Hot Loading Code

Using Erlang/Elixir to Deploy Code without restarting

Kenny Ballou

/dev/null > labs

November 27, 2017

1. Motivation
2. Hot Loading in IEx
3. Relups

About Me

- Hacker
- Developer (read gardener)
- Student
- @kennyballou
- github/kennyballou
- `https://git.devnulllabs.io`
- `https://kennyballou.com`

1 Motivation

2 Hot Loading in IEx

3 Relups

Hot Loading Code

Problem

- Deploying new code requires downtime
- Downtime can be exacerbated by cold boot problem
- Deployments are error prone
  - Especially when achieved manually
- Minimizing downtime can be expensive
- Automating deployments *can* be a herculean task

Hot Loading Code

Using Erlang/OTP, we can:

- Load new versions of modules/processes
- Load new versions of application trees

Hot Loading Code

Problems with Hot Loading Code

- Hard to do right
- Difficult to test

Deployment Elsewhere

Related Technologies and Approaches

- Kubernetes[1]
- “Rolling” Releases
- Blue/Green[13]
- AWS Elastic Container Service[2]

1 Motivation

2 Hot Loading in IEx

3 Relups

Manually Loading New Code

Black Magic

- `:sys.suspend/1`
- `:code.load_file/1`
- `:sys.change_code/4`
- `:sys.resume/1`

Hot Loading Code

Example Code

```elixir
def get(pid \\ __MODULE__, key, default \\ nil)
def get(pid, key, default) do
  GenServer.call(pid, {:get, key, default})
end

def put(pid \\ __MODULE__, key, value)
def put(pid, key, value) do
  GenServer.call(pid, {:put, key, value})
end

def handle_call({:get, key, default}, _caller, state) do
  {:reply, Map.get(state, key, default), state}
end

def handle_call({:put, key, value}, _caller, state) do
  {:reply, :ok, Map.put(state, key, value)}
end
```

Hot Loading Code

Erlang Functions

```
% elixirc kv.ex
% iex
iex> l KV
{:module, KV}
iex> KV.start_link()
{:ok, #PID<0.84.0>}
iex> KV.get(:a)
nil
iex> KV.put(:a, 42)
:ok
iex> KV.get(:a)
42
```

Hot Loading Code

Example Patchset

```
@@ -20,10 +21,12 @@ defmodule KV do
   end

   def handle_call({:get, key, default}, _caller, state) do
+    Logger.info("#{__MODULE__}: Handling get request for #{key}")
     {:reply, Map.get(state, key, default), state}
   end

   def handle_call({:put, key, value}, _caller, state) do
+    Logger.info("#{__MODULE__}: Handling put request for #{key}:#{value}")
     {:reply, :ok, Map.put(state, key, value)}
```

Hot Loading Code

Erlang Functions

```
% patch kv.ex kv.ex.patch
% elixirc kv.ex
```

Hot Loading Code

Erlang Functions

```
iex> :code.load_file(KV)
{:module, KV}
iex> :sys.suspend(KV)
:ok
iex> :sys.change_code(KV, KV, 0, nil)
:ok
iex> :sys.resume(KV)
:ok
```

Hot Loading Code

Erlang Functions

```
iex> KV.get(:a)
21:28:47.989 [info]  Elixir.KV: Handling get request for a
42
iex> KV.put(:b, 2)
21:28:53.729 [info]  Elixir.KV: Handling put request for b:2
:ok
```

Hot Loading Code

Using IEx Helpers

- `c/1` — compile
- `l/1` — load
- `r/1` — (recompile and) reload

Hot Loading Code

Using IEx Helpers

```
iex> r KV
warning: redefining module KV (current version loaded from Elixir.KV.beam)
  kv.ex:1

{:reloaded, KV, [KV]}
iex> KV.get(:a)

21:52:47.829 [info]  Elixir.KV: Handling get request for a
42
```

1 Motivation

2 Hot Loading in IEx

3 Relups

Hot Loading Code

- Manually swapping code isn't better
- More error prone than cold releases

Hot Loading Code

Release Upgrades — Relups

- Describe *how* to upgrade Erlang applications
- Uses Erlang release (`.rel`) files
- Accomplishes upgrading an entire system
  - Application upgrades are composed to achieve entire system replacement

Hot Loading Code

Elixir Projects

- `exrm` [3]
  - Stable
  - Requires `relx` [4]
  - Poor Umbrella Support
  - Actively being replaced
- `distillery` [5]
  - ~~Beta~~ Stable
  - Pure Elixir
  - Better Umbrella Support
  - Bugs
    - Supervisors are (still) painful
    - May require manual .appup writing
  - Future

Hot Loading Code

Demo

References I

- https://kubernetes.io.
- https://aws.amazon.com/ecs.
- https://github.com/bitwalker/exrm.
- https://github.com/erlware/relx.
- https://github.com/bitwalker/distillery.
- http://erlang.org/doc/reference_manual/code_loading.html.
- http://erlang.org/doc/man/sys.html.
- http://elixir-lang.org/docs/stable/elixir/.
- http://elixir-lang.org/docs/stable/elixir/Code.html.

References II

- http://erlang.org/doc/man/relup.html.
- http://andrealeopardi.com/posts/handling-tcp-connections-in-elixir/.
- https://git.devnulllabs.io/demos/octochat.git.
- Martin Fowler.
  https://www.martinfowler.com/bliki/BlueGreenDeployment.html.

Hot Loading Code

Using Erlang/Elixir to Deploy Code without restarting

Kenny Ballou

/dev/null > labs

November 27, 2017