

Erlang: The Perl of Functional Languages



POE: The Erlang of Perl

Kevin Scaldeferri

Portland Perl Mongers

12 Dec 2007

“I apologize for the length of this talk,  
but I didn't have time to make it shorter.”

*Mark Twain*

How did we end up with this talk?

# Models of Concurrency

- Processes
- Threads
- Transactions
- Message Passing

# Processes (Unix)

- Heavyweight
- Highly Isolated
- Safe & Robust
- Moderately Easy to Scale

# Threads

- Fairly lightweight
- (Almost) everything shared
- Fragile
- Easy to scale on 1 machine
- Hard to scale to N machines

# Transactions

- Databases, STM
- Fairly heavyweight
- Very good isolation, safety, robustness
- Hard to scale

# Message Passing (Actor Model)

- Concurrency via messages exchanged between actors
  - Erlang processes
  - POE sessions
- Very lightweight
  - [Shootout](#)

# Erlang

- A pure functional, weakly dynamically typed, interpreted language with concurrency primitives
- Syntax similar to Prolog
- Created at Ericsson for telecom applications in 1987

# Erlang Message Passing

- Nothing Shared
- Very Robust
- Easy to Scale

# Erlang Concurrency Primitives

- spawn - create a process
- ! - send a message to a process
- receive - listen for a message

`urlchecker.erl`

# POE

- “a framework for creating event-driven multitasking programs in Perl”

# POE::Kernel

- Coordinates everything
- Maintains a queue of events
- Parcels them out to Sessions to do work
- Figures out when there's nothing left to do

# POE::Session

- Processes / state machines
- Dispatch to subroutines based on events received
- Parent-child relationships

# Wheels

- Patterns that people kept reinventing

# Components

- Lots of these
  - HTTP
  - IRC
  - DNS
  - SMTP
  - ...

[urlchecker.pl](http://urlchecker.pl)

[ircbot.pl](http://ircbot.pl)

**ircbot.erl**

# Erlang vs. POE

- Preemptive
- 1 mailbox / process
- SMP & distribution
- A book
- Fault-tolerant
- Peculiar conventions
- Cooperative
- Single event queue
- No SMP
- No book
- Not really
- Peculiar conventions

# Erlang vs. Perl

- Dynamic
  - Interpreted, but HiPE
  - Pure functional
  - [CEAN](#)
  - A book
  - Not so much
  - Incomprehensible error messages
- Dynamic
  - Interpreted
  - Multi-paradigm
  - CPAN
  - Many books
  - Lots of jobs
  - Generally good error messages

# Erlang vs. Perl, continued

- REPL
- Lousy strings, file I/O
- Binary pattern matching
- Hot upgrades
- No REPL (debugger)
- Excellent strings, file I/O
- Pack/unpack
- Don't try this at home

# Resources

- <http://poe.perl.org/>
- #poe on irc.perl.org
- [poe@perl.org](mailto:poe@perl.org)
  
- <http://erlang.org/>
- <http://trapexit.org/>
- #erlang on freenode
- Erlang-questions@erlang.org