Erlang/Elixir

Creating concurrent and parallel systems

Tayllan Búrigo

Summary

- Erlang
- OTP + BEAM
- Elixir
- Concurrency vs. Parallelism
  - Concurrency models
- Erlang processes
- "Let it crash"
- Hot Code Reloading

DISCLAIMER

- Not an Elixir workshop!

Erlang

- Concurrent, functional programming language
- Focus to build massively scalable real-time systems with requirements on high availability
  - Distributed
  - Fault-tolerant
  - Soft real-time (voice call https://stackoverflow.com/a/17309403/5953895)
  - Highly available, non-stop applications (hot swapping code too)
- Developed at Ericsson by Joe Armstrong in 1986

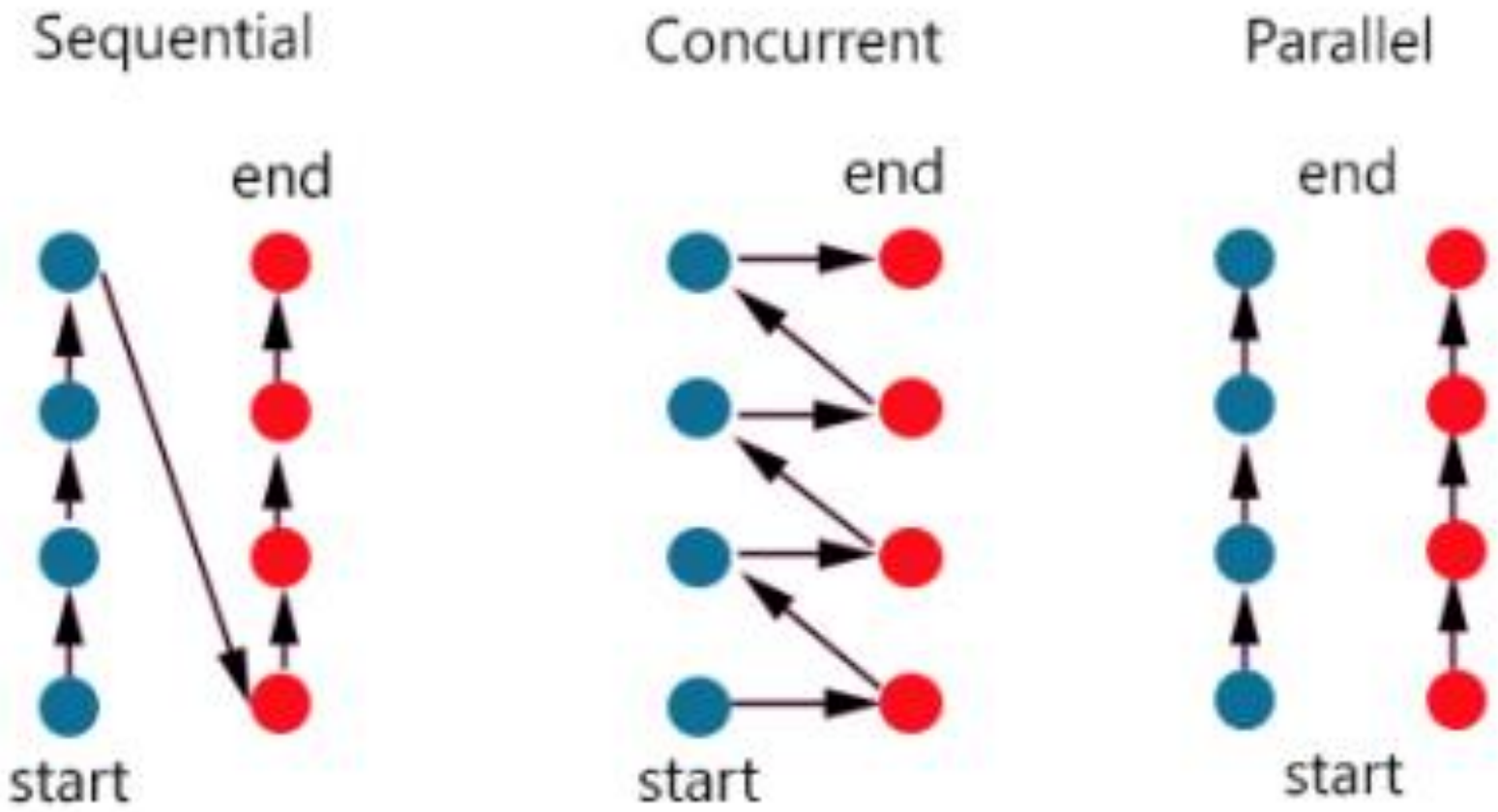
OTP (Open Telecom Platform)

- It WAS about Telecom
- Standard framework + libraries
  - Compiler/interpreter
  - Logger
  - Mnesia distributed database
- BEAM Virtual Machine (akin to Java's JVM)
  - Bogdan's Erlang Abstract Machine
  - Executes bytecode generated by OTP compiler

Elixir

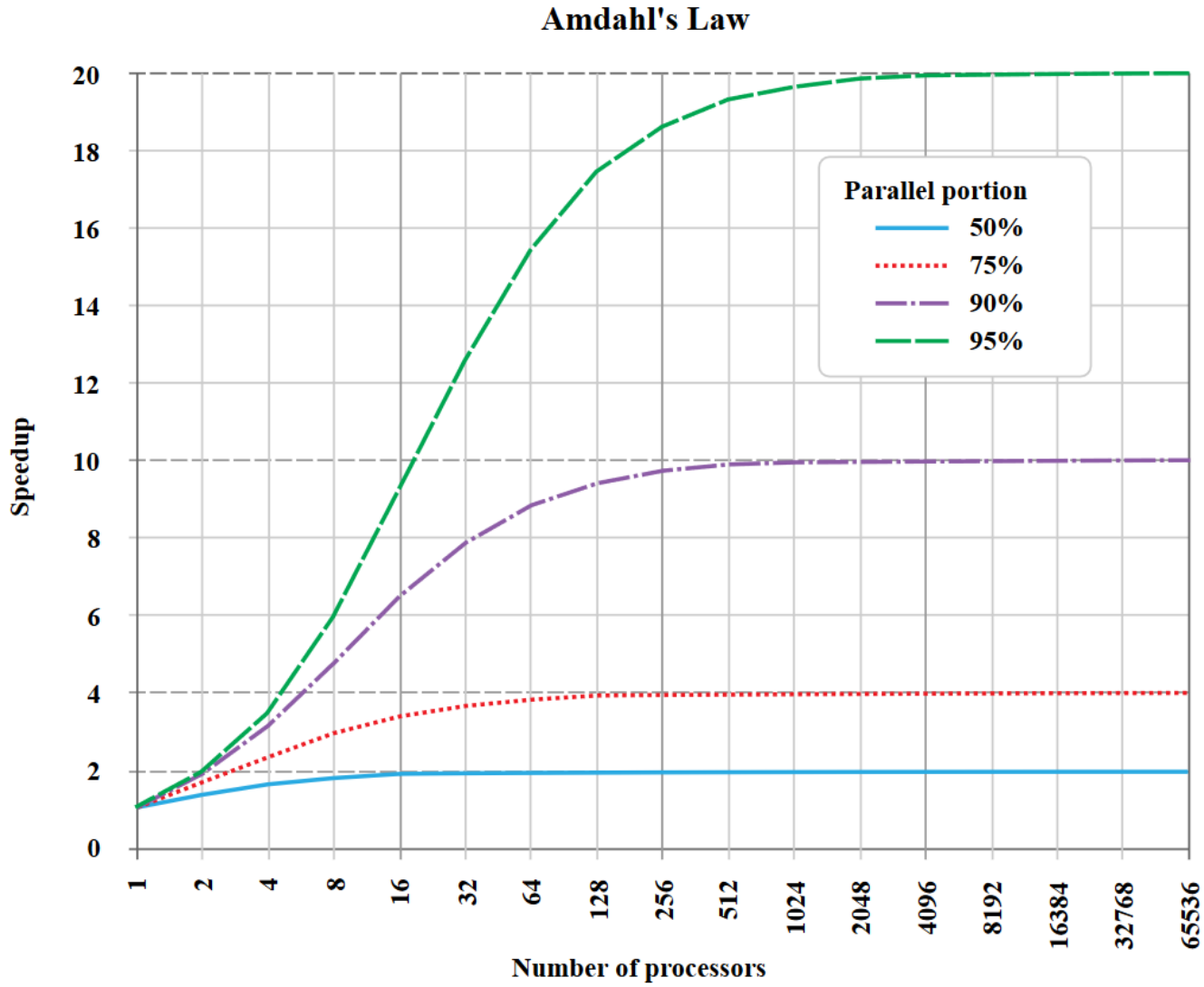
- Same properties as Erlang
- Leverages Erlang's Virtual Machine (BEAM), so it compiles to it
- Developed at Plataformatec by José Valim circa 2011

Concurrency vs. Parallelism



Concurrency vs. Parallelism

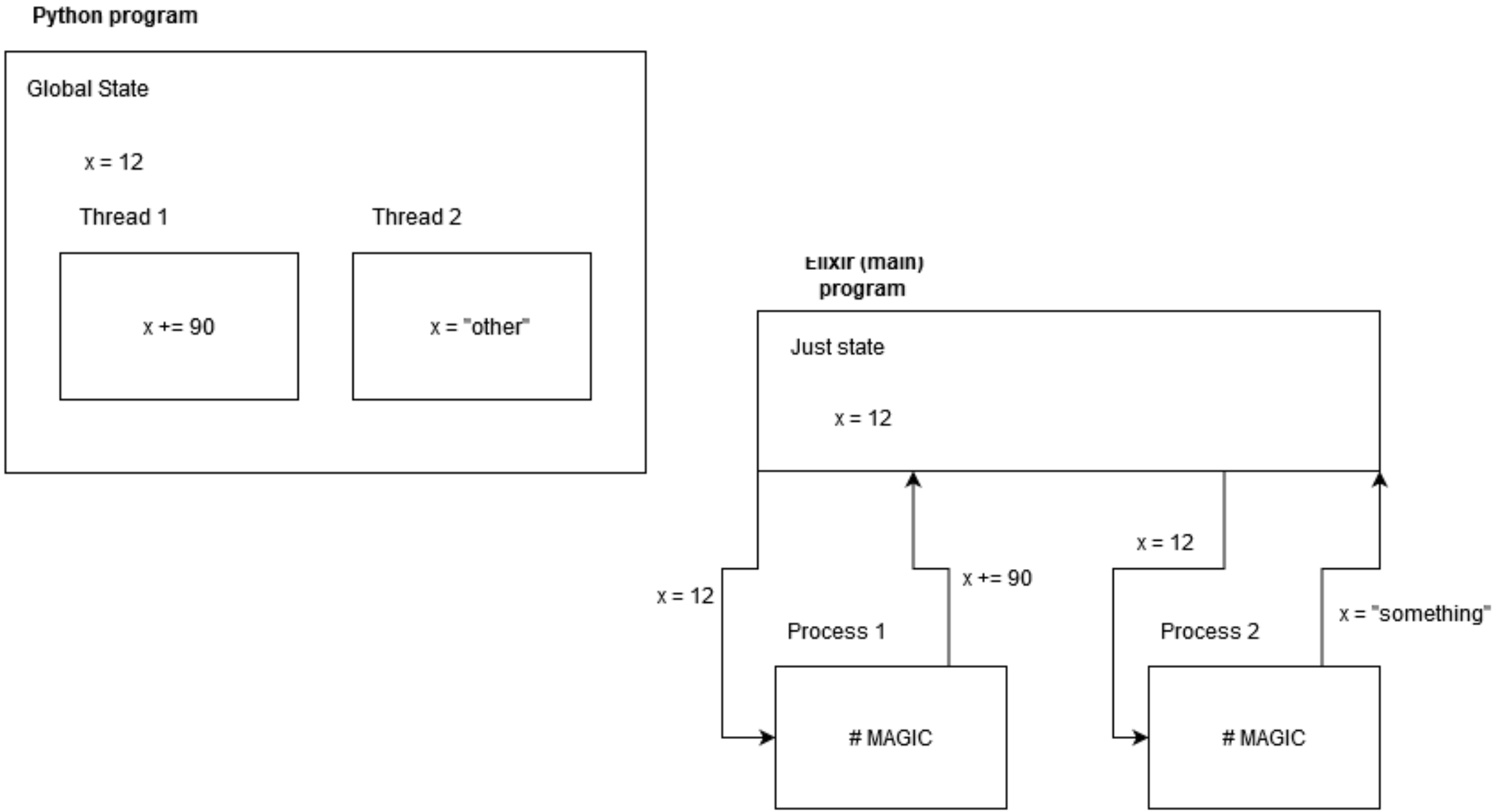
Amdahl's law



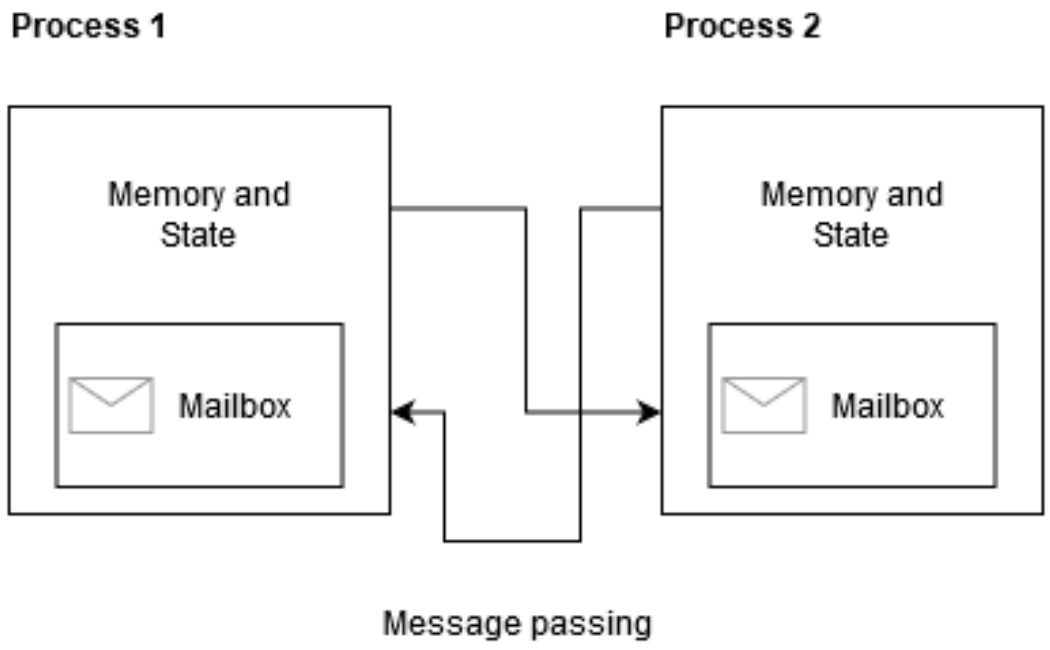
Concurrency Models

- Shared memory communication
  - Shared memory
  - Usually threads
  - Locking mechanism

- Message passing communication (Actor Model)
  - Exchange messages
  - Independent actors
  - No shared context


Python program
Global State
x = 12
Thread 1
Thread 2
x += 90
x = "other"
Elixir (main) program
Just state
x = 12
x = 12
x += 90
x = 12
x = "something"
Process 1
Process 2
MAGIC
MAGIC

Erlang/Elixir Processes



- VERY lightweight
- Fast to create
- Fast to terminate
  - "Let it crash!"

SHOW TIME!

We will:

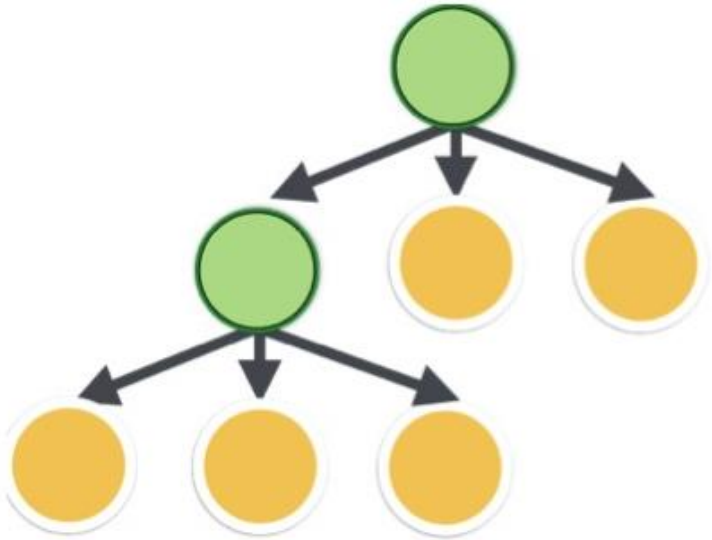
- See PIDs: process identifiers
- Spawn a new process
- Watch the process die
- Communicate with the process
  - Send messages
  - Receive message
- Simulate React's Store/State

"Let it crash"

- Most exceptions don't make sense (think "ConnectException")
- Program the happy path > Defensive programming

"Let it crash"

- Worker
- GenServer
- Supervisor
- Supervisor Tree



Hot Code Reloading

- Allows for hasle-free, quick reload of modules