Making WallStreet talk with GO

Matthew Campbell

Who Am I

- New York / Bangkok
- Run GO Meetup in Bangkok
- bloomberg.com (Rails)
- gucci.com (Rails)
- Co founder ErrPlane (GO)
- Thomson Reuters Messenger (GO)

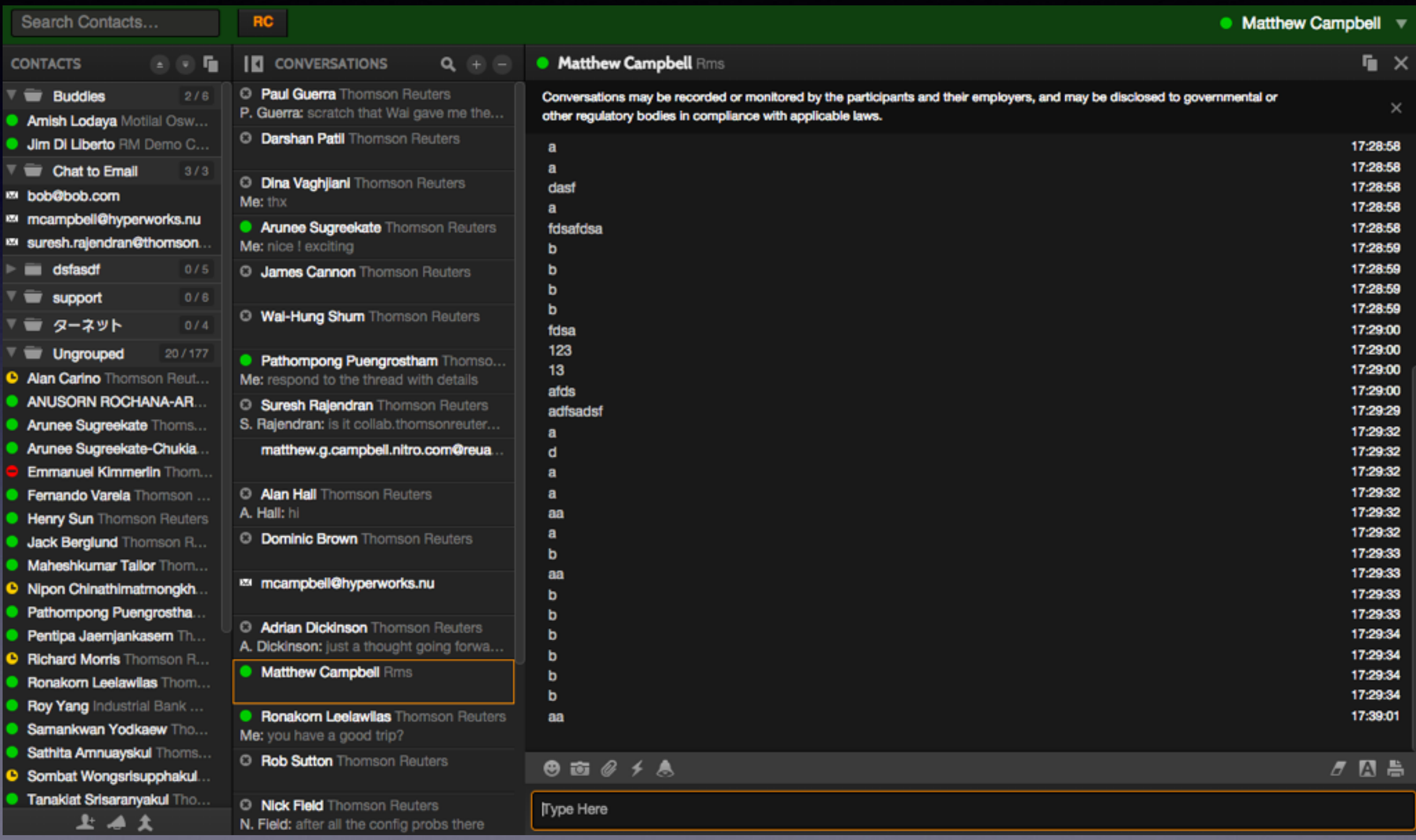
GOLANG Bangkok 2015

#langfight

What is this talk about?

- Running GO with a 300k users in production for multiple years
- Monitoring Large scale GO Servers
- Remote teams and GO

Eikon Messenger



Eikon Messenger

- 300k Traders / Fund Managers
- Federation with 20+ Bank/Financial Networks
- Strict compliance rules in 100+ countries
- Global usage, largest in SE Asia
- Downtime == Lost Trades == Upset customers

Protocols

- XMPP (Most XEPS)
- SIP (Microsoft SIP)
- BOSH (http xmpp)
- Chatrooms

Team

- 80% remote team
- Development
  New York/Canada/Portugal/Germany/Bangkok
- Ops London
- QA Bangkok

Remote Dev

- Daily Google Hangouts
- Pivotal Tracker
- Github pull request, code reviews
- Xmpp Chatroom, all day multi time zone
- Email group

Issues with GO

Missing Features

- Decent XML Dom Parser
- Regex with decent performance
- Serializing performance / allocations
- Better CGO integration

Garbage Collection

- Our single largest problem. Our app is latency sensitive.
- We have automated test suite to detect garbage collection problems
- Avoiding Heap allocations
- Managing Size of Heap

IO Latency

- Many Layers
- In process cache
- Local Redis instance on machine
- Mysql primary storage
- NSQ for delayed message queuing

Mysql (Roster Information)

Mysql (Roster Information)

Redis Server (Routing Information)

Redis Server (Routing Information)

Roster Storage

Presence Routing

Redis Server (Routing Information)

Redis Server (Routing Information)

Legacy C# Server

REDIS

REDIS

REDIS

REDIS

ETCD (Configuration)

In Memory Cache

Nitro

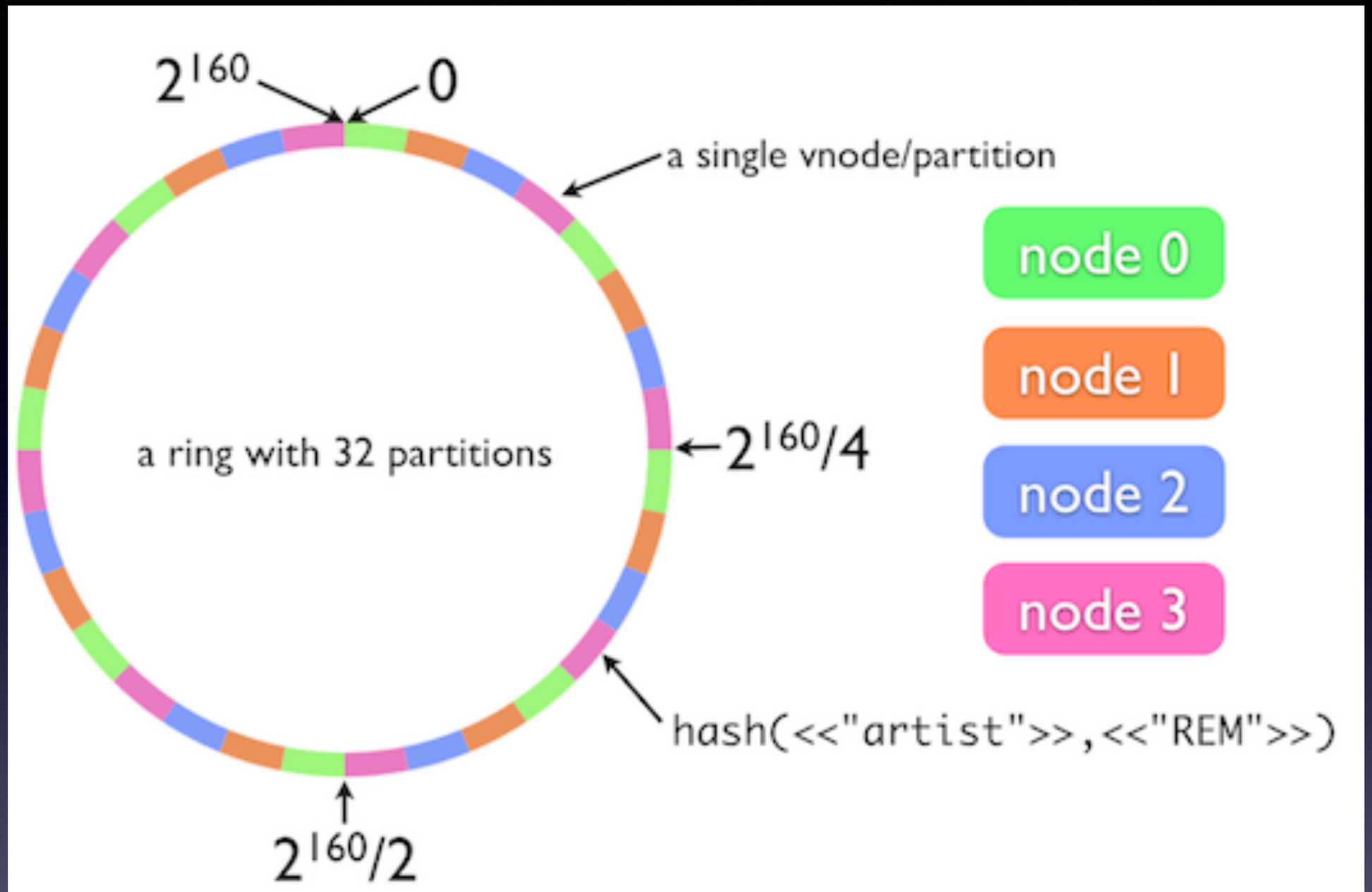
Single Physical Blade Server

NSQ

Mysql Compliance Message Storage

Compliance and Message Storage

ElasticSearch Long Term Message Storage



Clustered Redis

source: paperplanes.de

CGO Issues

- Libxml2 (Gokogiri) , PCRE (Perl Regex)
- Issues with to many Pthreads
- 1.4 broke a lot of C libraries, with stack moving
- Broke GDB for a while
- Faster sometimes, sometimes significantly slower cause of thread dispatch

Following GO Libraries

- For the first year, we tracked all GO dependencies
- We often time found bugs in upstream libraries with our automated integration suite
- Over time this got to risky, and we chose to freeze our GO Libraries

Tracking GO Lang

- PAINLESS !
- We track Go Lang official releases, within 2 weeks of each release we have gone to production using the new compiler
- We have had a few integration tests break during upgrades
- Only production code that broke, was in CGO ;(

Continuous Delivery

- Jenkins
- Integration level tests, that fire up a cluster of our servers, and validates user level interactions at protocol level. Including federating with several open source messaging servers
- Unit tests for functional areas

Plugin Architecture

- Plugins are processes
- Go doesn’t support dynamic linking
- Plugins talk to main process via TCP
- Plugins can be run in process, but still use tcp, to simulate exact same properties

Chat To Email

Yahoo

TCP

Server #1

Twitter

Nitro

TCP

Server #2

Chatrooms

Metrics

- Annotate everything
- Us GO-Metrics
- Graphite / InfluxDb

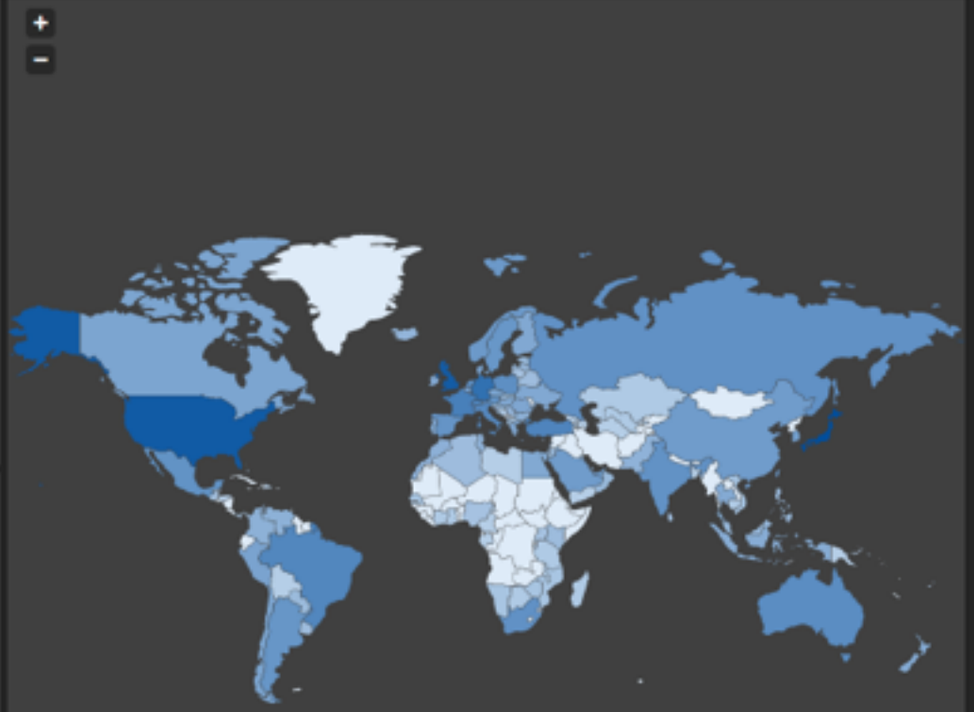
ukbp-claicpr01	ukbp-claicpr02	ukbp-claicpr03
0	0	0
ukbp-claicpr04	ukbp-claicpr05	ukbp-claicpr06
0	0	0
ukbp-claicpr07	ukbp-claicpr08	ukbp-claicpr09
0	0	0
ukbp-claicpr10	ukbp-claicpr11	ukbp-claicpr12
0	0	0

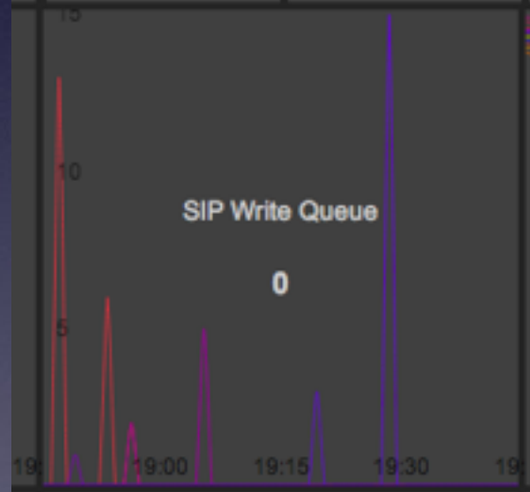
Messages in Queue (Xmpp)

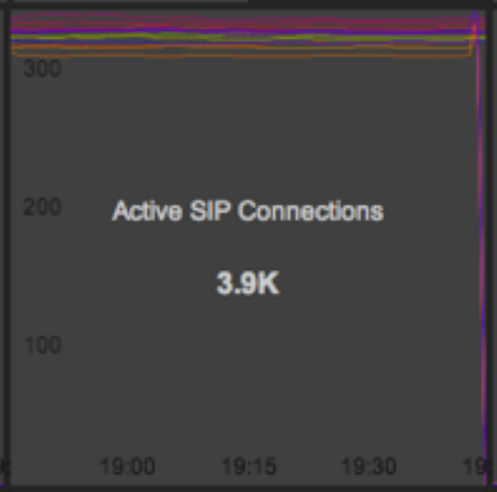
0

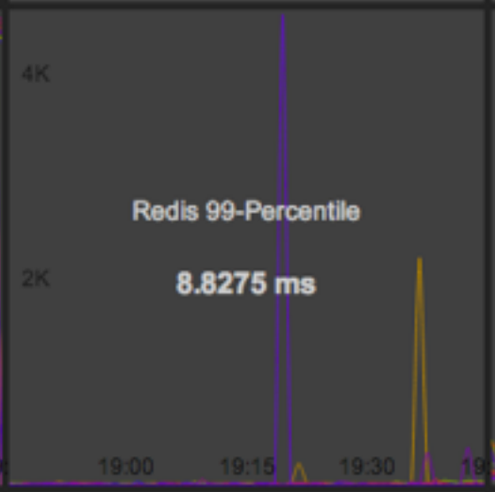
Active Xmpp(Bosh) Conn

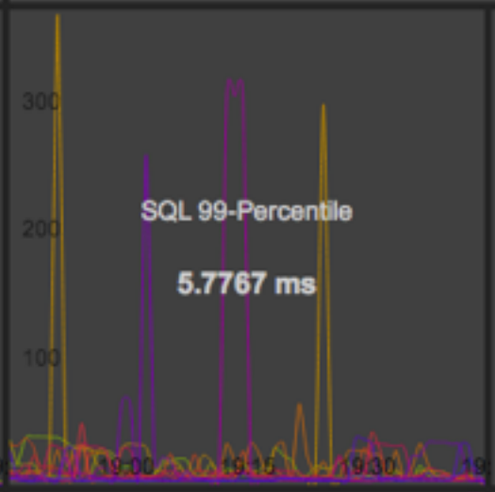
30.1K

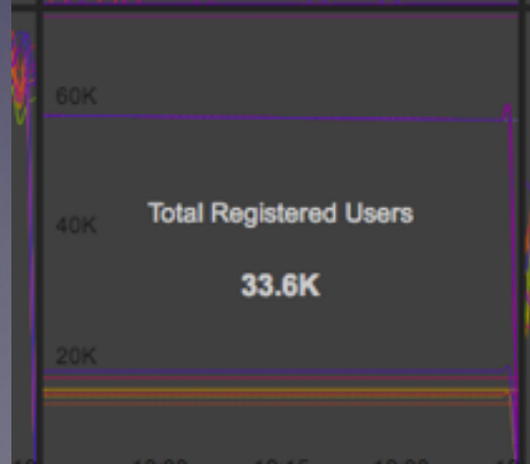


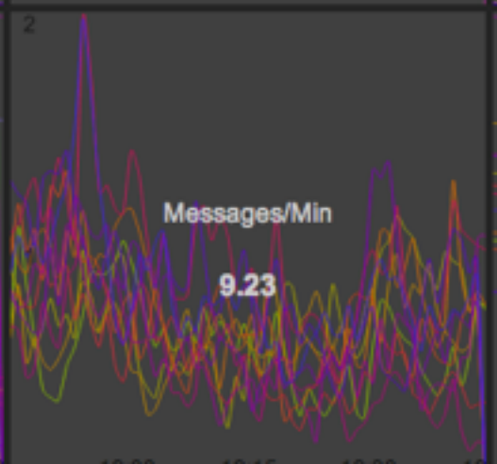


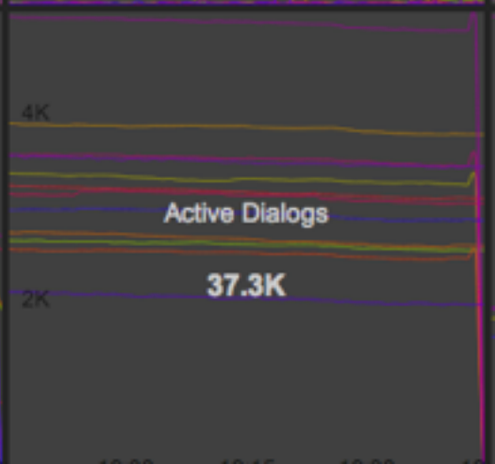


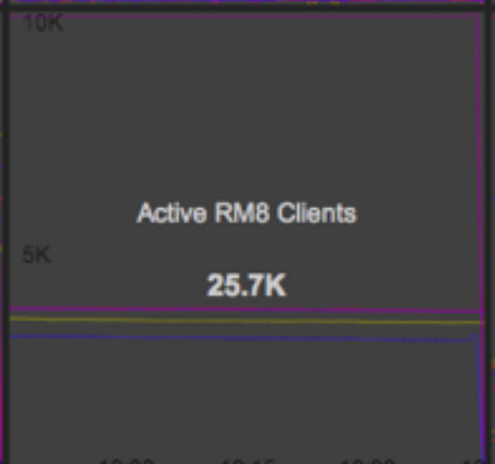


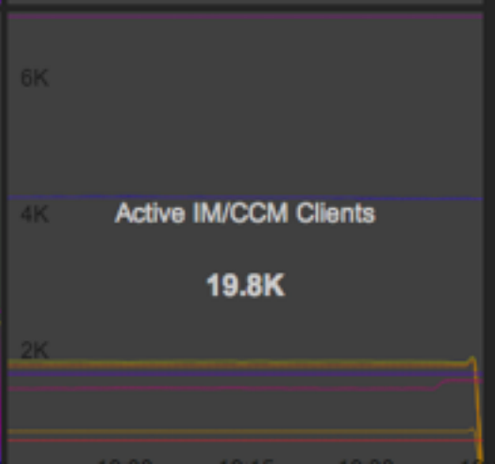












Single Points of Failure

- Mysql Multimaster
- Redis Master / Slave
- Sentinel for Mysql Failover
- Custom scripts + ETCD for Mysql Failover

Packaging

- RPMs
- FPM Ruby Gem
- Single binary + Configs

Managing Config

- ETCD vs Zookeeper
- Push
- Pull

Compliance

- NSQ
- Elastic Search
- Mysql
- C# process

Mysql (Roster Information)

Mysql (Roster Information)

Redis Server (Routing Information)

Redis Server (Routing Information)

Roster Storage

Presence Routing

Redis Server (Routing Information)

Redis Server (Routing Information)

Legacy C# Server

REDIS

REDIS

REDIS

REDIS

ETCD (Configuration)

In Memory Cache

Nitro

Single Physical Blade Server

NSQ

Mysql Compliance Message Storage

Compliance and Message Storage

ElasticSearch Long Term Message Storage

Upgrades

- In place upgrades
- Spawn a new process
- Pass file handles for TCP via a Named Pipe
- New process kills parent

What's Next

- Multiple live datacenters, currently Hot/Cold
- Removal of all CGO Code
- Using GO Generate for config classes

Q & A

Contact

- Feel Free to email me any further questions
- We’re Hiring!
- mcampbell@hyperworks.nu