

Erlangen: asynchronous, distributed  
message passing for Clozure CL  
(lightning talk for ELS 2017)

Max Rottenkolber <max@mr.gy>

Tuesday, 4 April 2017

# Erlangen

- asynchronous message passing
- reliable (supervision trees)
- inherently distributed
- inspired by Erlang, yet very different

## “You can't do that, it won't be like Erlang!”

- uses native OS threads (can't have 10,000)
- does not enforce strict isolation (within the boundary of a node)
- Common Lisp is synchronous (blocking I/O)

... there are also many reasons to prefer CL/Erlangen over Erlang

- for instance, Erlangen's message queues are bounded and never leak memory

# I did it, its amazing: lessons learned

- *uses native OS threads*

even if processes are heavy, supervision trees are worth their weight in gold

- *does not enforce strict isolation*

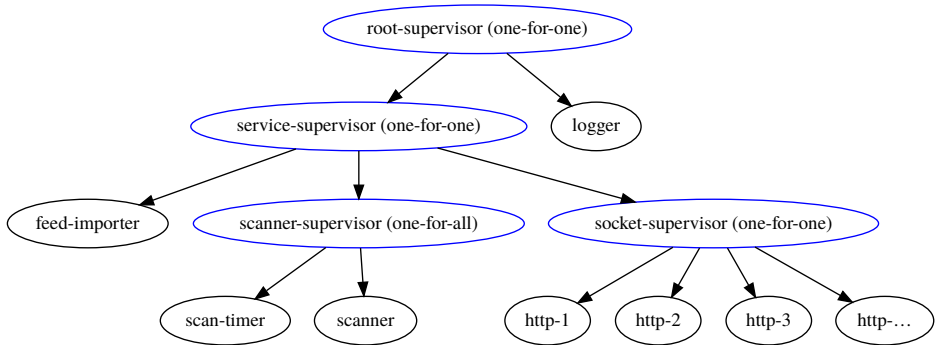
just don't modify objects you send (send = free)

- *Common Lisp is synchronous (blocking I/O)*

tempted to look at asynchronous I/O (IOLib): complete dead end, no tasteful way to write code

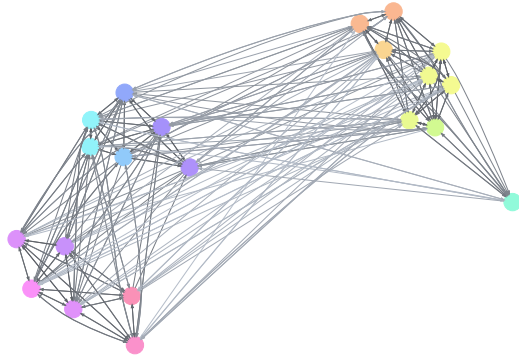
blocking is fine as long as you always supply a timeout

# Write reliable services in Common Lisp



Supervision tree of a real-life application (processes can be spread on remote nodes)

# Research: explore distributed designs



live visualization of 300 SLOC Kademia DHT implementation on Erlangen

## Numbers (on Intel Xeon E31245 3.30GHz)

- 5us for message delivery, 10x more than erts 7.3 (that's not even bad, actually)
- up to 2 million messages per second
- you can have 1,000 processes easily

...no serious optimization work done so far!

## No SBCL, . . . support because

- portability layers are boring
- portability layers are boring
- portability layers are boring
- I won't write one *or* maintain one
- Erlangen already includes modifications to CCL
- . . . maybe I want to swap out CCL's threading, GC, I/O, . . . implementations?
- `erlang-kernel` is distributed as an executable that includes Quicklisp



# Hack it!

- [github.com/eugeneia/erlang](https://github.com/eugeneia/erlang) AGPL-3.0
- branch, fork all you like
- nothing is set in stone yet, wild ideas welcome
- mail me at [max@mr.gy](mailto:max@mr.gy)

i n t e r  
s t e l l a r