Welcome!

# Scaling with Erlang

Building a large pubsub-with-history system

# *Me*

Paul Peregud

- Senior developer at SilverSoft Sp. z o.o., tech lead of Talos project for Onet S.A.
- Developer at LivePress Inc.

*paulperegud@gmail.com*

# Topics

- API
- Performance
- Scalability
- Reliability

# Business side

- Events
  - Sports
  - Politics
  - Tech
- Authors
- Delivery system (aka Talos system)
- Client's browser

# Hardware

- 3 nodes
  - Xeon E5-2650
  - 32 cores
  - 166 GB RAM
  - 1 Gbps

# APIs

- SockJS (end users)
  - Websocket
  - Xhr-streaming
  - Jsonp-polling
- HTTP REST (authors)

# SockJS

- Original
  - Up to 100k per node
  - Latency long tail (up to 60 sec)

# SockJS rewrite

- 1 active process per client (any protocol)
- avoid multiple encoding of same data
- mobile browsers support added

# Results?

- No lock congestion on timers
- Up to 700k of connections per node
- Stable, low latency

# Broadcast speed

- 1.1 latency for 500k users (as measured for 95 percentile for messages of size 500b)

# Scalability

- SMP
- Minimizing
- Automatization
- Logging and debugging
- Tools

# SMP

- Pubsub-with-history evolution
  - Single process pubsub
  - Public ets
  - Next step:
    - named public ets per scheduler
    - phash2(id(), n)
    - up to 300k of subscribes per second!

# Other stuff?

- Things like real-time stats are helpful, but non-essential. Make sure they are not taking too much resources. Design it to be self-limiting

# Performance

- Latency, latency, latency!
- Socket accept rate
  - (troublesome with HTTP-based protocols)
- Broadcast speed
- Hardware issues

# HTTP accept rate?

- Add
  - More servers
  - More sockets
  - **More acceptors**

# Minimizing

- any message flow
- broadcast messages
- auxiliary messages
- subscriber counts
- stats messages

# Automation

- Up
- Down
- Limits

# Logging and debugging

- Single error does not matter!
- Statistics matter - watch out for elevated error rate!
- Logging may be expensive
- Out-Of-Band logging

# Tools

- web panels aka dashboards
- CLI tools
- hot code upgrades
- and **rollbacks!**

# Message delivery

- reordering
- caching and sending with subscribe with message id
- idempotency

# Mnesia

- Cons
  - It's not magic! ;)
- Pros
  - Zero configuration
  - Blazing fast reads using mnesia:fun/2

# Unanswered questions

- Removing dead node from cluster
- Network spit
- Limits of scaling for single writer scenario

# Fault tolerance

- Nodes can and will crash
- Automatic reconnect is easy with static list of nodes

# Talos authors

- Paul Peregud
- Gleb Peregud
- Peter Flis