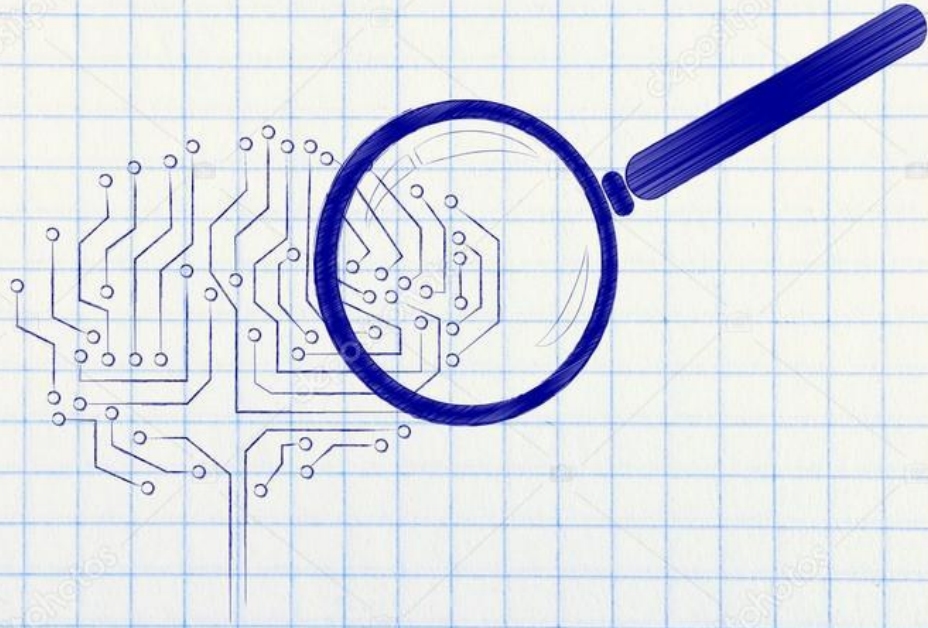


Elixir GenStage

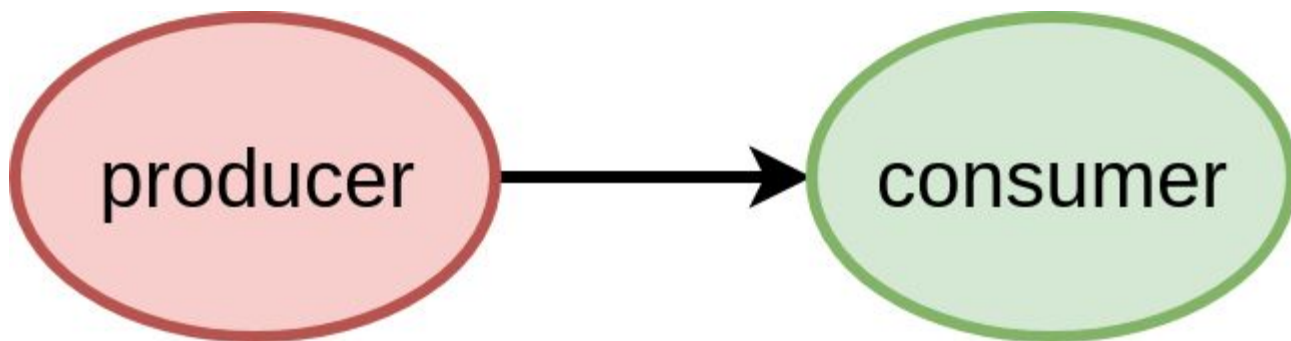
Простая обработка
больших данных

Максим Барулин



WHAT IS DATA PROCESSING?

Простой конвейер



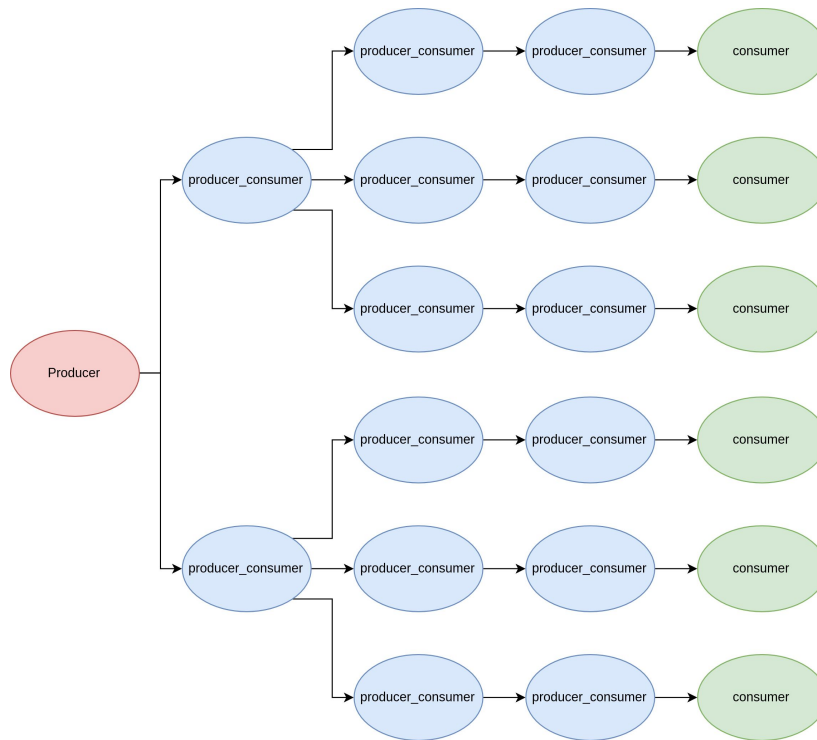
Предварительная подготовка



Сложная обработка



Обработка в нескольких потоках



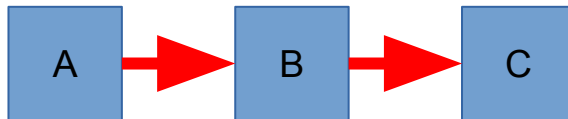
Добавим каплю эликсира



GenStage

Это “спецификация и поток вычислений для Elixir”

GenStage - это возможность определить конвейер задач, выполняемый независимыми шагами (или этапами) в отдельных процессах.



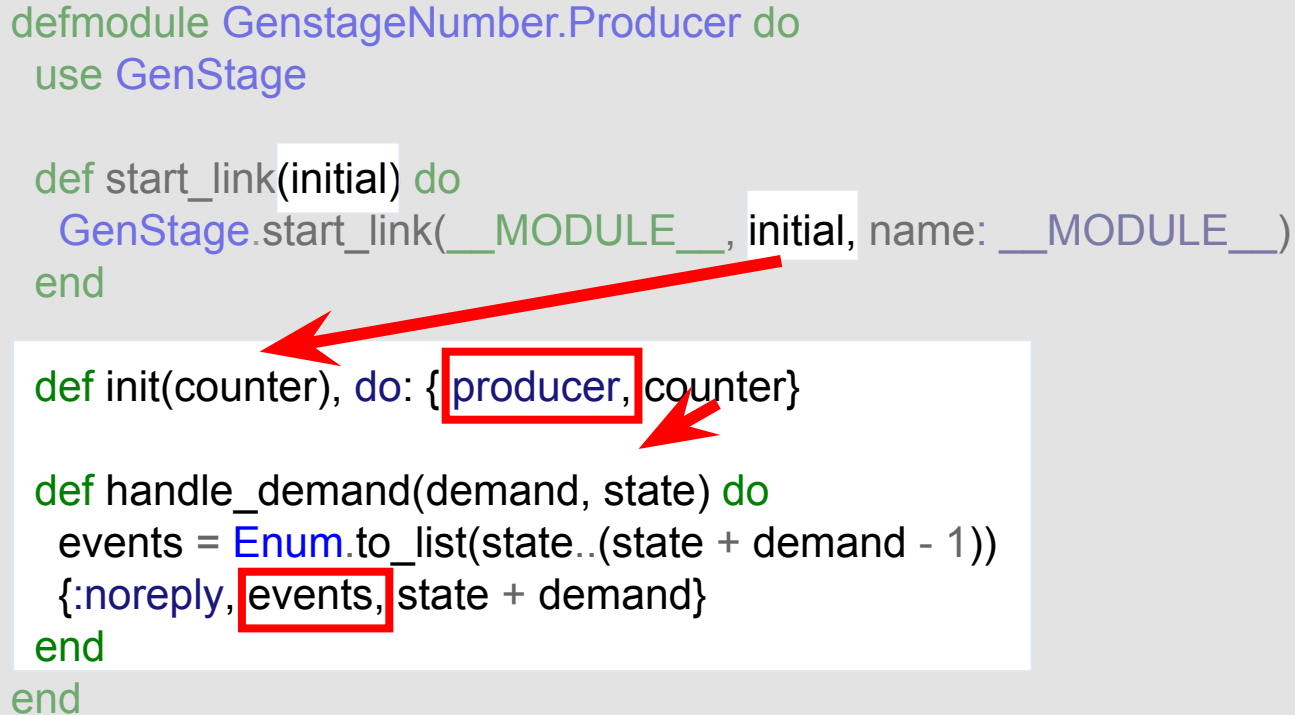
Производитель

```
defmodule GenstageNumber.Producer do
  use GenStage

  def start_link(initial) do
    GenStage.start_link(__MODULE__, initial, name: __MODULE__)
  end

  def init(counter), do: {producer, counter}

  def handle_demand(demand, state) do
    events = Enum.to_list(state..(state + demand - 1))
    {:noreply, events, state + demand}
  end
end
```



Производитель-потребитель

```
defmodule GenstageNumber.ProducerConsumer do
  use GenStage

  require Integer

  def start_link do
    GenStage.start_link(__MODULE__, :state_doesnt_matter, name: __MODULE__)
  end

  def init(state) do
    {:producer_consumer, state, subscribe_to: [GenstageNumber.Producer]}
  end

  def handle_events(events, _from, state) do
    numbers =
      events
      |> Enum.filter(&Integer.is_even/1)

    {:noreply, numbers, state}
  end
end
```

Потребитель

```
defmodule GenstageNumber.Consumer do
  use GenStage

  def start_link do
    GenStage.start_link(__MODULE__, :state_doesnt_matter)
  end

  def init(state) do
    {:consumer, state, subscribe_to: [GenstageNumber.ProducerConsumer]}
  end

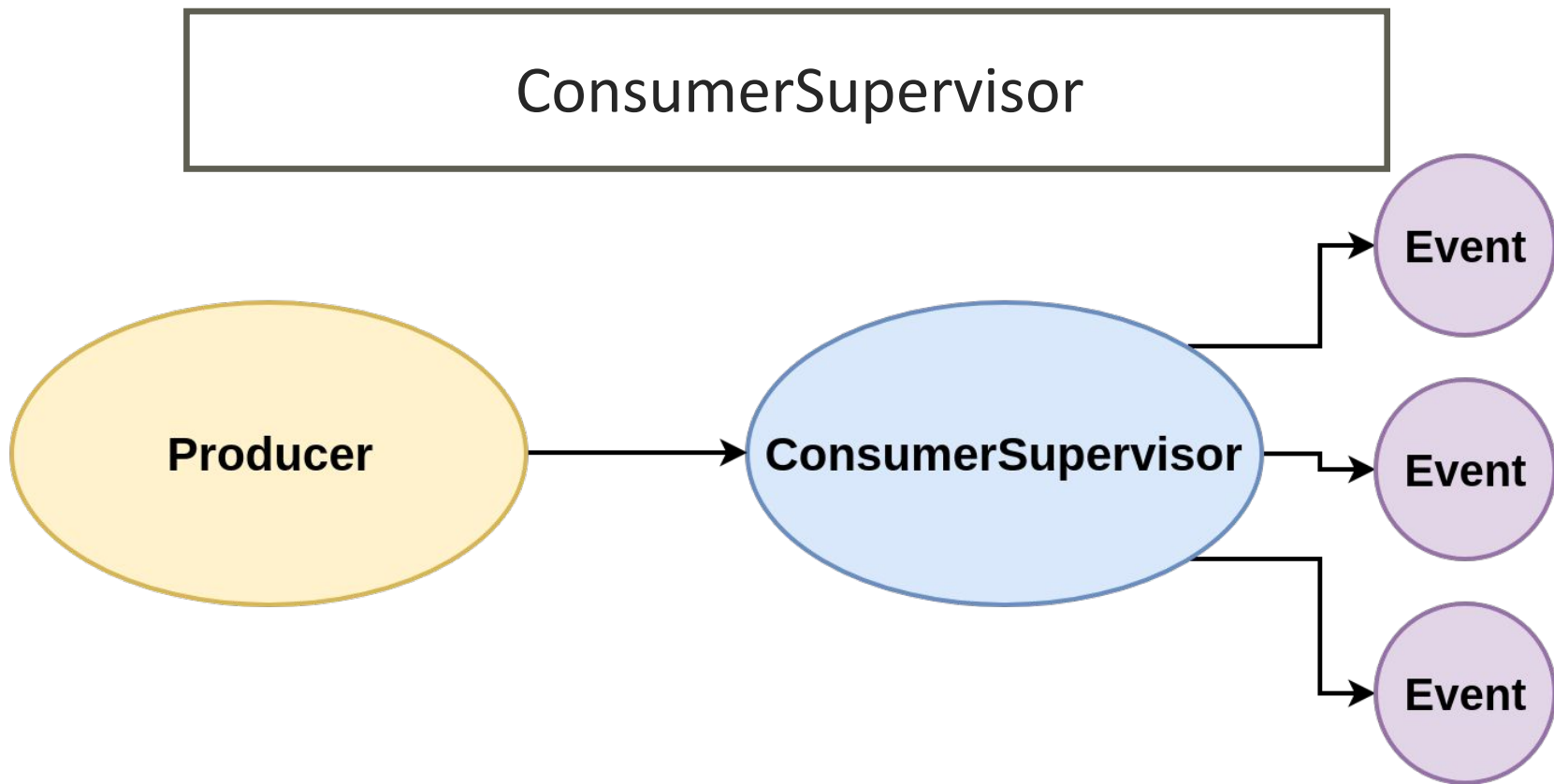
  def handle_events(events, _from, state) do
    for event <- events do
      IO.inspect({self(), event})
    end

    {:noreply, [], state}
  end
end
```

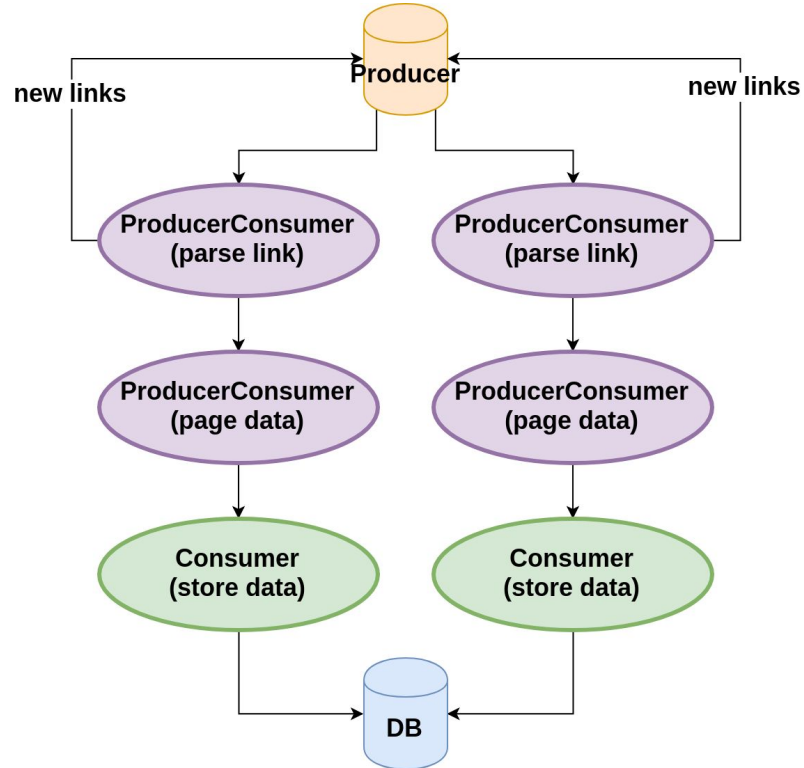
Запускаем

```
def start(_type, _args) do
  children = [
    {GenstageNumber.Producer, [0]},
    {GenstageNumber.ProducerConsumer, []},
    {GenstageNumber.Consumer, []}
  ]

  opts = [strategy: :one_for_one, name:
GenstageNumber.Supervisor]
  Supervisor.start_link(children, opts)
end
```



Пример





https://github.com/Slavenin/gen_stage_example

Learn You Some Elixir for great good!