Phoenix Akka/Akka.Net Users Group

Purpose

- To build an Akka community in the Phoenix metro area.

Meeting Places

- Gangplank in Chandler.
- Concord Servicing in Scottsdale.

Sponsor

- Looking for one - we want pizza! (And beer.. .:))

Akka.Net

What is it & Why should you care?

> whoami

Alfredo Herrera
DevOps Engineer @ Concord Servicing

We're hiring!

What is Akka?

"Akka is a toolkit for building highly concurrent, distributed, and resilient message-driven applications"

At it's core it's a..

- Concurrency Model
- Error Handling Model
- Event-based Model

It also can be a...

- Simplified development
- Distributed Computing Model

Concurrency Model

- Akka is an implementation of the Actor Model of computation.
  - Erlang-like.
- Actors, are shared-nothing units of computation
  - Can be thought of as lightweight processes.
  - A single machine can have millions running concurrently.
  - “High Performance 50 million msg/sec on a single machine.”
  - “Small memory footprint; ~2.5 million actors per GB of heap.”
  - “400 bytes per actor.”
- Actors are isolated processes
  - i.e. Cannot get to another actor's’ state directly.
- Actors communicate with other actors via asynchronous message passing.
  - Most commonly fire-and-forget messages.
  - Wait in own mailbox for response.

Error-Handling Model

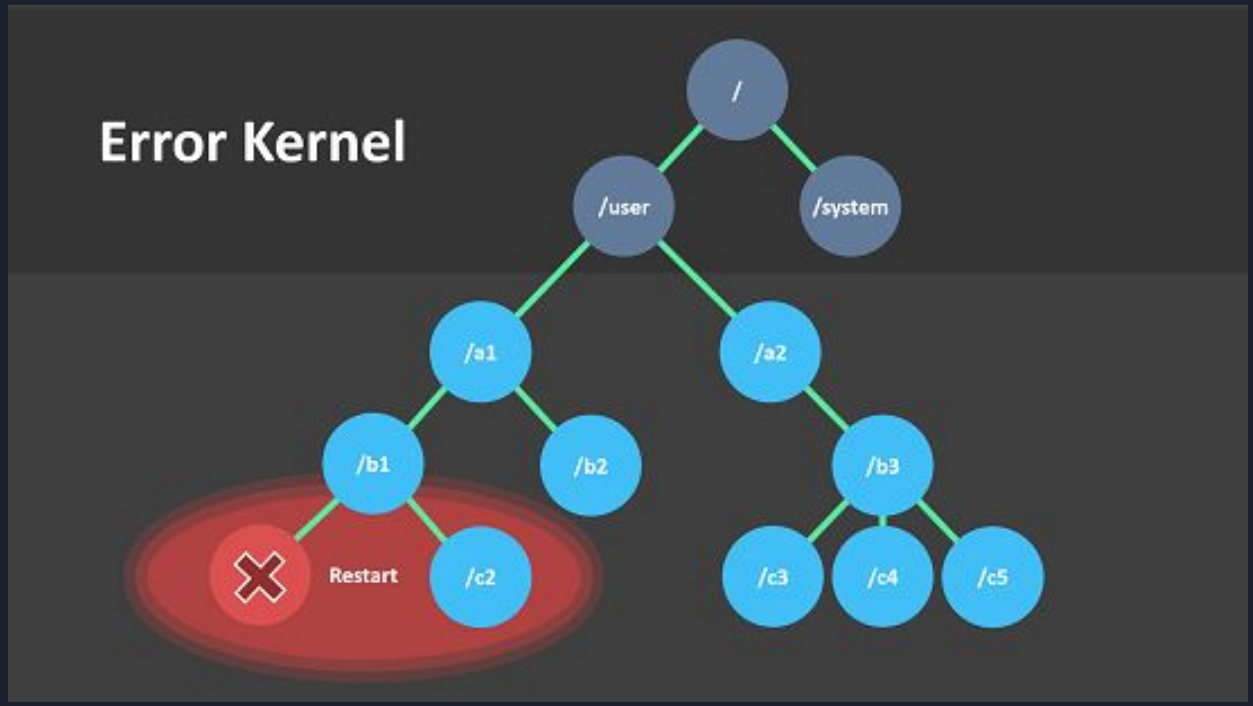
If and actor dies - it’s parent is notified.

- Build supervision trees.
- Place riskiest operations/work in leaves of tree.

“Let it crash” or “Program the happy path”

- Minimize try/catch and error-condition checking in business code
- Easier to reason about, cleaner.

Supervision Tree



Event-Based Model

Messages

- All communication between actors done via immutable messages.
- Facilitates integration with other services (i.e. Kafka).

State - No longer evil!

- Actor’s Mailbox is processed synchronously.
- Ideal for handling state.
- Events can be persisted.

Using Akka

When to consider Akka.

- High resiliency requirements.
- Event-driven architecture.
- Microservices (especially if modeling domain in events).

When Akka would not be appropriate.

- Simple or small apps.
- Computation intentive requirements.
- Already using Erlang or Elixir.

Challenges

- Immutability - must be done by the developer.
- Lack of familiarity in enterprise IT departments.
- Not yet widely adopted by .Net shops. Vicious circle.

Demo Code

https://github.com/alfredherr/MeetUpDemos/tree/master/AkkaDemo

Questions