 Heroku customers

- Social game leader – FarmVille, Mafia Wars, Café World
- Over 230 million monthly users
- **Membase Server** is the 500,000 ops-per-second database behind FarmVille and Café World
After: Data layer scales like application logic layer
Data layer now scales with linear cost and constant performance.

Scaling out flattens the cost and performance curves.
A practical path to NoSQL database adoption
Technical Details - Backup
vBucket mapping

Key $\rightarrow$ vBucket (hash function)

vBucket $\rightarrow$ Servers (table lookup)

All possible membase keys

vBucket 1

vBucket 2

vBucket 3

vBucket $n$

vBucket $m$

vBucket 1 $\rightarrow$ Server$_1$ / Server$_2$, Server$_3$

vBucket 2 $\rightarrow$ Server$_2$ / Server$_3$, Server$_4$

vBucket $n$ $\rightarrow$ Server$_p$ / Server$_q$, Server$_r$

vBucket-$Server$ Map - Example

vBucket$_1$ $\rightarrow$ Server$_A$, Server$_B$, Server$_C$

vBucket$_2$ $\rightarrow$ Server$_A$, Server$_B$, Server$_C$

vBucket$_3$ $\rightarrow$ Server$_B$, Server$_A$, Server$_C$

vBucket$_4$ $\rightarrow$ Server$_B$, Server$_A$, Server$_C$

vBucket$_5$ $\rightarrow$ Server$_C$, Server$_A$, Server$_B$

vBucket$_6$ $\rightarrow$ Server$_C$, Server$_A$, Server$_B$

Host Server/Replica Servers
User action results in the need to change the VALUE of KEY

1. Application updates key’s VALUE, performs SET operation

2. Membase (memcached) client hashes KEY, identifies KEY’s master server

3. SET request sent over network to master server

4. Membase replicates KEY-VALUE pair, caches it in memory and stores it to disk

5.
Membase data flow – under the hood

SET request arrives at KEY’s master server

1. Listener-Sender
2. RAM*
3. SSD
4. Disk
5. SET acknowledgement returned to application

Master server for KEY

Replica Server 1 for KEY

Replica Server 2 for KEY
Membase Architecture

Data Manager

- memcached protocol listener/sender
- membase storage engine

Cluster Manager

- REST management API/Web UI
- Heartbeat
- Process monitor
- Configuration manager
- Global singleton supervisor
- Erlang/OTP

Port Numbers:
- 8080
- 4369
- 21100 – 21199
Membase Architecture

- **11211** memcapable 1.0
- **11210** memcapable 2.0

- moxi
- memcached protocol listener/sender
- membase storage engine

- REST management API/Web UI
- Heartbeat
- Process monitor
- Configuration manager
- Global singleton supervisor
- Rebalance orchestrator
- Node health monitor
- vBucket state and replication manager

- Erlang/OTP
- HTTP
- 8080
- erlang port mapper
- 4369
- distributed erlang
- 21100 – 21199
Data buckets are secure membase “slices”