Erlang: conceitos e curiosidades

Linguagem funcional, concorrente e tolerante a falhas.

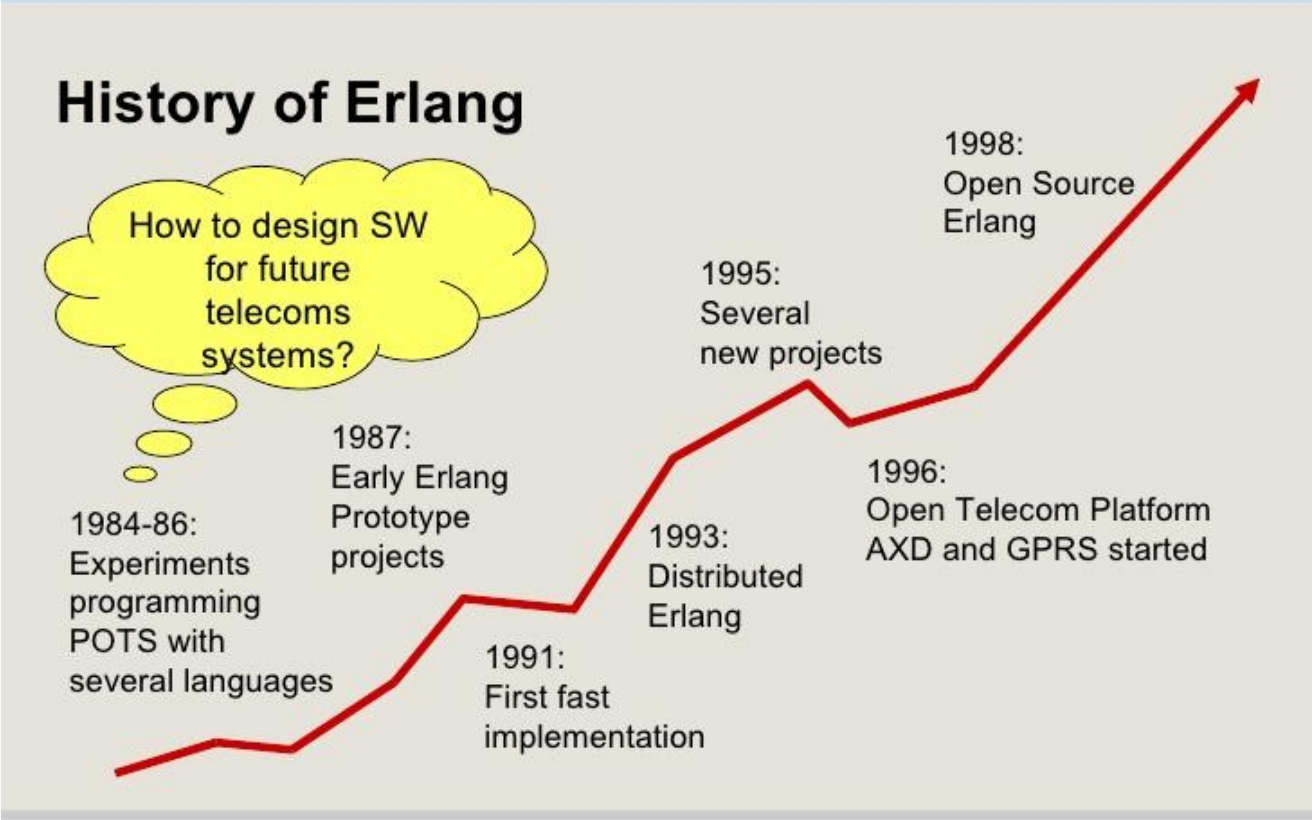
FLISOL - 2016 Isabella Fontes

Start

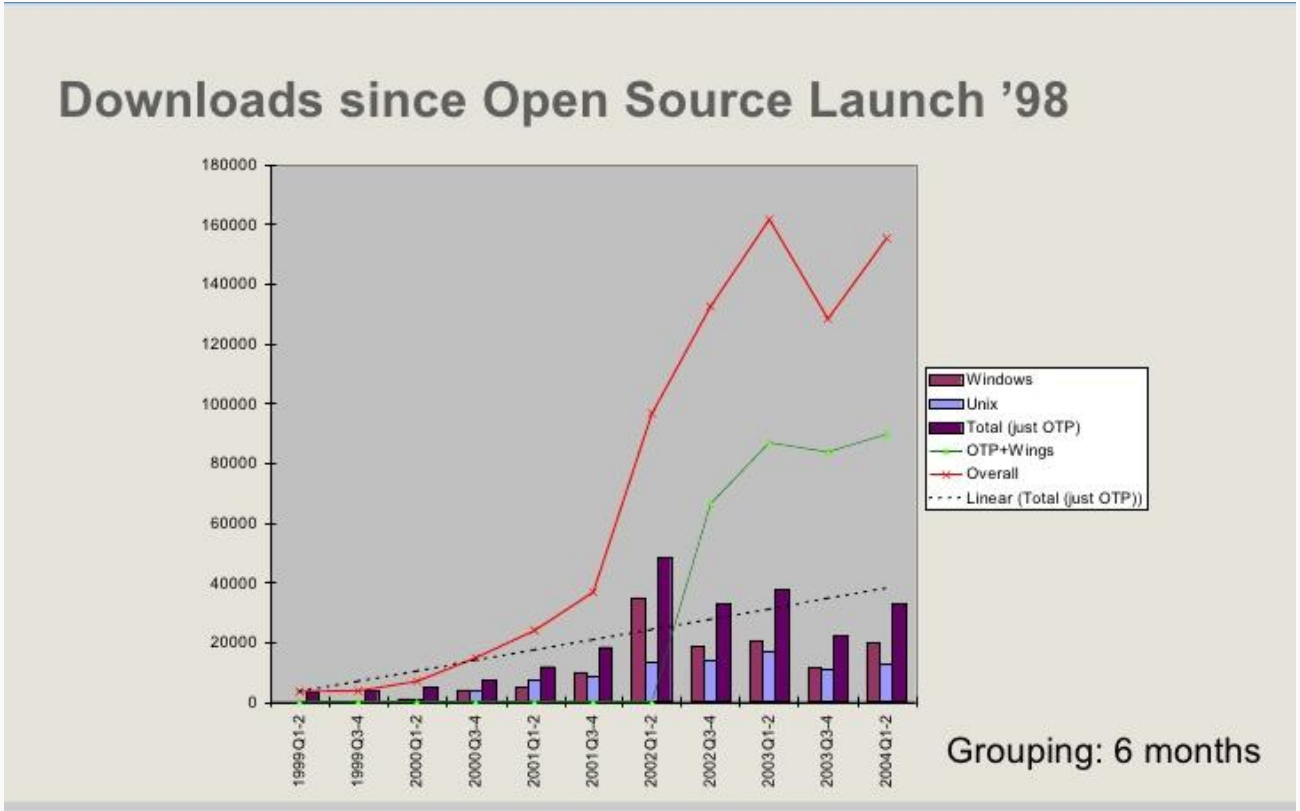
- Erlang nasceu no laboratório de ciência da computação da Ericsson ~1980
- O foco era sistemas para telecom
- Joe Armstrong considerado criador
- Influenciada por linguagens como ML, Ada, Module, Prolog e Smalltalk
- Em 1998 a Ericsson tornou Erlang open source sob a licença EPL.

“ERicsson **LANG**uage”*

*joke


History of Erlang
How to design SW for future telecoms systems?
1984-86: Experiments programming POTS with several languages
1987: Early Erlang Prototype projects
1991: First fast implementation
1993: Distributed Erlang
1995: Several new projects
1996: Open Telecom Platform AXD and GPRS started
1998: Open Source Erlang
Rev PA1
2003-12-01
3

Start Downloads since Open Source Launch ‘98

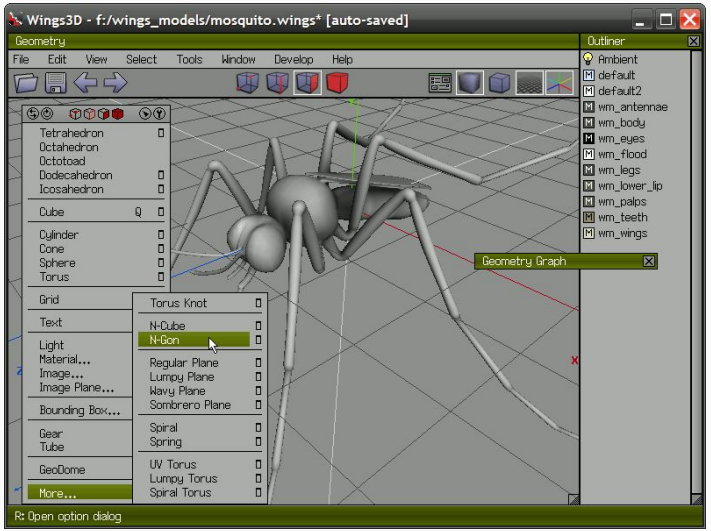


Companies use Open source Erlang Applications

- **RabbitMQ**: AMQP messaging protocol implementation. AMQP is an emerging standard for high-performance enterprise messaging;
- **CouchDB**: “schema-less” document-oriented database, providing scalability across multicore and multiserver clusters;
- **Ejabberd:** system provides an Extensible Messaging and Presence Protocol (XMPP) based instant messaging (IM) application server.

Companies use Open source Erlang Applications

- **Wings 3D**: a 3D modeller based on Nendo;
- **Yaws**: “Yet Another Web Server”.
- **EDDIE**: Distributed TCP/IP based Clusterware.



WINGS 3D

powered by Yaws webserver

eddie

Companies use Erlang Applications

- **Facebook**, no backend de seu sistema de chat, lidando com ~100 milhõs de usuários ativos;
- **Del.icio.us (Yahoo!)**, ~ 5 milhões de usuários e ~150 milhões de bookmarks;
- **Amazon SimpleDB**, o serviço de dados Amazon EC2;
- **GitHub**, sistema de backend, lidando com milhares de transações concorrentes;
- **Twitter,** microbloging.

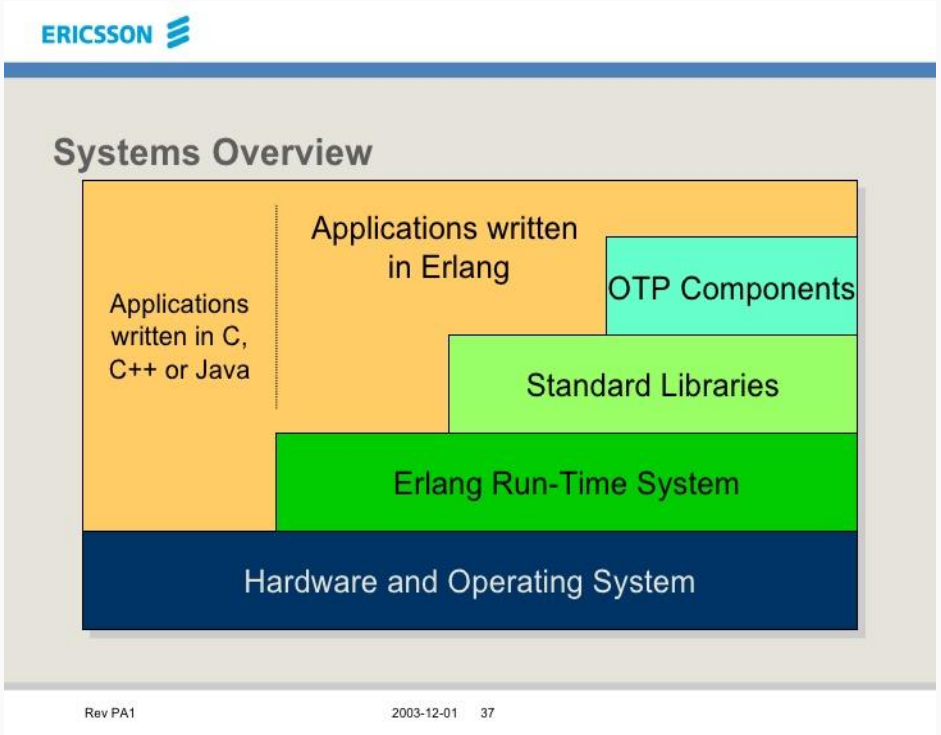
Erlang The Erlang Language

- Escrita de código simples e objetivo;
- Organização em módulos reutilizáveis;
- Tolerância a falhas, concorrência realmente pesada, computação distribuída, atualização da aplicação sem derrubá-la, sistemas de tempo real, este é o nicho de Erlang, foi para isto que Erlang nasceu;
- Escalabilidade;
- Erlang é conhecido por conseguir 9x9s de confiabilidade (99.9999999% uptime, portanto, menos do que 31,536 milissegundos de tempo de inatividade por ano), com sistemas de produção real (dentro da indústria de telecomunicações). [1]

[1] Joe ArmStrong (http://ll2.ai.mit.edu/talks/armstrong.pdf)

Erlang System Overview

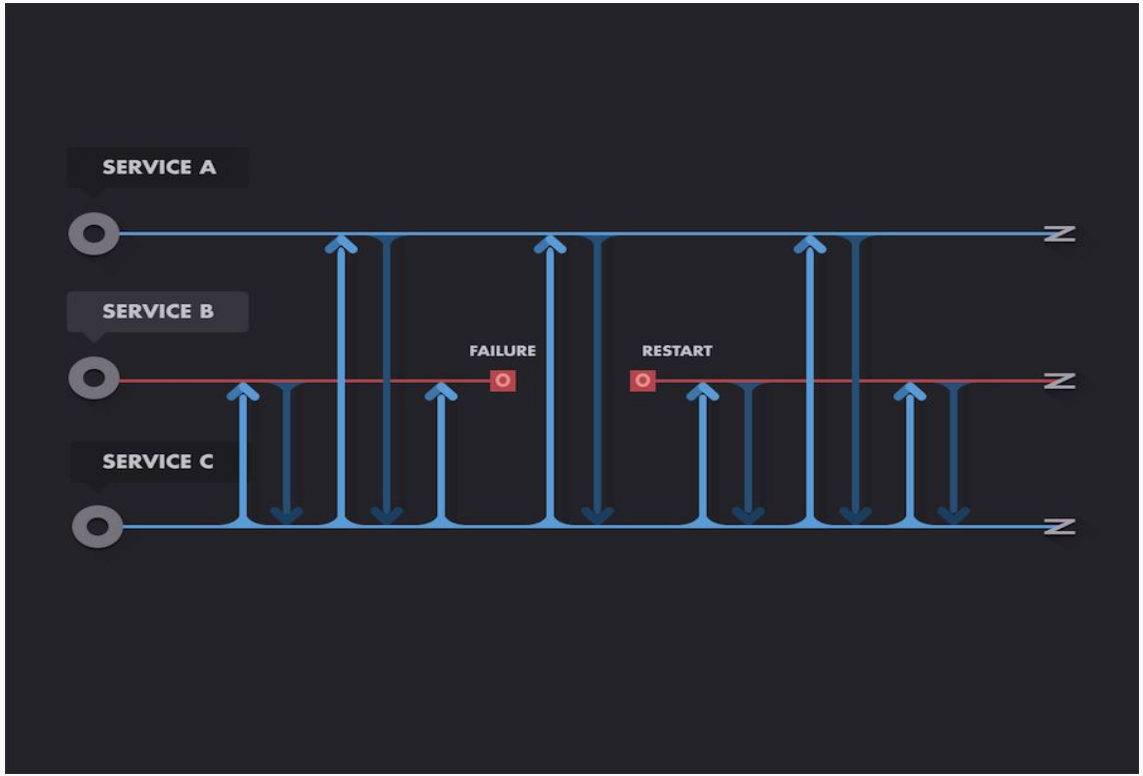
- OTP: Open Telecom Platform
  - Middleware for Erlang development
  - Modules and standards designed to help you build applications.
  - Behaviors
  - Components:
    - Error handling, Reporting and logging
    - CORBA, Java & C Support
    - HTTP Server + Client
    - FTP Client
    - XML
    - etc



Erlang Fault-tolerance

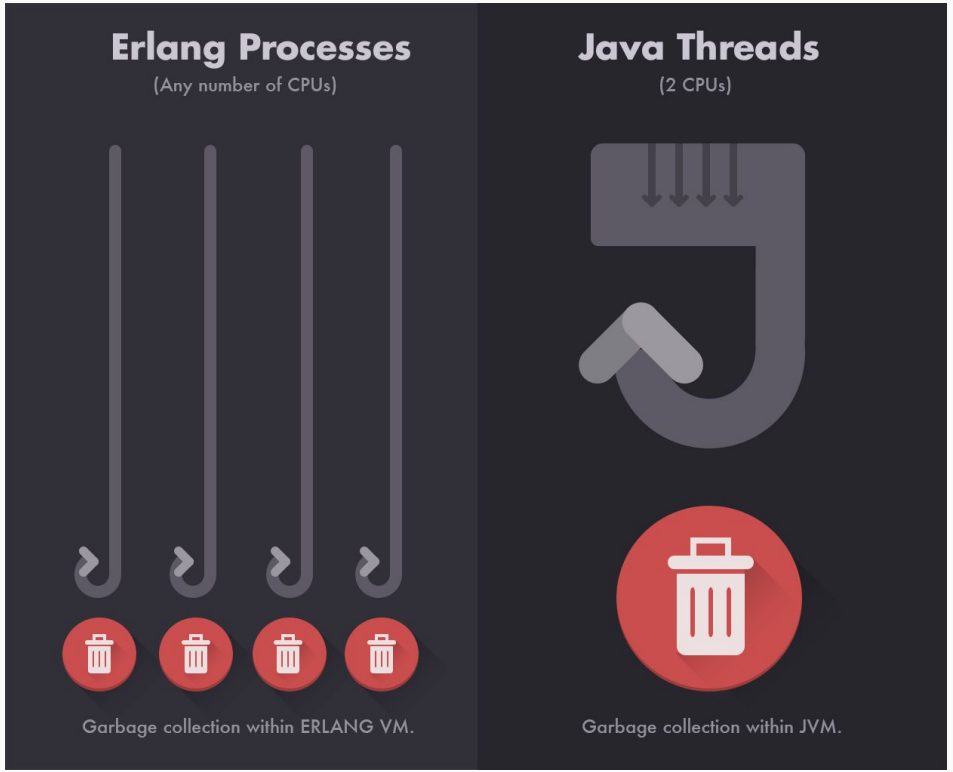
Tolerancia a falha em Cloud Computing

- Robustez a erros - Continuar funcionando mesmo com falhas;
- Ex: Serviço C envia pedidos aos serviços A e B. Apesar de falhas de serviço B, temporariamente, o resto do sistema continua, relativamente desimpedido.



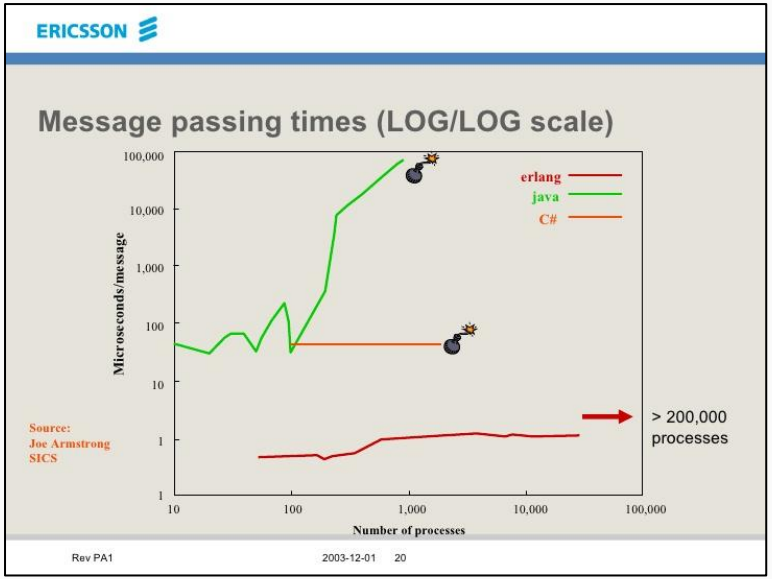
Erlang Garbage Collection

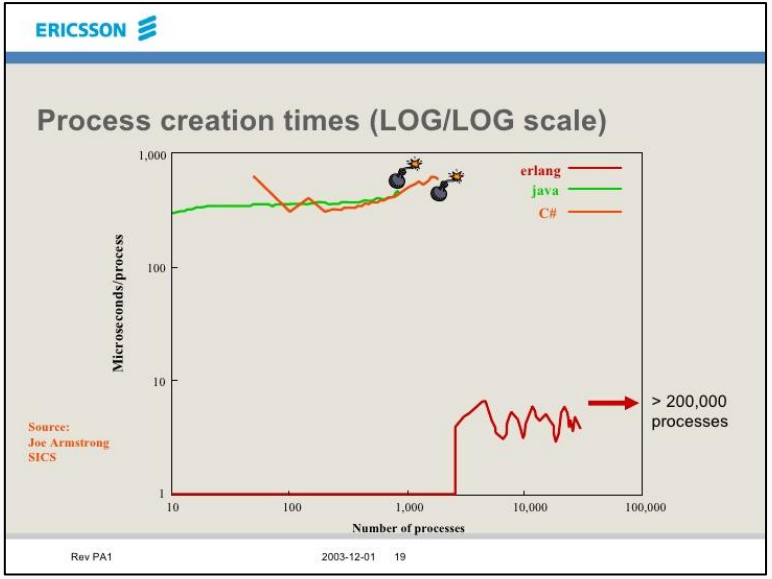
- Garbage Collection



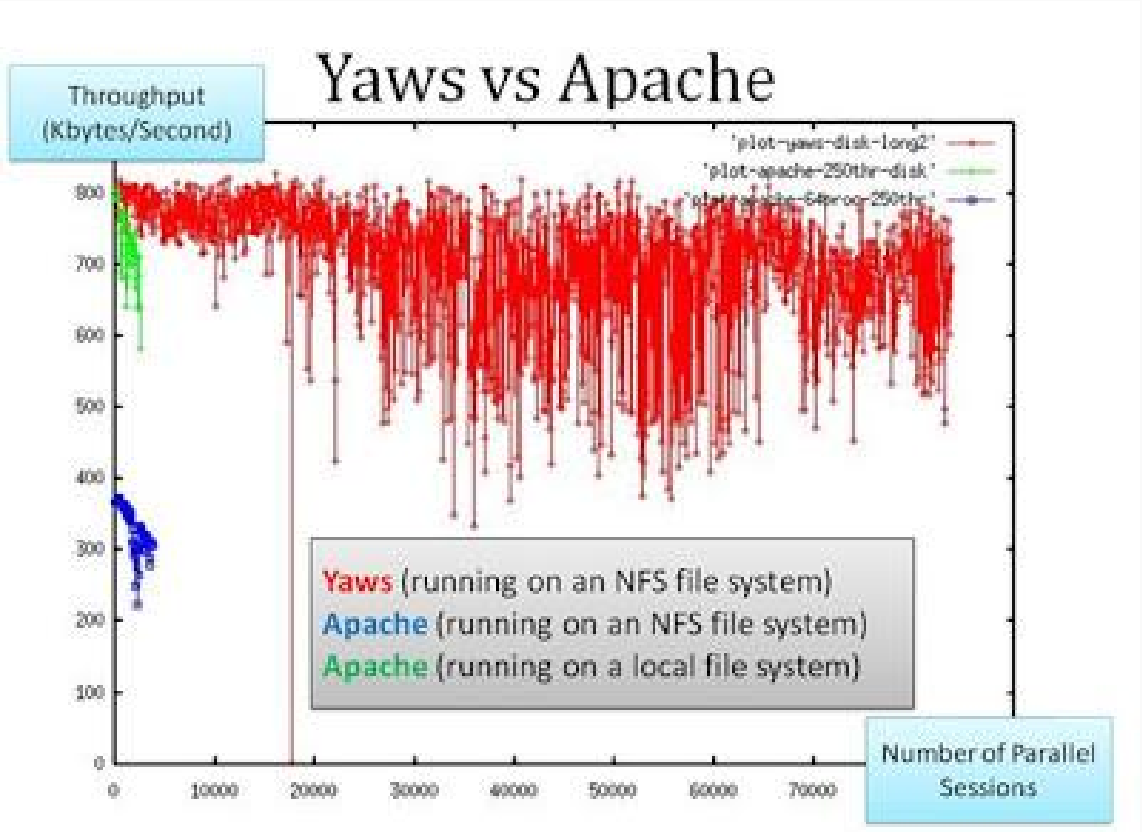
Erlang Concurrency Oriented Programming

- Erlang vs Java vs C#

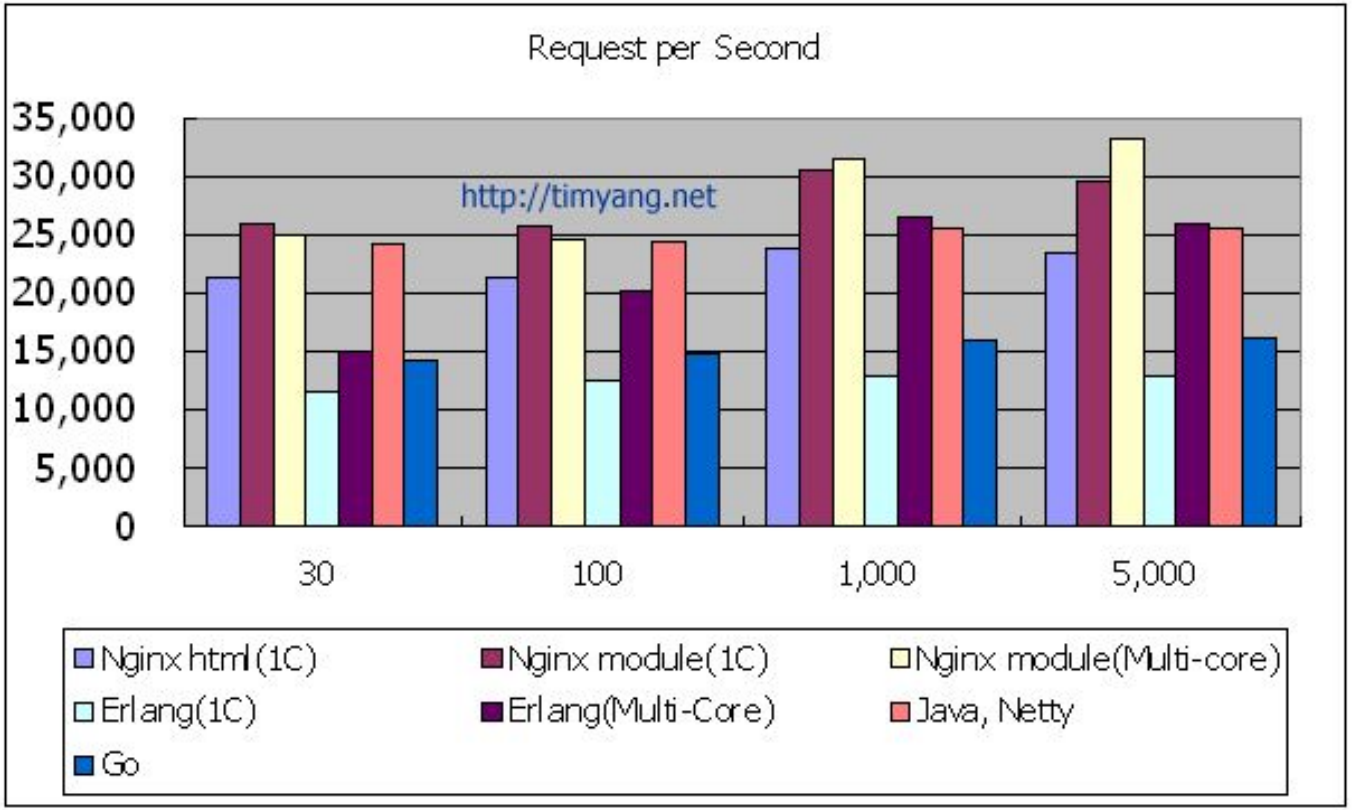




Erlang Yaws vs Apache



Erlang


Request per Second
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
http://timyang.net
30
100
1,000
5,000
Nginx html(1C)
Nginx module(1C)
Nginx module(Multi-core)
Erlang(1C)
Erlang(Multi-Core)
Java, Netty
Go

Erlang Data Types Example

```
43> io:format("Bem vindos ao FLISOL 2016").
Bem vindos ao FLISOL 2016ok
44> io:format("Bem vindos ao FLISOL 2016~n").
Bem vindos ao FLISOL 2016
ok
45> io:format("Bem vindos ~p.~n", ["ao FLISOL 2016~n"]).
Bem vindos "ao FLISOL 2016~n".
ok
46> [First|Rest] = [1,2,3,4,<<"binary">>,4.5, atom@].
[1,2,3,4,<<"binary">>,4.5,atom@]
47> First.
1
48> Rest.
[2,3,4,<<"binary">>,4.5,atom@]
49> [One|Other] = Rest.
[2,3,4,<<"binary">>,4.5,atom@]
50> One.
2
51> Other.
[3,4,<<"binary">>,4.5,atom@]
52> 5+6.
11
53> 2*a.
** exception error: an error occurred when evaluating an arithmetic expression
     in operator  */2
        called as 2 * a
54> 2*error.
** exception error: an error occurred when evaluating an arithmetic expression
     in operator  */2
        called as 2 * error
```

Erlang Hello Word program

color.erl | tut.erl | Settings

```
1  -module(tut).
2  -export([double/1, fac/1, convert_length/1,
3      list_max/1, reverse/1, test_if/2, month_length/2,
4          start/0, say_something/2]).
5
6  double(X) ->
7      2 * X.
8
9  fac(1) ->
10     1;
11 fac(N) ->
12     N * fac(N - 1).
13
14 convert_length({centimeter, X}) ->
15     {inch, X / 2.54};
16 convert_length({inch, Y}) ->
17     {centimeter, Y * 2.54}.
18
19 list_max([Head|Rest]) ->
20     list_max(Rest, Head).
21 list_max([], Res) ->
22     Res;
23 list_max([Head|Rest], Result_so_far) when Head > Result_so_far ->
24     list_max(Rest, Head);
25 list_max([Head|Rest], Result_so_far)  ->
26     list_max(Rest, Result_so_far).
27 reverse(List) ->
28
29     reverse(List, []).
30
31 reverse([Head | Rest], Reversed_List) ->
32     reverse(Rest, [Head | Reversed_List]);
33 reverse([], Reversed_List) ->
```

/home/isabella/Desktop/tut.erl* 1:1

color.erl | tut.erl | Settings

```
42             not_leap;
43         trunc(Year / 4) * 4 == Year ->
44             leap;
45         true ->
46             not_leap
47     end,
48     case Month of
49         sep -> 30;
50         apr -> 30;
51         jun -> 30;
52         nov -> 30;
53         feb when Leap == leap -> 29;
54         feb -> 28;
55         jan -> 31;
56         mar -> 31;
57         may -> 31;
58         jul -> 31;
59         aug -> 31;
60         oct -> 31;
61         dec -> 31
62     end.
63
64 say_something(What, 0) ->
65     done;
66 say_something(What, Times) ->
67     io:format("~p~n", [What]),
68     say_something(What, Times - 1).
69
70 start() ->
71     spawn(tut, say_something, [<<"flisol 2016 message 1">>, 5]),
72     spawn(tut, say_something, [<<"flisol 2016 message 2">>, 5]),
73     spawn(tut, say_something, [<<"flisol 2016 message 3">>, 5]).
74
```

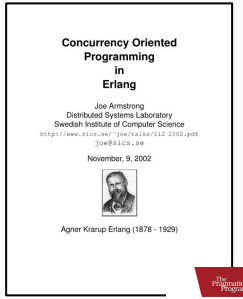
/home/isabella/Desktop/tut.erl* 74:1

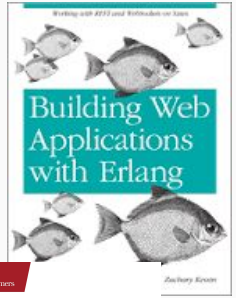
Melhores Indicações

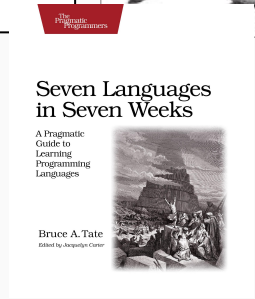
- **Seven Languages in Seven Weeks**
  A Pragmatic Guide to Learning Programming Languages
  by Bruce A. Tate

- **Building Web Applications with Erlang**
  Working with REST and Web Sockets on Yaws
  by Zachary Kessin

- **Concurrency Oriented Programming in Erlang**
  Distributed Systems Laboratory Swedish Institute of Computer Science
  By Joe Armstrong







Indicações

http://grokpodcast.com/2011/07/21/episodio-39-erlang-parte-1-de-3/

http://grokpodcast.com/2011/07/21/episodio-40-erlang-parte-2-de-3/

http://grokpodcast.com/2011/07/21/episodio-41-erlang-parte-3-de-3/

Indicações

http://learnyousomeerlang.com/content

http://erlang.org/doc/getting_started/users_guide.html

https://www.toptal.com/erlang/a-cloud-at-the-lowest-level-built-in-erlang

https://www.process-one.net/en/ejabberd/

Obrigada a todos pela atenção :)

FLISOL 2016
Pederneiras, São Paulo
Brasil