

EUC 2015
MongooseIM
The Right Tool for
Scalable Messaging

Michał Piotrowski
michal.piotrowski@erlang-solutions.com

Erlang

MongooseIM - The Right Tool for Scalable Messaging

Agenda

- What is MongooseIM
- Motivation behind 1.5.0
- Story about upcoming 1.6.0



Erlang

MongooseIM - The Right Tool for Scalable Messaging

What is MongooseIM?

- XMPP server
- Based on ejabberd but heavily modified
- Instant messaging for Social Media, Gaming and Telecommunications
- Solution designed for high volume enterprise and business purposes
- Easily scalable distributed system
- Highly customisable platform thanks to industry standards and Open Source technology



Erlang

MongooseIM - The Right Tool for Scalable Messaging

Target domains

Social Media

(messaging is the key part of social web sites)

IoT

Gaming

(multi-user chats are integral part of the gameplay)

Telecoms

(chat features are part of standard communication software)

MongooseIM - The Right Tool for Scalable Messaging

Features

- Horizontal linear cluster scalability
- Configurable database backends
- Support for WebSockets
- Multi-User Chat
- Pluggable modules
 - Authentication
 - XMPP extensions
- Custom extensions

MongooseIM - The Right Tool for Scalable Messaging

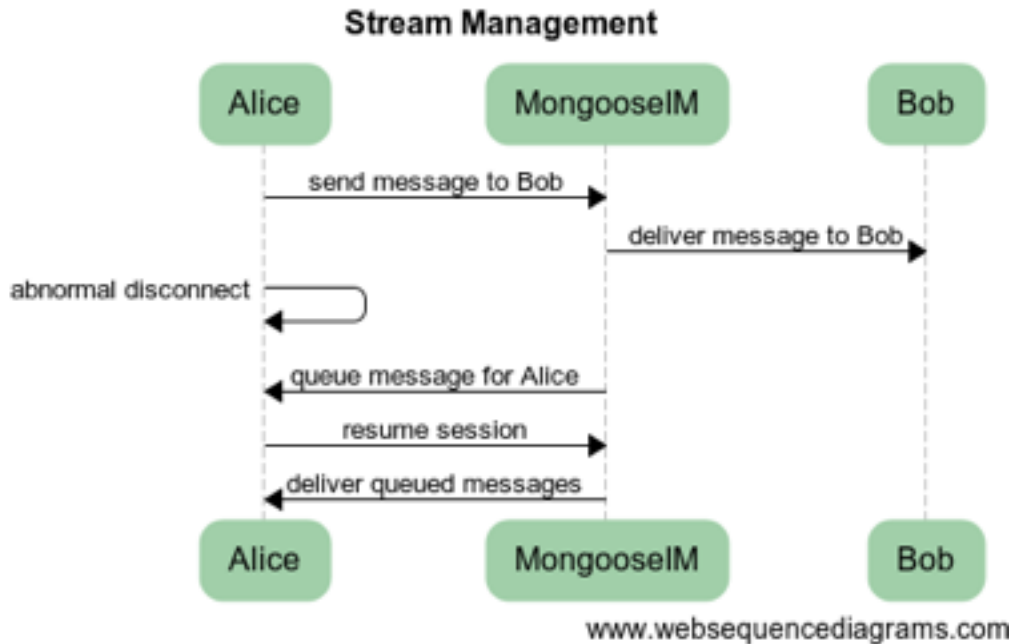
1.5.x - modern mobile messaging challenges

- Motivation behind 1.5.0
 - Meeting the challenges of modern mobile messaging with XMPP
 - <https://github.com/esl/MongooseIM/wiki/Meeting-the-challenges-of-modern-mobile-messaging-with-XMPP>
- MongooseIM 1.5.0 released on Dec 2, 2014
 - <https://github.com/esl/MongooseIM/releases/tag/1.5.0>
- Version 1.5.1 released on Apr 2, 2015
 - <https://github.com/esl/MongooseIM/releases/tag/1.5.1>

MongooseIM - The Right Tool for Scalable Messaging

1.5.x - modern mobile messaging challenges

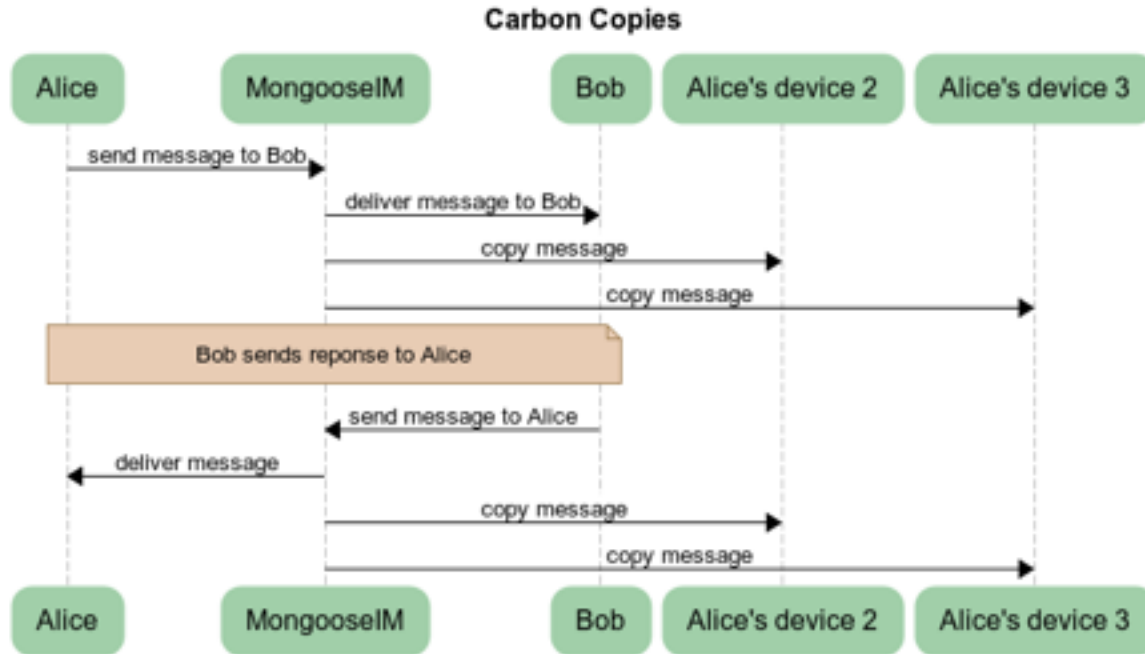
- Actively managing a live stream
 - XEP-0198



MongooseIM - The Right Tool for Scalable Messaging

1.5.x - modern mobile messaging challenges

- Complete engagement across devices
 - XEP-0280



MongooseIM - The Right Tool for Scalable Messaging

1.5.x - modern mobile messaging challenges

- Message history and synchronisation
 - XEP-0313 - Message Archive Management
 - available backends
 - MySQL, PostgreSQL and other ODBC DBs
 - Apache Cassandra
 - planned backends
 - Riak 2.0
 - DynamoDB

MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - coming soon

- Motivation behind 1.6.0
 - DevOps friendliness
 - Integration with Riak 2.0

MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - what's already done

- DevOps friendliness
 - changed metrics backend from folsom to exometer
 - more metrics
 - stanza size histograms (plain XML, compressed, encrypted)
 - Erlang distributed network traffic
 - ODBC workers network traffic
 - Module's backend operation time statistics
 - MAM write / lookup
 - Roster get

MongooseIM - The Right Tool for Scalable Messaging

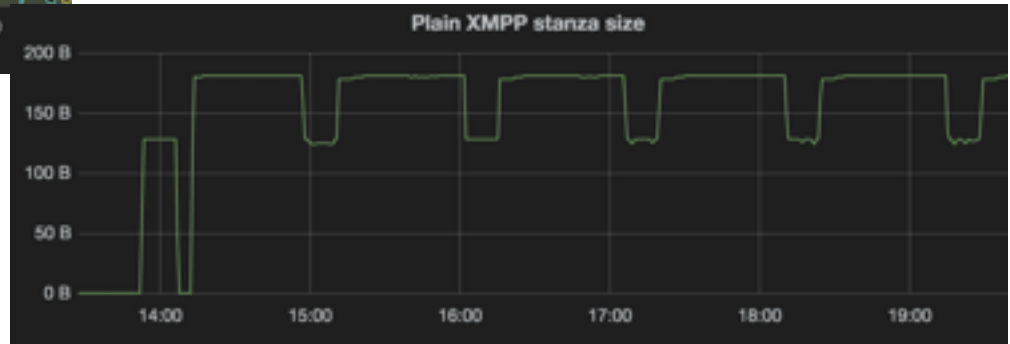
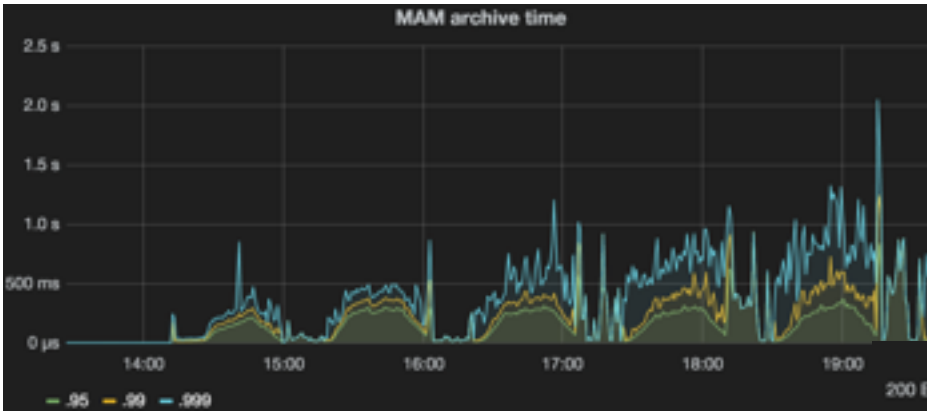
1.6.0 - DevOps friendliness

- Why exometer?
 - better performance
 - more flexible - allows to create metrics calling a function when needed

```
ExEvalSingleValue = {[{l, [{t, [value, {v, 'Value'}]}]}], [value]}.  
exometer:new([global, totalSessionCount],  
              {function, ejabberd_sm, get_total_sessions_number, [],  
               eval, ExEvalSingleValue}).
```

MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - metrics sample



MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - what's already done

- Riak 2.0 integration
 - user base
 - message archive
 - private storage

MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 user base

- XMPP JID encoded in bucket and key
 - host as part of bucket
 - username is the key
- other user data put into Riak 2.0 map
- allows search with Yokozuna
- every hosted domain has it's own bucket

MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - first approach

- weekly buckets
 - easier to remove old archives
- archiving messages is easy
- every message is put to new objects (one for sender one for receiver)
- object's key contains local and remote jids and also timestamp
- simple `term_to_binary` as object's value
- reading archives is complicated
 - need to fold over all weeks in worst case
 - map-reduce key filtering used to search for relevant messages
- poor performance after less than 100k archived messages

MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - load tests

- single user scenario
 - connect user
 - read last 10 msgs from archive
 - write messages to other users
- load testing tool
 - amoc: <https://github.com/esl/amoc>
- single MongooseIM node
- Riak cluster of 3 nodes

MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - load tests - first phase

- load test goal
 - connect 10k users
 - send 500 msgs/s - archive 1k / s
 - archive sync less than 30s
- achieved results:
 - around 300 successfully connected users
 - avg archive lookup time 60s!
- need to optimise something

MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - second approach

- buckets per user
 - less keys to filter
 - still easy archiving
 - simpler archive reading
 - hard old archive cleaning
- the same key and object value

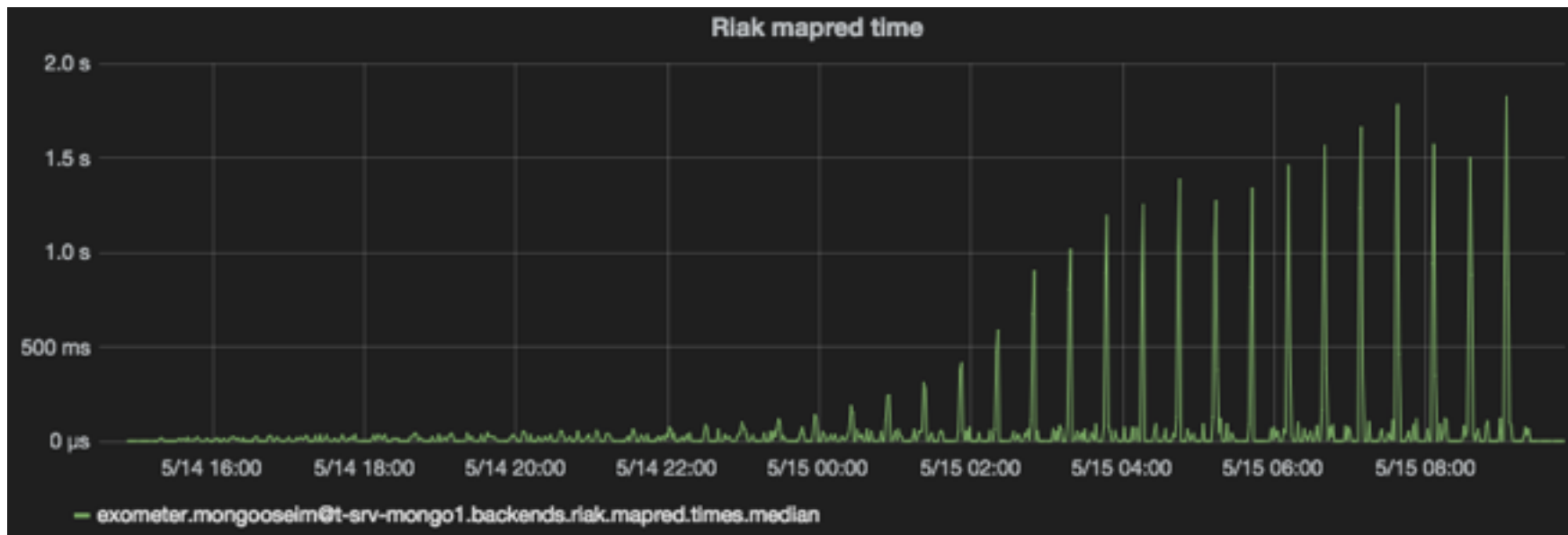
MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - second approach

- achieved results:
 - around 1k successfully connected users
 - avg archive lookup time less than 1s
 - but only for first couple of hours (when the archive is small)
- map-reduce key filtering cannot be used to read the archive

MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - second approach



MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - third approach

- back to weekly buckets
- use Riak 2.0 search to read archive
- key is the same
- object is Riak 2.0 map
- searching
 - by `_yz_rk` - filtering by archive owner and remote user
 - by `msg_id` (from map) - timestamp and id filtering

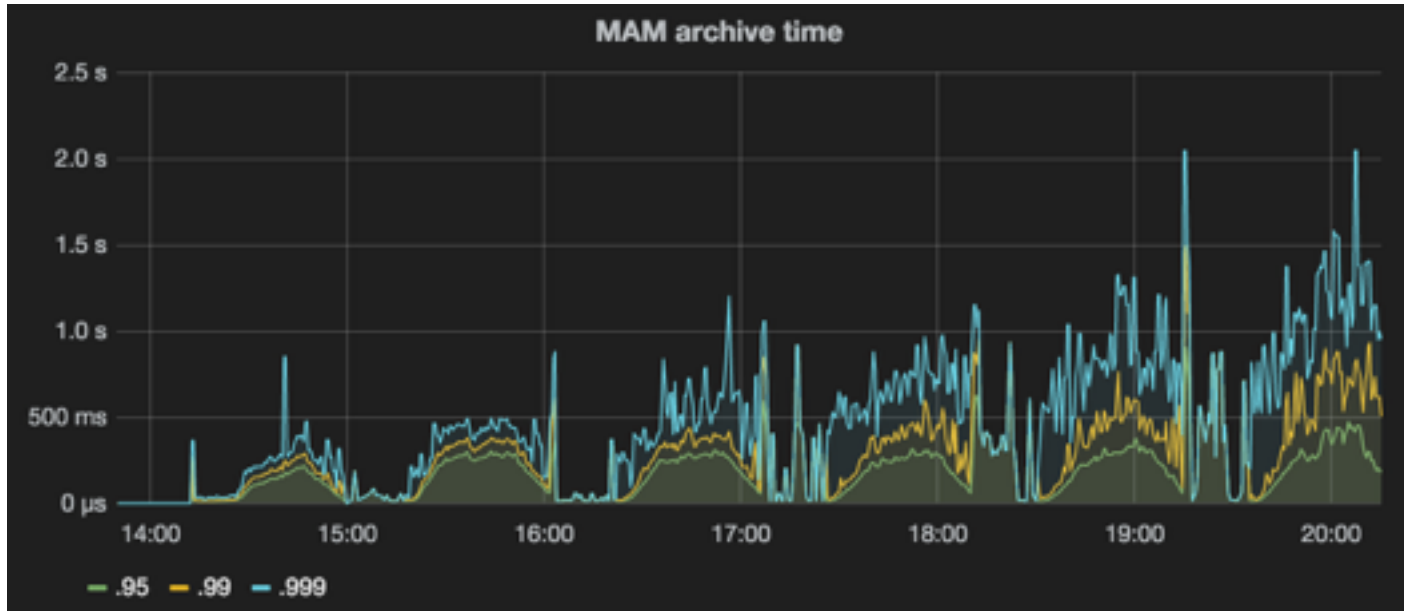
MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - third approach - load tests

- achieved results
 - 10k connected users
 - 500 sent msgs / s (1k Riak writes / s)
- avg archive time ~150ms
- .999 percentile archive time 2s
- lookup time < 3s

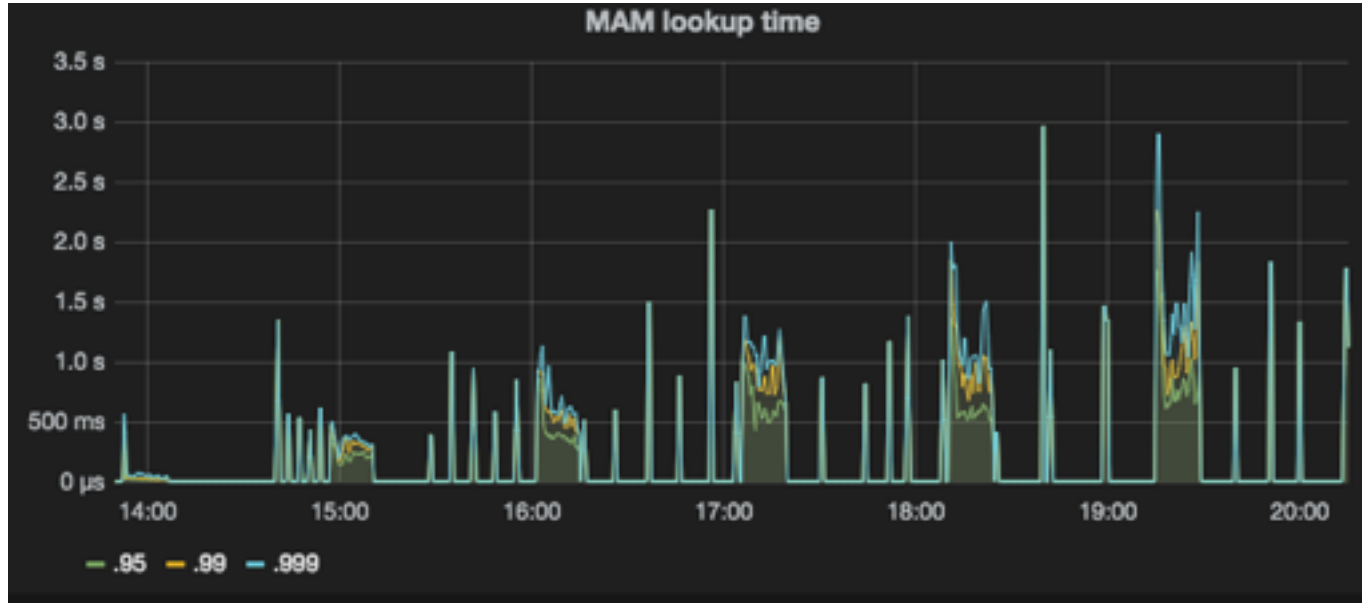
MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - third approach - load tests



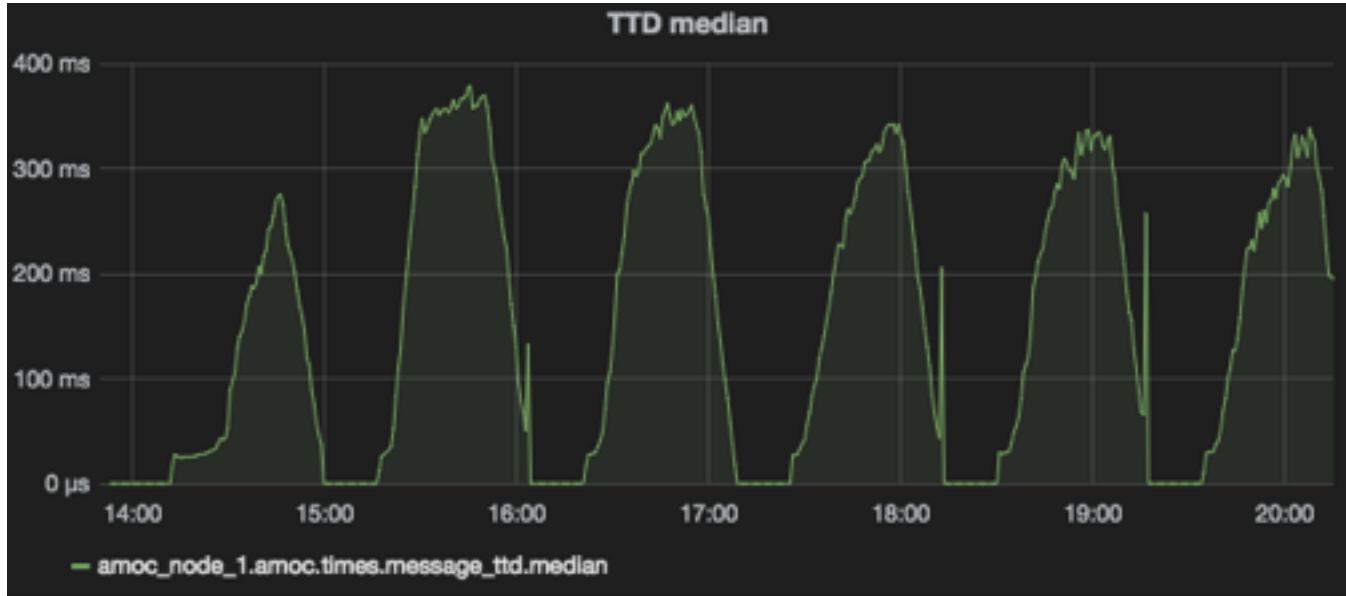
MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - third approach - load tests



MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - third approach - load tests



MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - Riak 2.0 MAM - third approach - summary

- objects indexing puts significant overhead
 - without indexing avg archive time 5 - 6ms (1k Riak writes / s)
 - with indexing avg archive time ~150ms!
- use Riak 2.1.1 where the write was optimised
- do not put `term_to_binary` into object which is indexed
 - the format breaks indexing (even if Apache Solr is set not to index such field)
- further research to optimise the archive

MongooseIM - The Right Tool for Scalable Messaging

1.6.0 - stay tuned

Thank you for your attention!

Questions?

