

Productive | > Reliable | > Fast

Phoenix framework



Milad Rastian

@slashmili

Why another language/framework?

Why another language/framework?

❖ Ruby is Slow

Why another language/framework?

- ❖ ~~Ruby is Slow~~
- ❖ Rails Performance

Why another language/framework?

- ❖ ~~Ruby is Slow~~
- ❖ ~~Rails Performance~~
- ❖ People are still complaining!

O.K. What are the alternatives?

O.K. What are the alternatives?

❖ Go

Go?

```
package main

import "github.com/gin-gonic/gin"

func main() {
    r := gin.Default()
    r.GET("/ping", func(c *gin.Context) {
        c.JSON(200, gin.H{
            "message": "hello world",
        })
    })
    r.Run() // listen and server on 0.0.0.0:8080
}
```

O.K. What are the alternatives?

- ❖ ~~Go~~
- ❖ Scala / Clojure

Scala/Clojure?



O.K. What are the alternative?

- ❖ ~~Go~~
- ❖ ~~Scala/Clojure~~

Why do we love Ruby?

- ❖ Community
- ❖ The joy of programming
- ❖ Productivity

You might still say it's not
performing as you want!

Seriously!?

What if?

What if?

- ❖ You can have all the benefits of Ruby

What if?

- ❖ You can have all the benefits of Ruby
- ❖ Without sacrificing performance.



Meet Elixir

Why Elixir/Phoenix?

❖ Fast

Erlang VM

Distributed

Highly Available



Fault-tolerant



Benchmark

Framework	Throughput (req/s)	Latency (ms)
Phoenix 0.13.1	179,685	0.61ms
Gin (GO)	176,156	0.65ms
Play (Java/Scala)	171,236	1.89ms
Rails	11,903	8.50ms

<https://gist.github.com/omnibs/e5e72b31e6bd25caf39a>

Why Elixir/Phoenix?

- ❖ Fast
- ❖ Reliable

Reliable

- ❖ Process Monitoring / Supervision
- ❖ Recover Failure

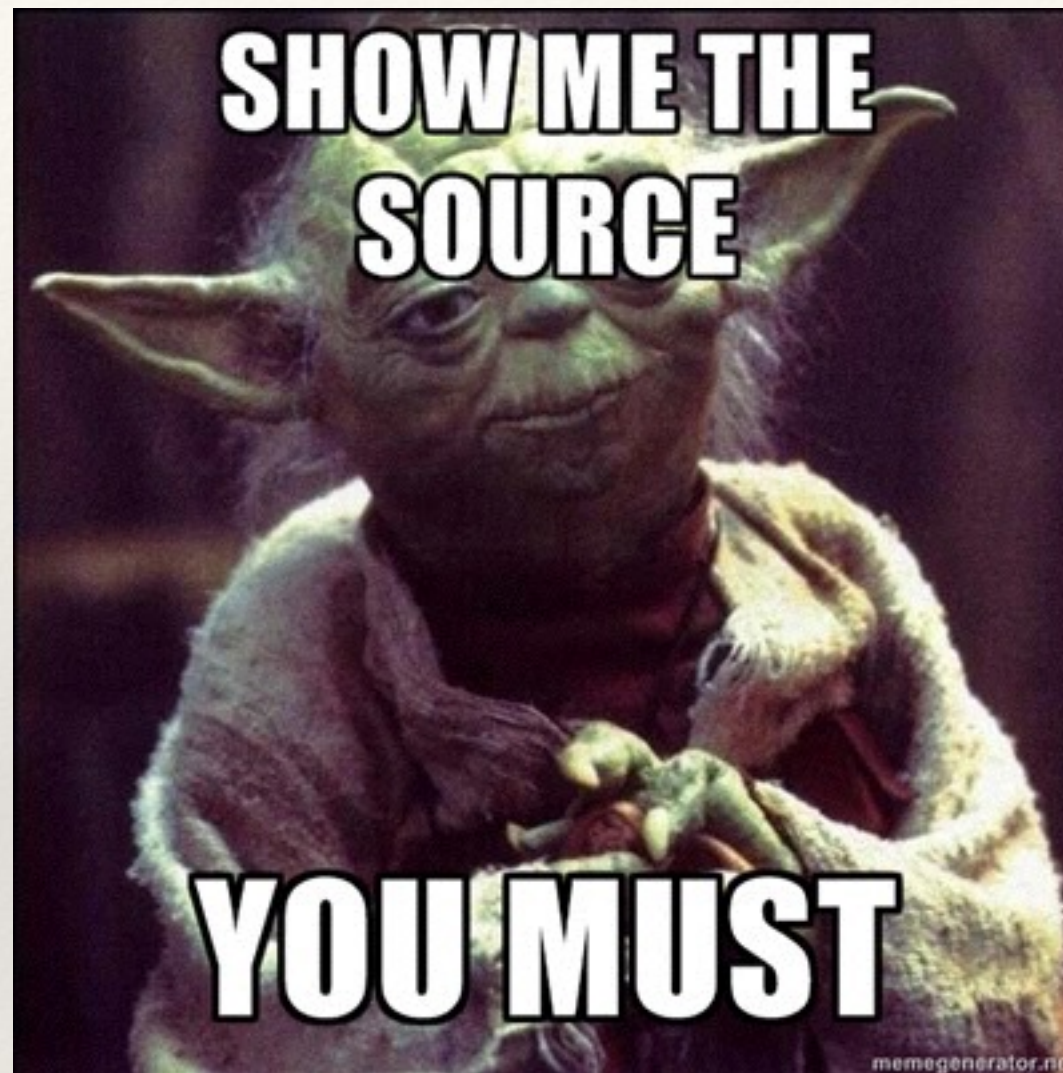
Why Elixir/Phoenix?

- ❖ Fast
- ❖ Reliable
- ❖ Productive

Productive

```
1 defmodule Etel.Router do
2   use Etel.Web, :router
3
4   pipeline :browser do
5     plug :accepts, ["html"]
6     plug :fetch_session
7     plug :fetch_flash
8     plug :protect_from_forgery
9     plug :put_secure_browser_headers
10  end
11
12  pipeline :api do
13    plug :accepts, ["json"]
14  end
15
16  scope "/", Etel do
17    pipe_through :browser # Use the default browser stack
18
19    get "/", PageController, :index
20  end
21
22  # Other scopes may use custom stacks.
23  scope "/api", Etel do
24    pipe_through :api
25  end
26 end
```


There is no need to trade beautiful, maintainable
code
to gain performance



<https://github.com/slashmili/talks/tree/master/2016/phoenix-klxrb-march>

Static Assets with Brunch

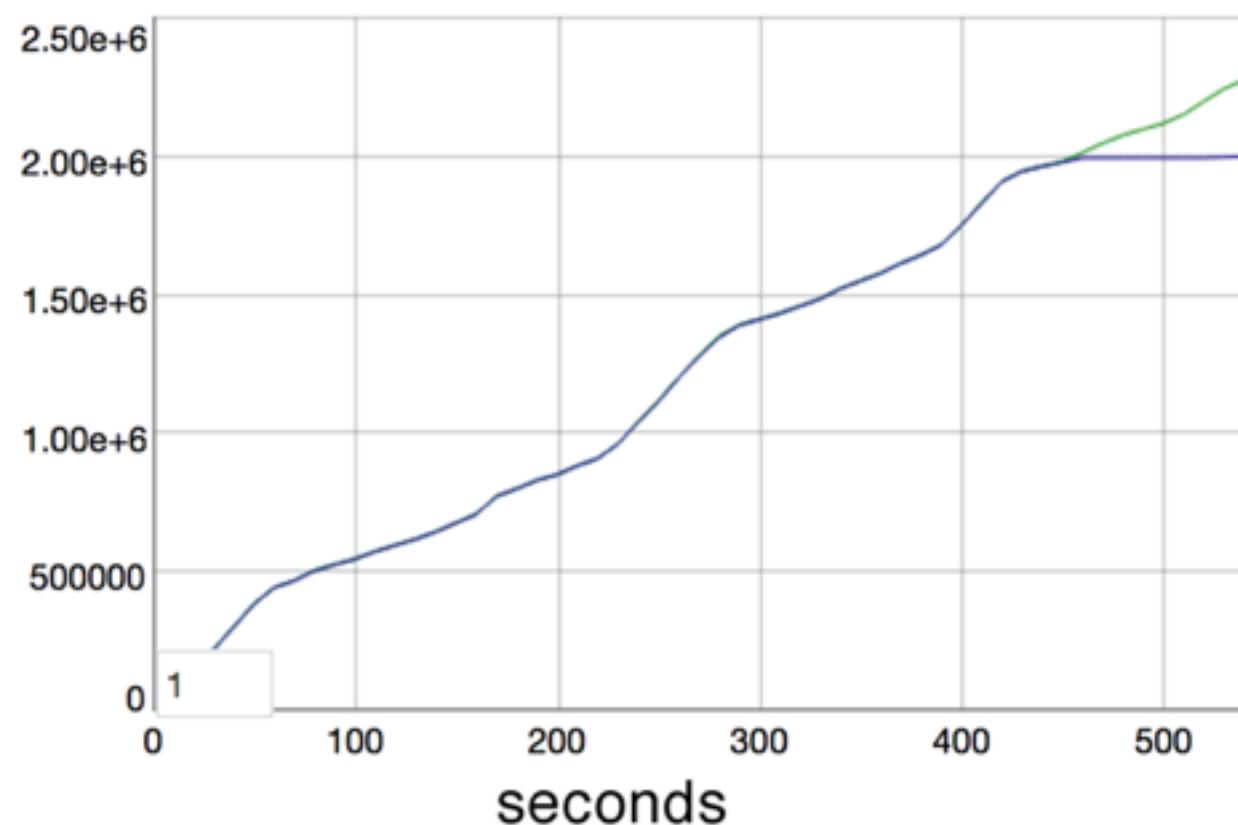


Channels



2 Million Websocket Connections in Phoenix

Simultaneous Users

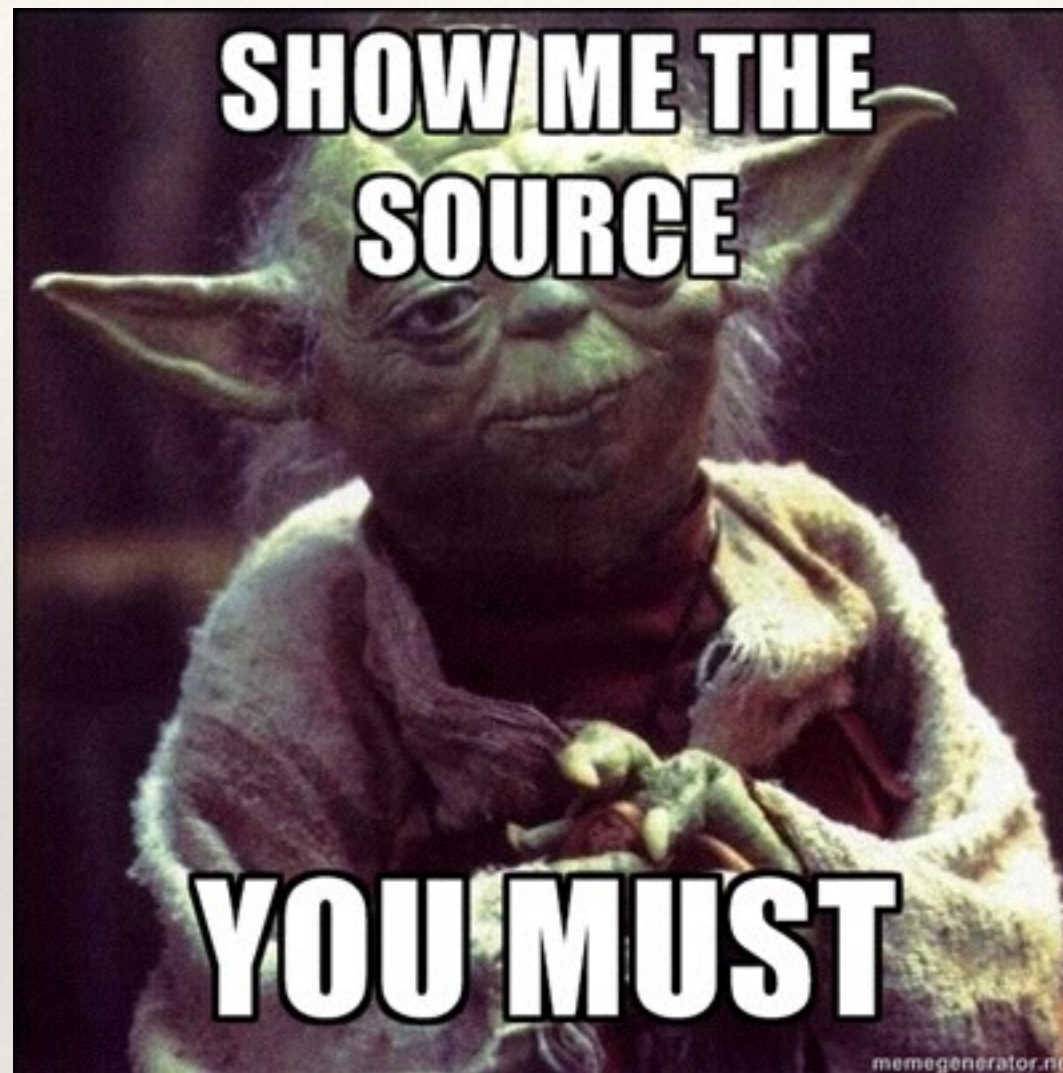


1700045
1763630
1999975
1999984

subscribers

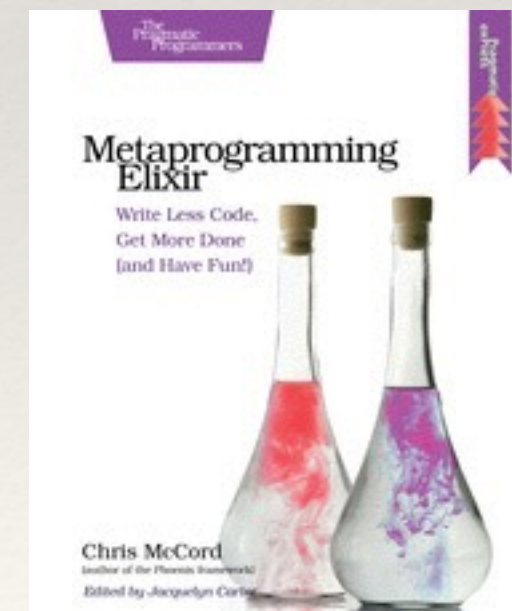
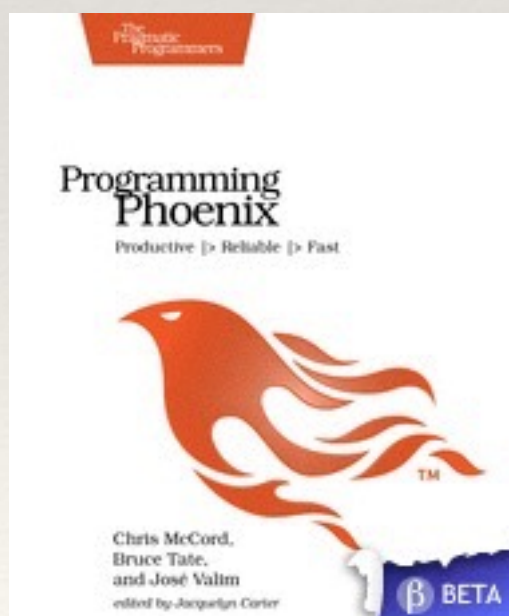
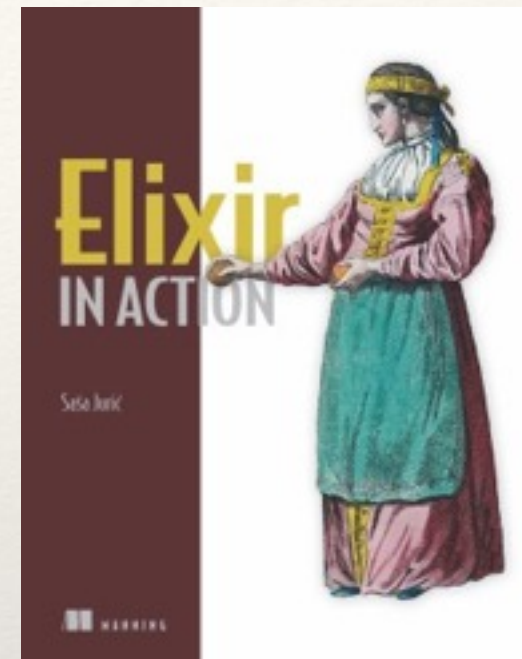
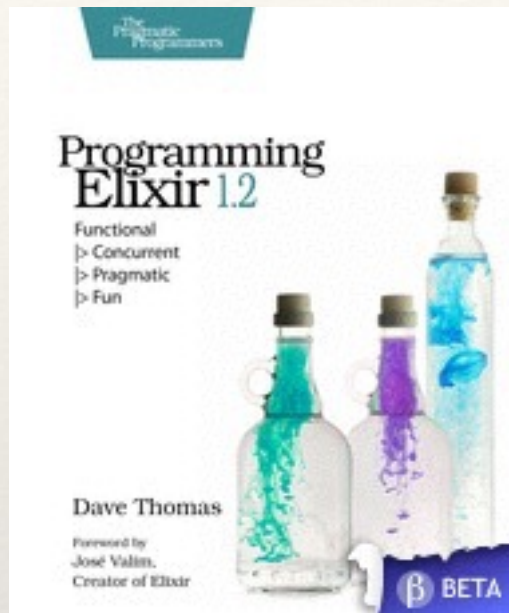
1	[0.0%	11	[0.5%	21	[0.0%	31	[0.0%
2	[0.0%	12	[0.5%	22	[0.0%	32	[0.0%
3	[0.0%	13	[0.0%	23	[0.0%	33	[0.0%
4	[1.0%	14	[0.0%	24	[0.5%	34	[0.0%
5	[0.5%	15	[0.0%	25	[0.0%	35	[0.0%
6	[0.5%	16	[0.0%	26	[0.0%	36	[0.0%
7	[0.0%	17	[0.0%	27	[0.0%	37	[0.0%
8	[1.0%	18	[0.0%	28	[0.5%	38	[0.0%
9	[0.0%	19	[0.0%	29	[0.0%	39	[0.0%
10	[0.0%	20	[0.0%	30	[0.0%	40	[0.0%
Mem	[83765	/	128906	MB	Tasks: 22, 150 thr; 2 running				
Swp	[0	/	0	MB	Load average: 5.98 5.45 3.98				
							Uptime: 5 days, 11:17:13				

<http://www.phoenixframework.org/blog/the-road-to-2-million-websocket-connections>



<https://github.com/slashmili/talks/tree/master/2016/phoenix-klxrb-march>

Books



Thanks

slides and codes

<http://slashmili.org/talks>