Erlang Solutions Ltd.

Massive IM Scalability using WebSockets

Michał Ślaski

What am I chatting about?

What am I chatting about?

- Chat features

What am I chatting about?

- Chat features
- Chat protocols

What am I chatting about?

- Chat features
- Chat protocols
- Chat servers

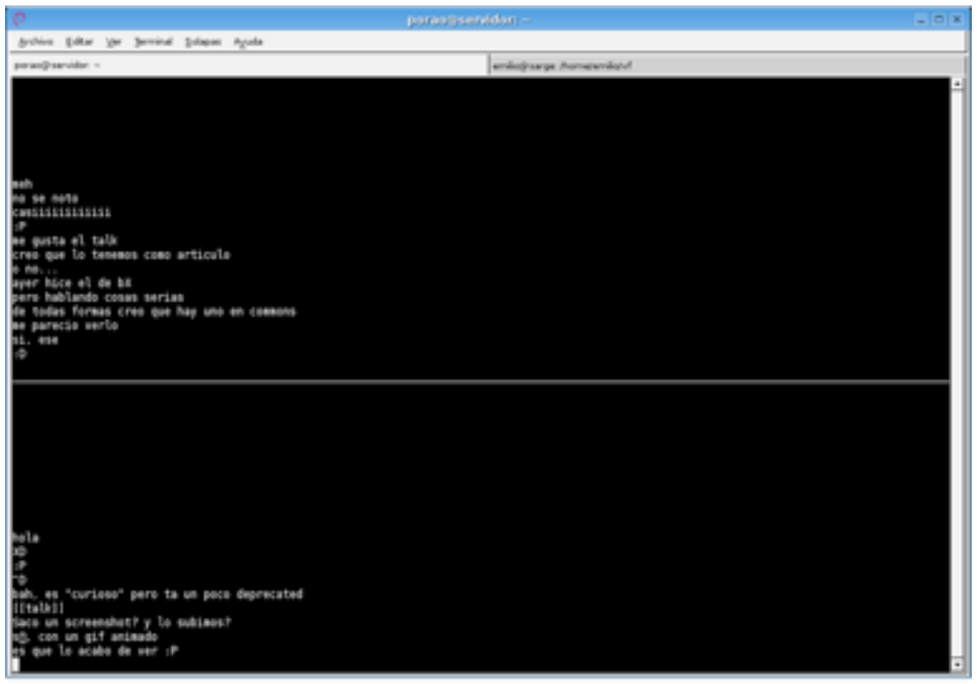
What am I chatting about?

- Chat features
- Chat protocols
- Chat servers
- Some tests

Chat feature

- text-based communication between two or more participants

- First messengers like
  - talk
  - IRC
  - ICQ
  - AOL Instant Messenger



Chat feature

- Modern messengers integrate text, voice and video
  - ooVoo
  - Tango.me
  - Voxer
  - WhatsApp

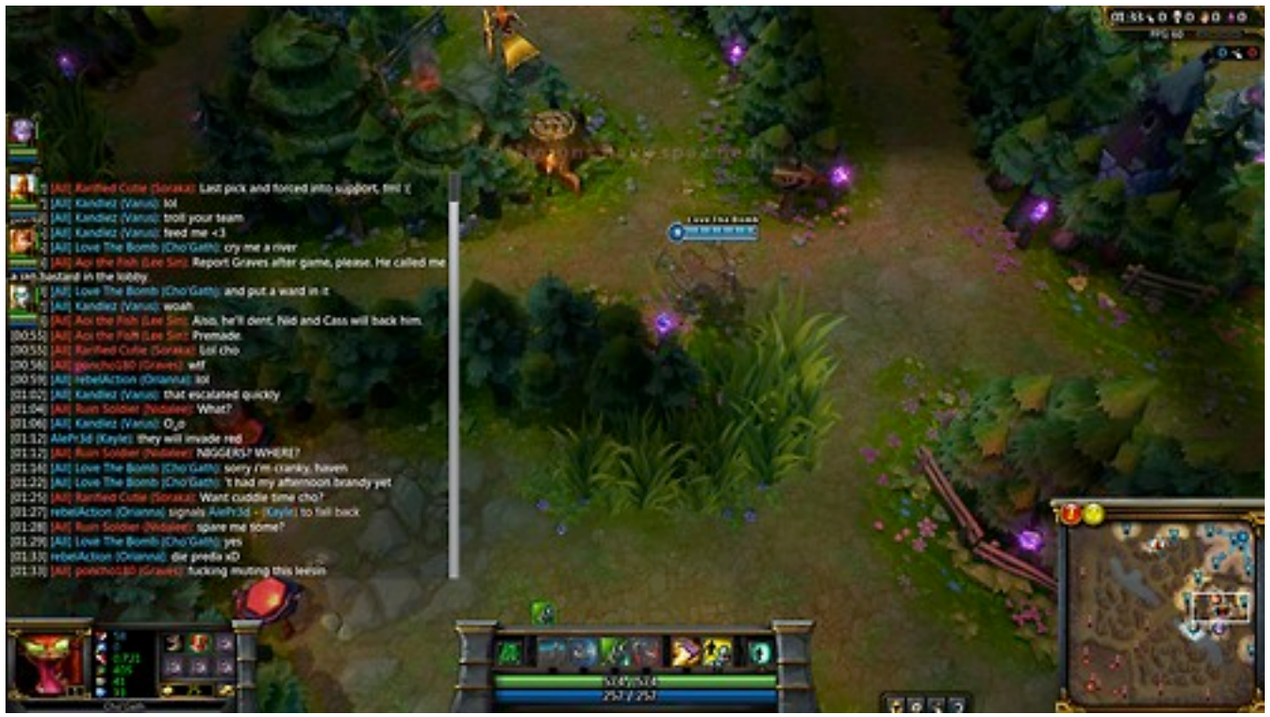
Chat feature

- The ability to chat with your fellows brings interesting “social” feature

- so social networking providers integrate a chat feature too
  - Facebook chat
  - Google Talk
  - NKTalk
  - Tok.tv
  - BigLive

Chat feature

- games offer in-game chat feature
  - battle.net
  - League of Legends



Chat feature

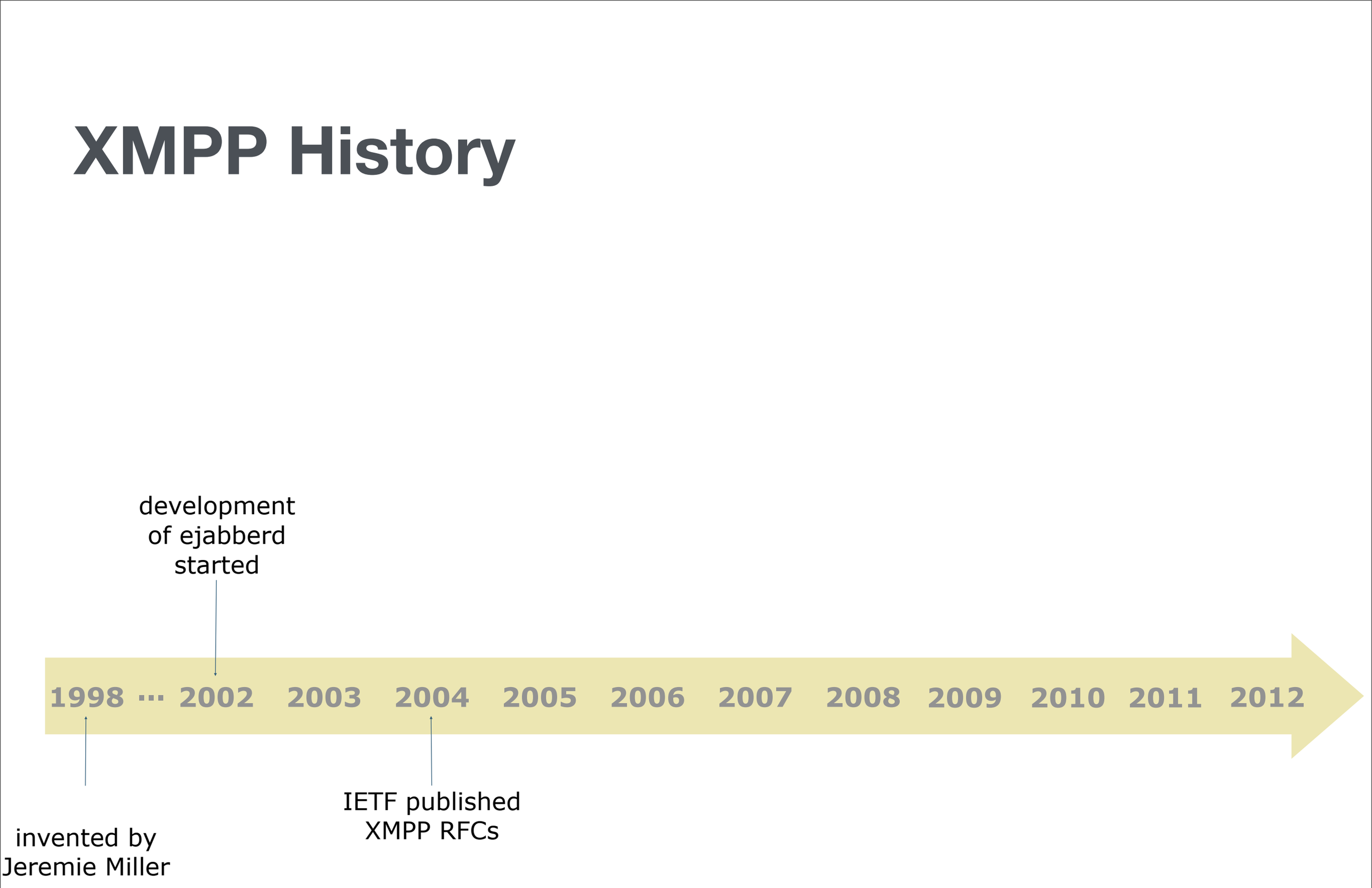
- Chat protocol should cater to
  - web
  - mobile devices
  - on-line games
  - m2m

XMPP - instant messaging protocol

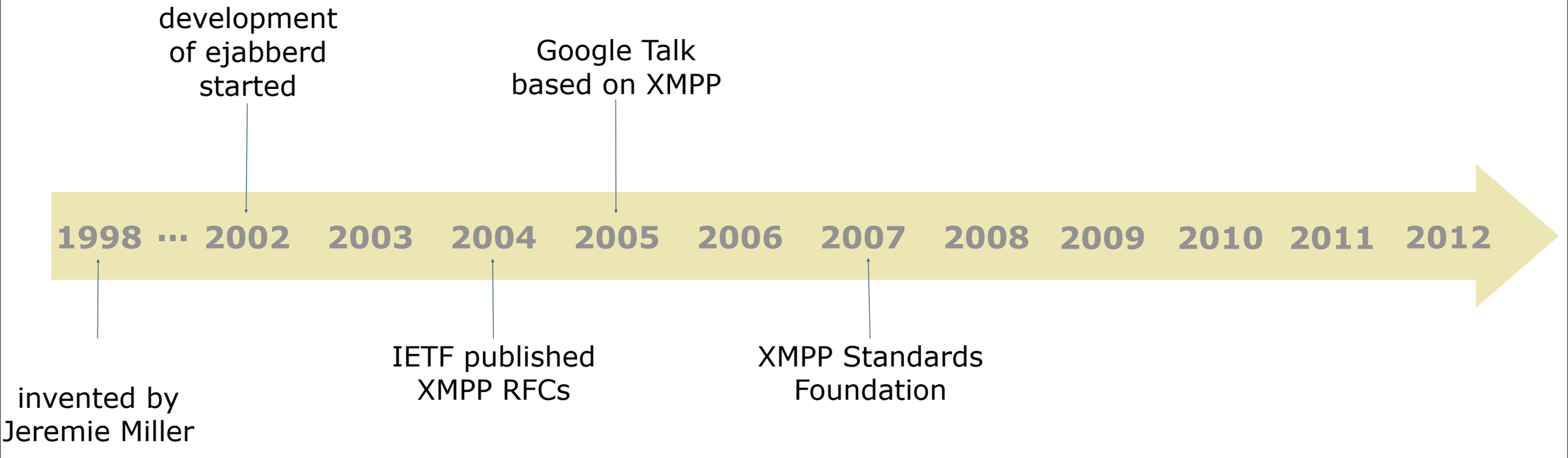
- open standards-based protocol
- XMPP stands for
  e**X**tensible **M**essaging & **P**resence **P**rotocol
- based on XML
- Decentralised client-server architecture

XMPP History

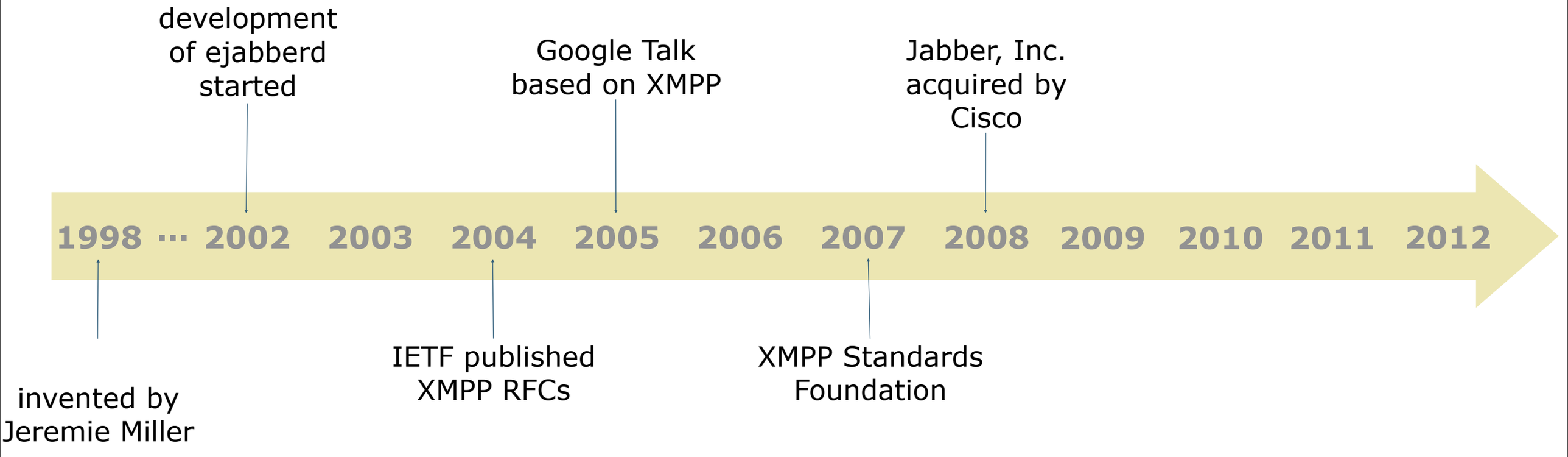
XMPP History



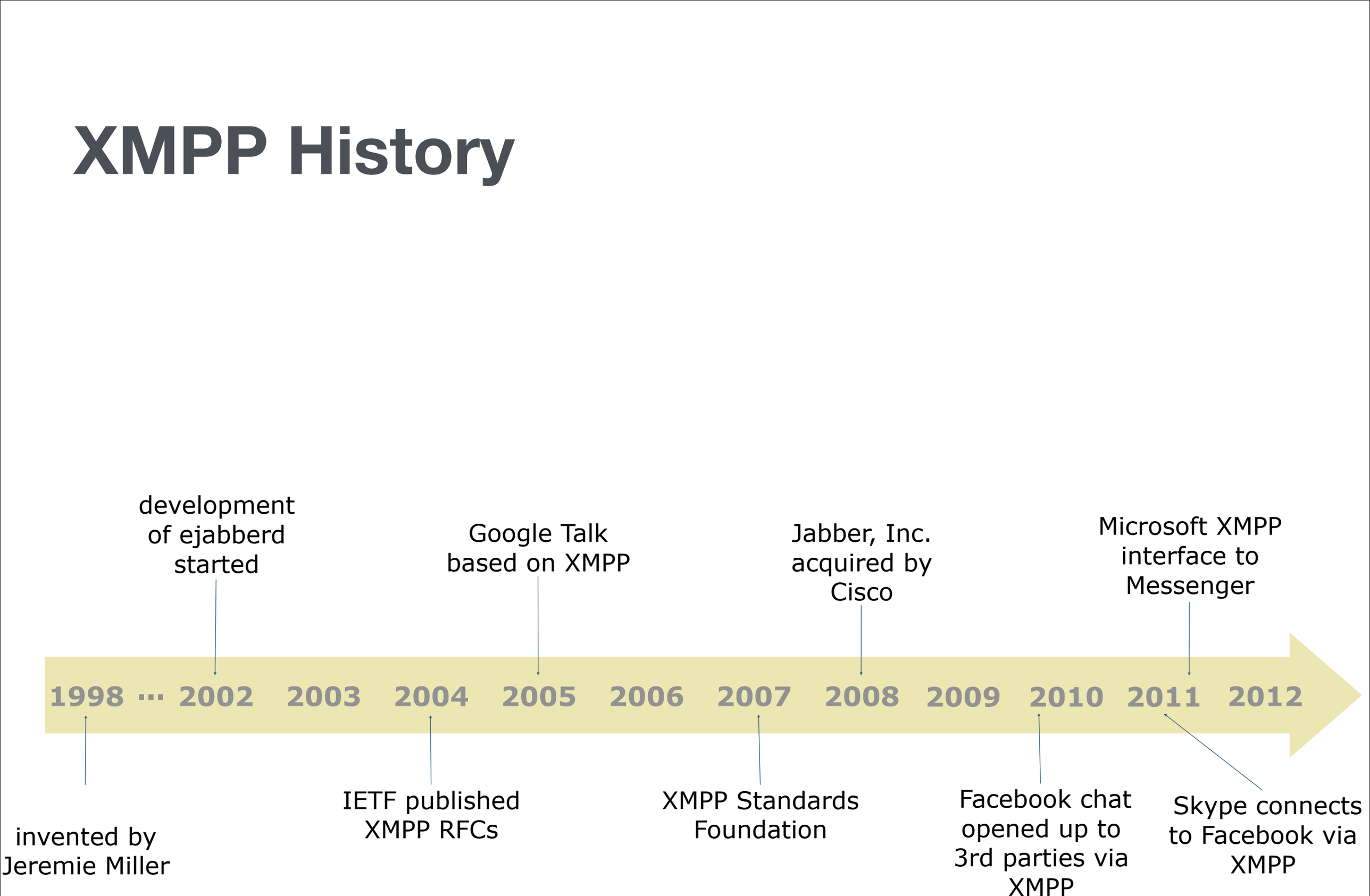
XMPP History



XMPP History



XMPP History



XMPP Properties

Open standard

Secure

Flexible

Decentralised

Efficient

Proven

XMPP Properties

Open standard

Standard formalised by the IETF through RFCs and extensions are published by the XSF

Secure

Flexible

Decentralised

Efficient

Proven

XMPP Properties

Open standard

Secure

Channel encryption, strong authentication, may be isolated from the public network

Flexible

Decentralised

Efficient

Proven

XMPP Properties

Open standard

Secure

Flexible

Custom functionality can be built on top of XMPP

Decentralised

Efficient

Proven

XMPP Properties

Open standard

Secure

Flexible

Decentralised

Anyone can run XMPP server
and there is no central master server

Efficient

Proven

XMPP Properties

Open standard

Secure

Flexible

Decentralised

Efficient — Solves issues of polling approaches

Proven

XMPP Properties

Open standard

Secure

Flexible

Decentralised

Efficient

Proven

In use since 1998, many implementations and deployments, millions of end users

XMPP Properties

Open standard

Secure

Flexible

Decentralised

Efficient

Proven

Client

- Client (user) identified by JID similar to email address
- Client connects to server using TCP/IP
- Each connection represented by resource id

user@jabber.org/home

Client

- Client (user) identified by JID similar to email address
- Client connects to server using TCP/IP
- Each connection represented by resource id

user@jabber.org/home

Protocol: Message

- Basic method of sending information
- Used for
  - instant messaging
  - group chats
  - notifications

```
<message from="user@jabber.org/home"
         to="friend@gmail.com"
         type="chat">
  <body>Let me know what you think!</body>
  <subject>Brainstorm</subject>
</message>
```

Protocol: Message

- Basic method of sending information
- Used for
  - instant messaging
  - group chats
  - notifications

from
stamped by server

```
<message from="user@jabber.org/home"
         to="friend@gmail.com"
         type="chat">
  <body>Let me know what you think!</body>
  <subject>Brainstorm</subject>
</message>
```

Protocol: Presence

- Advertises the availability
- Requires presence subscription
- online
  - away
  - dnd
  - xa
  - chat
- unavailable (offline)

```
<presence/>


<presence>
  <show>away</show>
  <status>eating</status>
</presence>
```

Protocol: Info/Query

- Structure for request-response exchange
- Result tracked using the id attribute
- Payload qualified by the namespace

Client

```
<iq type="get" id="aad6a">
  <query xmlns="jabber:iq:roster"/>
</iq>
```

Server

```
<iq type="result" id="aad6a"
    from="user@jabber.org/Psi" to="user@jabber.org/Psi">
  <query xmlns="jabber:iq:roster">
    <item subscription="both" jid="friend@gmail.com"/>
  </query>
</iq>
```

XMPP

- Non-IM use cases
  - remote system control
  - signalling
  - VoIP
  - geolocation

ejabberd

Jabber/XMPP instant messaging server

- cross-platform
- fault-tolerant
- can be distributed on a cluster

Implements many XEPs

- supports MySQL, PostgresSQL, ODBC, LDAP
- SASL authentication, STARTTLS, SSL

ejabberd

Scalability issues

RAM - all sessions replicated on all nodes in cluster

Web long-polling may be not efficient enough

MongooseIM

fork of ejabberd 2.1.x

- follows Erlang/OTP principles

focused on performance and scalability

- alternative session backend for Redis
- lower memory footprint

Implements

popular XEPs like Service Discovery, MUC, Privacy Lists

XMPP over WebSocket

MongooseIM

Has open sourced regression tests

github.com/esl/ejabberd_tests

Good baseline for building custom XMPP services

Load tests - environment

Load tests - environment

Server node

AMD FX-8150 8x3.3GHz

32GB RAM

Load tests - environment

Server node

AMD FX-8150 8x3.3GHz

32GB RAM

Test environment

Erlang/OTP R15B02

ejabberd 2.1.11

MongooseIM 1.1

MySQL 5.5.24

Ubuntu 12.04 LTS 3.2.0-23-generic

Tsung 1.5.0a

Load tests - environment

Load tests - environment

ejabberd & MongooseIM configuration

Sessions: in Mnesia

User credentials: in MySQL

Rosters: in MySQL

Load tests - environment

ejabberd & MongooseIM configuration

Sessions: in Mnesia

User credentials: in MySQL

Rosters: in MySQL

Erlang VM tweaks:

+K true

+A 5

+P 10000000

-env ERL_MAX_PORTS 1000000

-env ERL_FULLSWEEP_AFTER 2

-env ERL_MAX_ETS_TABLES 100000

Load tests - “max users”

User arrival rate: 150/s

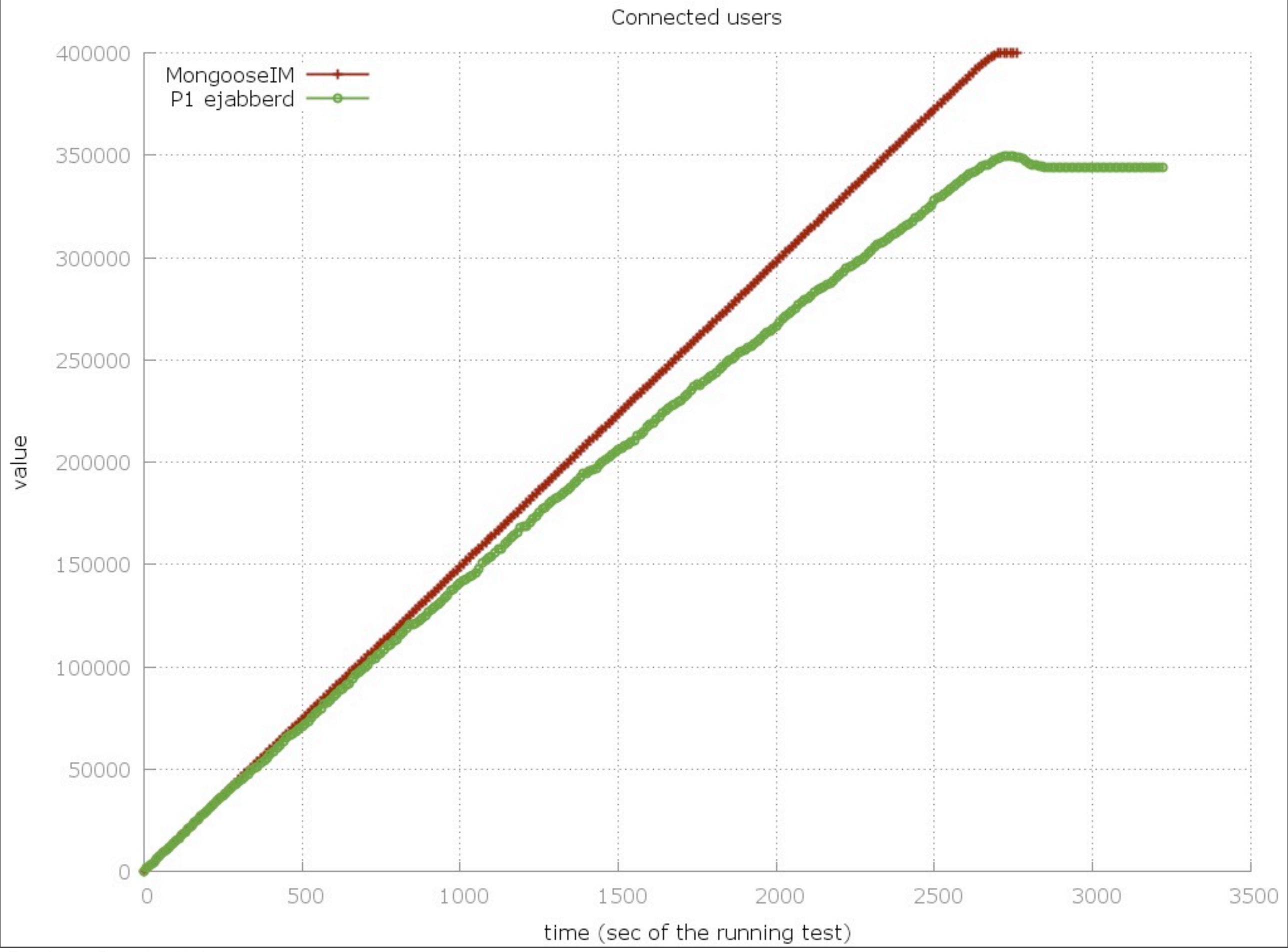
User count: 400k

Roster size: 100

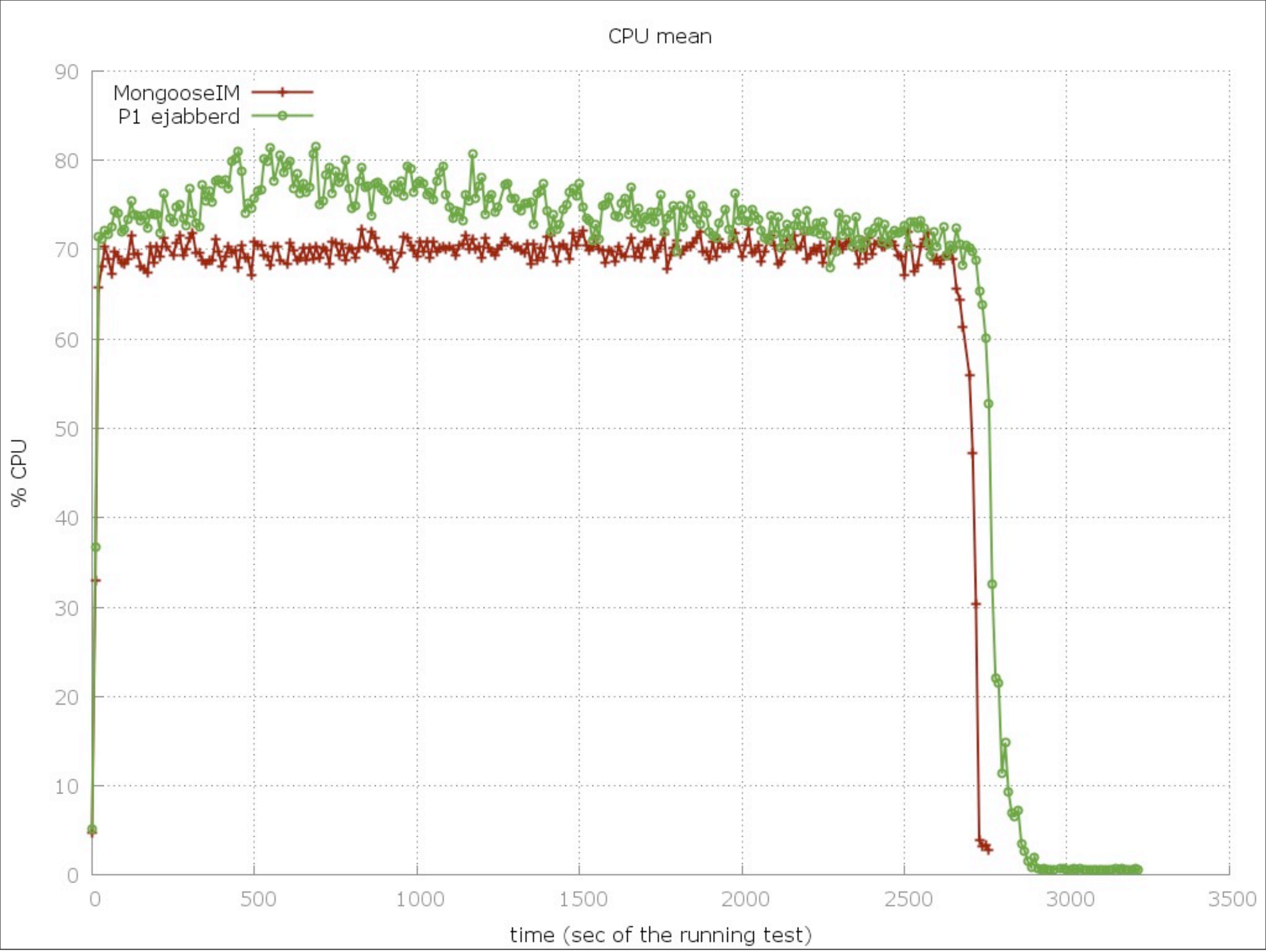
No message exchange

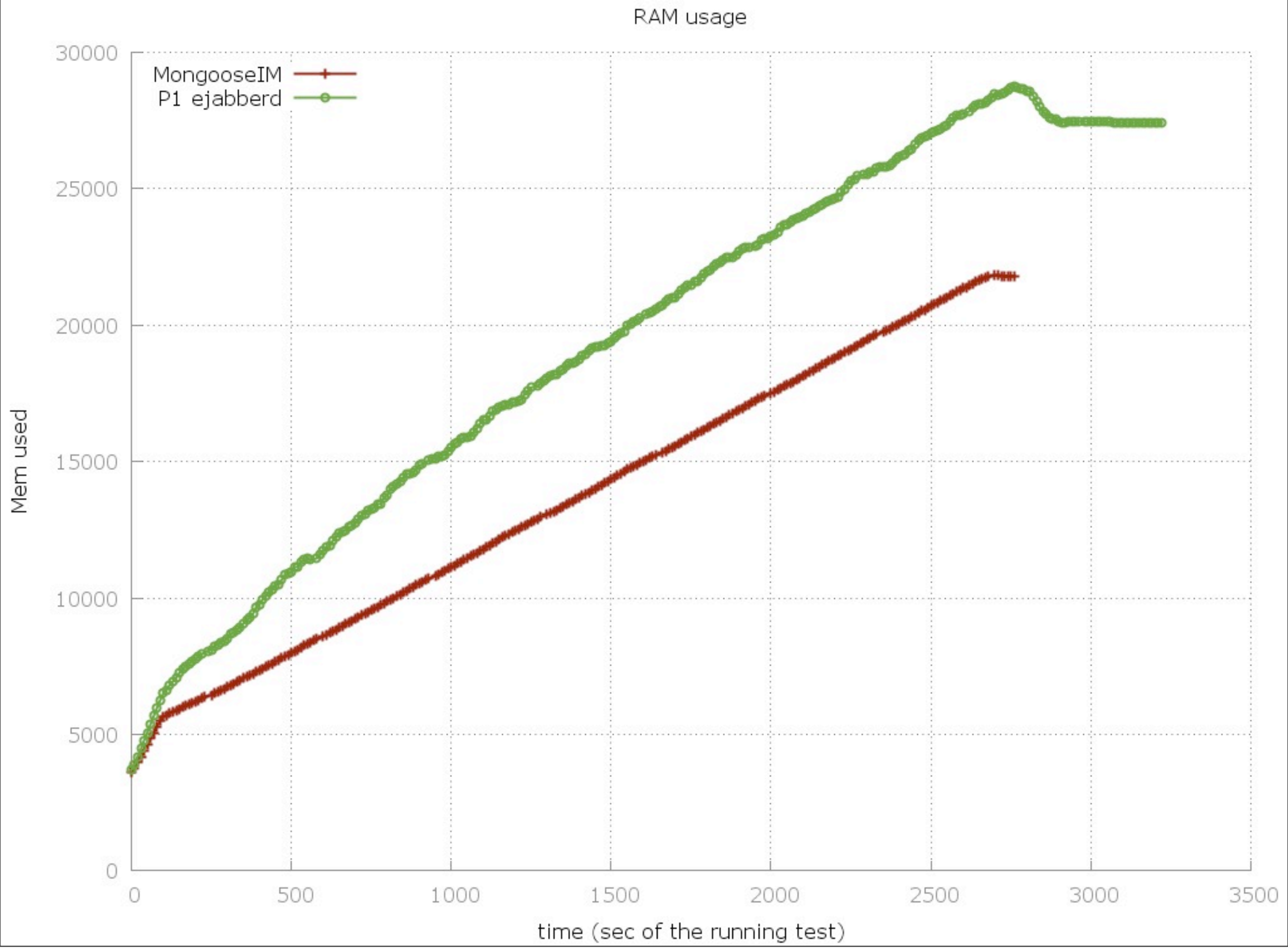
No message exchange

The aim was to check resource usage with maximum possible count of online users connected to server.



Tuesday, 11 December 2012


CPU mean
MongooseIM
P1 ejabberd
% CPU
time (sec of the running test)
0
10
20
30
40
50
60
70
80
90
0
500
1000
1500
2000
2500
3000
3500


RAM usage
MongooseIM
P1 ejabberd
Mem used
30000
25000
20000
15000
10000
5000
0
0
500
1000
1500
2000
2500
3000
3500
time (sec of the running test)

Mean connection request duration

MongooseIM
P1 ejabberd

Requests duration (msec)

5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0

0
500
1000
1500
2000
2500
3000
3500

time (sec of the running test)

Mean login duration

MongooseIM
P1 ejabberd

Duration (msec)

90000
80000
70000
60000
50000
40000
30000
20000
10000
0

0
500
1000
1500
2000
2500
3000
3500

time (sec of the running test)

Load tests - “max message rate”

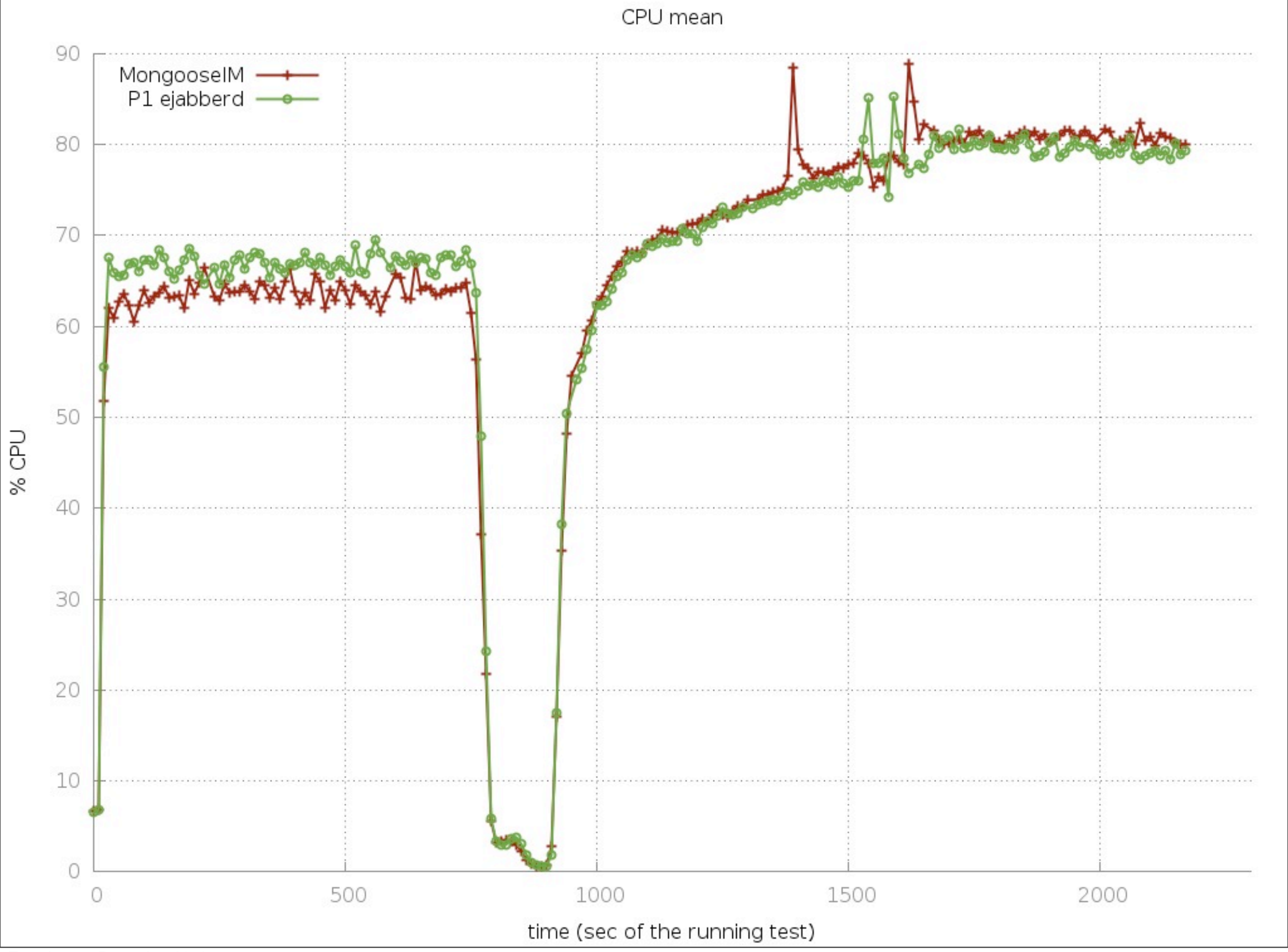
User arrival rate: 100/s

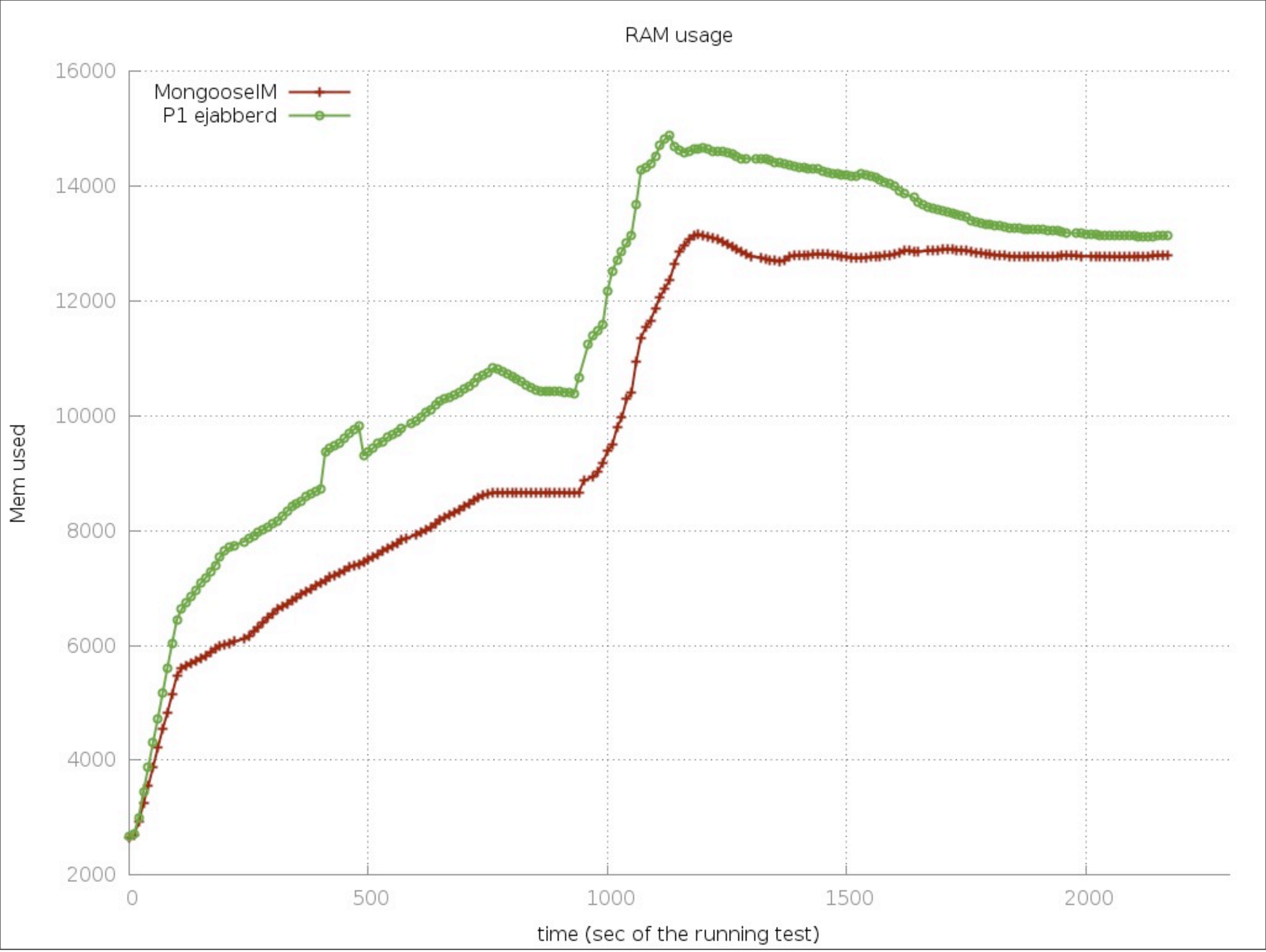
User count: 75k

Roster size: 100

Message rate: ~21k per second

The aim was to check resource usage and message delivery latency under heavy load.


CPU mean
MongooseIM
P1 ejabberd
% CPU
90
80
70
60
50
40
30
20
10
0
0
500
1000
1500
2000
time (sec of the running test)


RAM usage
MongooseIM
P1 ejabberd
Mem used
16000
14000
12000
10000
8000
6000
4000
2000
0
500
1000
1500
2000
time (sec of the running test)

client-server web techniques

- Long-Polling
- WebSockets
- ServerSent Events
- Socket.io

XMPP web techniques

- XMPP over Long-Polling (BOSH)
  - more expensive, long latency

- XMPP over WebSockets
  - communicating over WebSockets is sending messages, which is the same way XMPP communicates

XMPP web techniques

- XMPP client libraries for JavaScript
  - JSJaC (BOSH and WebSocket)
  - Strophe.js (BOSH, there is a branch with WebSocket)

WebSocket support on desktop

	IE	Firefox	Chrome	Safari	Opera
3 versions back	7.0	14.0	20.0	4.0	11.5
2 versions back	8.0	15.0	21.0	5.0	11.6
Previous version	9.0	16.0	22.0	5.1	12.0
Current	10.0	17.0	23.0	6.0	12.1
Near future		18.0	24.0		12.5
Farther future		19.0	25.0		

source: caniuse.com

WebSocket support on desktop

	IE	Firefox	Chrome	Safari	Opera
3 versions back	7.0	14.0	20.0	4.0	11.5
2 versions back	8.0	15.0	21.0	5.0	11.6
Previous version	9.0	16.0	22.0	5.1	12.0
Current	10.0	17.0	23.0	6.0	12.1
Near future		18.0	24.0		12.5
Farther future		19.0	25.0		

source: caniuse.com

WebSocket support on mobile

Show all versions	IOS Safari	Android Browser	Opera Mobile	Blackberry Browser	Chrome for Android	Firefox for Android
	4.0-4.1	2.3				
	4.2-4.3	3.0	10.0			
	5.0-5.1	4.0	11.5			
Current	6.0	4.1	12.0	7.0	18.0	15.0
Near future			12.1	10.0		

source: caniuse.com

WebSocket support on mobile

Show all versions	IOS Safari	Android Browser	Opera Mobile	Blackberry Browser	Chrome for Android	Firefox for Android
	4.0-4.1	2.3				
	4.2-4.3	3.0	10.0			
	5.0-5.1	4.0	11.5			
Current	6.0	4.1	12.0	7.0	18.0	15.0
Near future			12.1	10.0		

source: caniuse.com

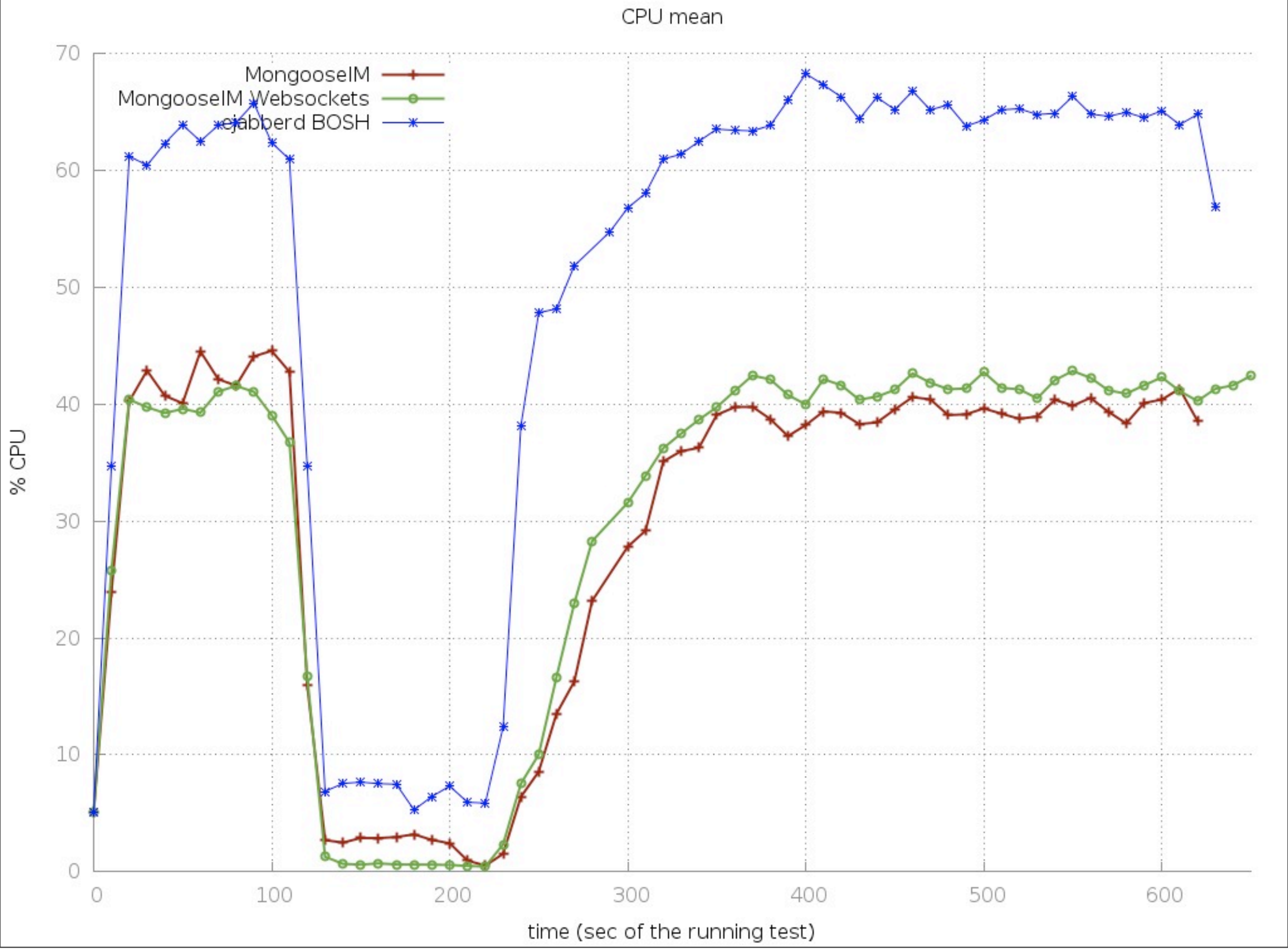
Load tests - “WebSockets vs. BOSH”

User arrival rate: 90/s

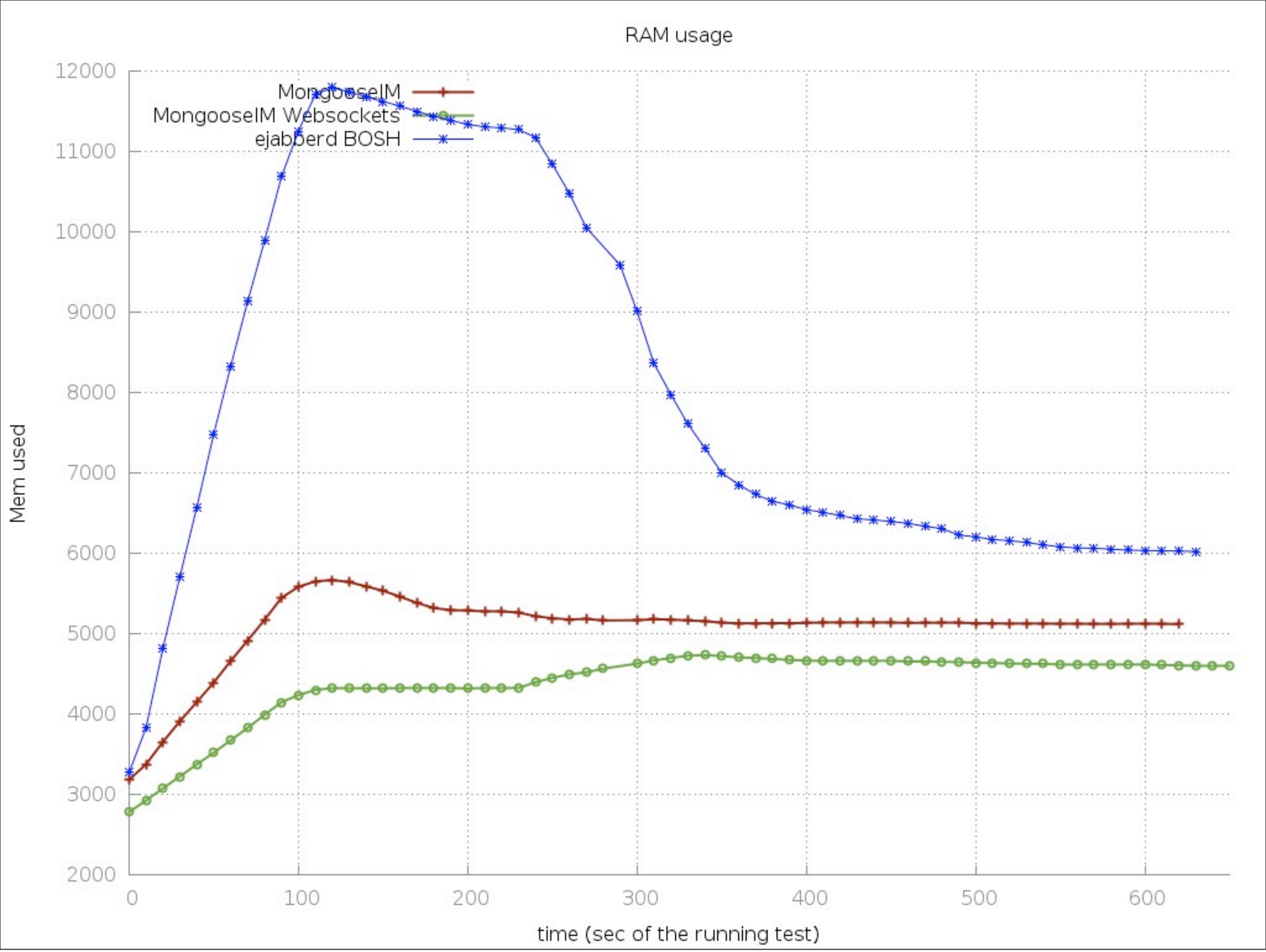
User count: 10k

Message rate: 6k/s

The aim was to compare resource usage of WebSockets vs. BOSH.



Tuesday, 11 December 2012


RAM usage
MongooseIM
MongooseIM Websockets
ejabberd BOSH
Mem used
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
0
100
200
300
400
500
600
time (sec of the running test)

Conclusions

Conclusions

Chat feature

- messengers
- social networking
- hijacked by other than chat applications

Conclusions

Chat feature

- messengers
- social networking
- hijacked by other than chat applications

XMPP

- open-standard protocol
- used in web and games

Conclusions

Chat feature

messengers

social networking

hijacked by other than chat applications

XMPP

open-standard protocol

used in web and games

XMPP over WebSockets

more efficient than BOSH

MongooseIM

Download

www.erlang-solutions.com/downloads/

Fork and contribute

https://github.com/esl/ejabberd

Contact us

ejabberd@erlang-solutions.com