



Erlang Training and Consulting Ltd

# Thinking in a Highly Concurrent, Mostly-functional Language

QCON London, March 12<sup>th</sup>, 2009

**Francesco Cesarini**  
francesco@erlang-consulting.com

```
counter_loop(Count) ->  
  receive  
    increment ->  
      counter_loop(Count + 1);  
    {count, To} ->  
      To ! {count, Count},  
      counter_loop(Count)  
  end.
```

# Erlang



**After you've opened the top of your head, reached in and turned your brain inside out, this starts to look like a natural way to count integers. And Erlang does require some fairly serious mental readjustment.**

**However... having spent some time playing with this, I tell you...**

Tim Bray, Director of Web Technologies - Sun Microsystems

Copyright 2008 - Erlang Training and Consulting Ltd



**... If somebody came to me and wanted to pay me a lot of money to build a large scale message handling system that really had to be up all the time, could never afford to go down for years at the time, I would unhesitatingly choose Erlang to build it in.**

Tim Bray, Director of Web Technologies - Sun Microsystems

Copyright 2008 - Erlang Training and Consulting Ltd



# Syntax

Copyright 2008 - Erlang Training and Consulting Ltd



# Concurrency

Copyright 2008 - Erlang Training and Consulting Ltd



## Products: AXD301 Switch - 1996

A Telephony-Class, scalable (10 - 160 GBps) ATM switch

Designed from scratch in less than 3 years

### AXD 301 Success factors:

- Competent organisation and people
- Efficient process
- Excellent technology (e.g. Erlang/OTP)



Copyright 2008 - Erlang Training and Consulting Ltd

*Erlang*

## Products: AXD301 Switch - 1996

### Erlang: ca 1.5 million lines of code

- Nearly all the complex control logic
- Operation & Maintenance
- Web server and runtime HTML/JavaScript generation

### C/C++: ca 500k lines of code

- Third party software
- Low-level protocol drivers
- Device drivers

### Java: ca 13k lines of code

- Operator GUI applets



Copyright 2008 - Erlang Training and Consulting Ltd

*Erlang*

## Concurrency Modeling

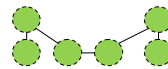
Model for the natural concurrency in your problem

In the old days, processes were a critical resource

- Rationing processes led to complex and unmanageable code

Nowadays, processes are very cheap: if you need a process - create one!

Example: AXD301 process model



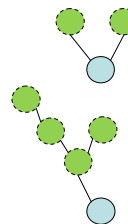
1<sup>st</sup> prototype:  
6 processes/call



2 processes/call



1 process/all calls



2 processes/  
call transaction

4-5 processes/  
call transaction

Copyright 2008 - Erlang Training and Consulting Ltd

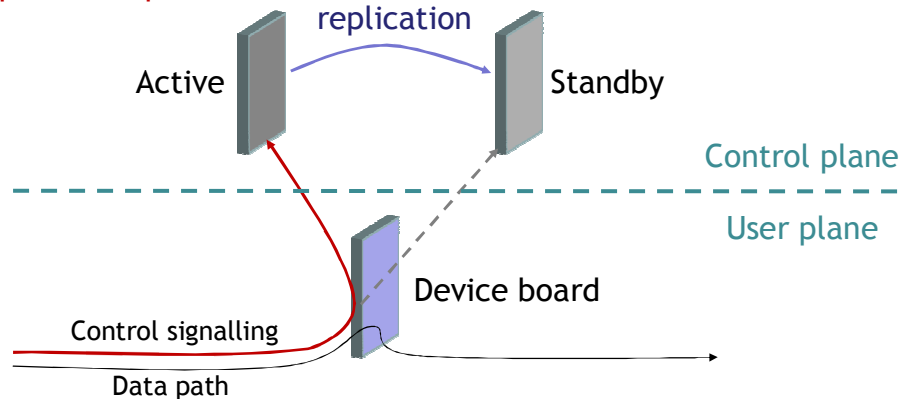
*Erlang*

## 1+1 Redundancy - Good ol' Telecoms

~ 35 000 calls  
per processor pair

Stable-state  
replication

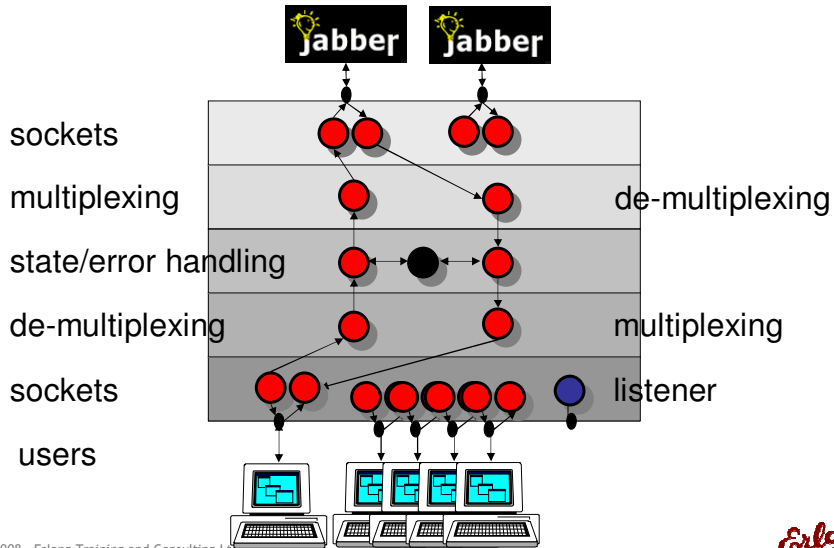
No ongoing sessions  
lost at "failover"



Copyright 2008 - Erlang Training and Consulting Ltd

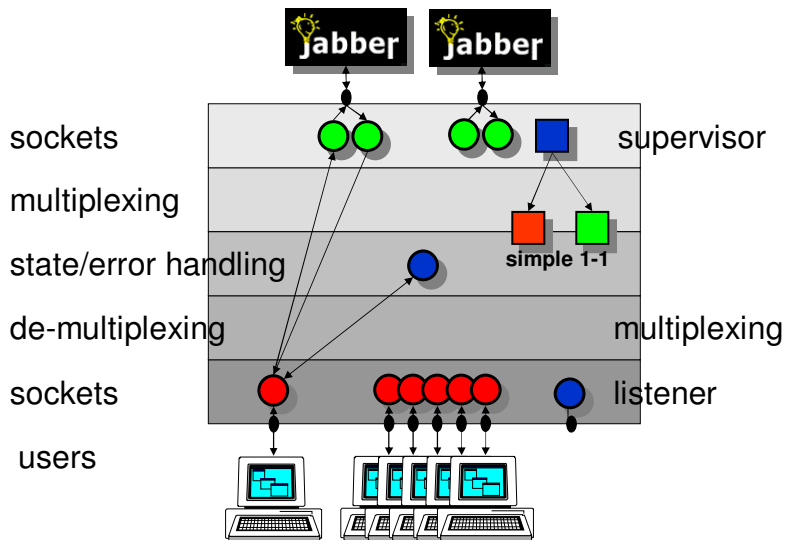
*Erlang*

## First IM Proxy Prototype - 2000



*Erlang*

## First IM Proxy Prototype - 2000



*Erlang*

## Products: EjabberD IM Server - 2002

A distributed XMPP server

Started as an Open Source  
Project by *Alexey Shchepin*

Commercially Supported by  
Process-One (Paris)

- 40% of the XMPP IM market
- Used as a transport layer
- Manages 30,000 users / node

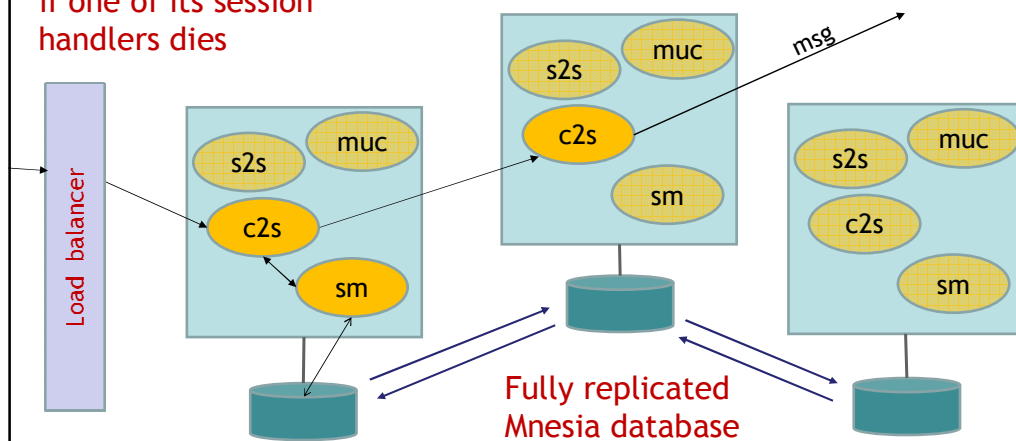


Copyright 2008 - Erlang Training and Consulting Ltd



## Fully Replicated Cluster - Ejabberd 2002

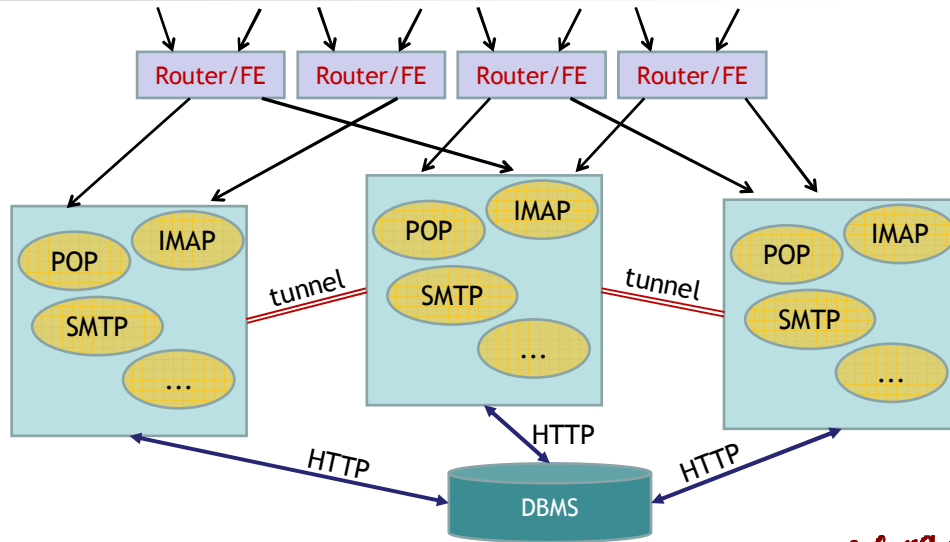
Client must re-connect  
if one of its session  
handlers dies



Copyright 2008 - Erlang Training and Consulting Ltd

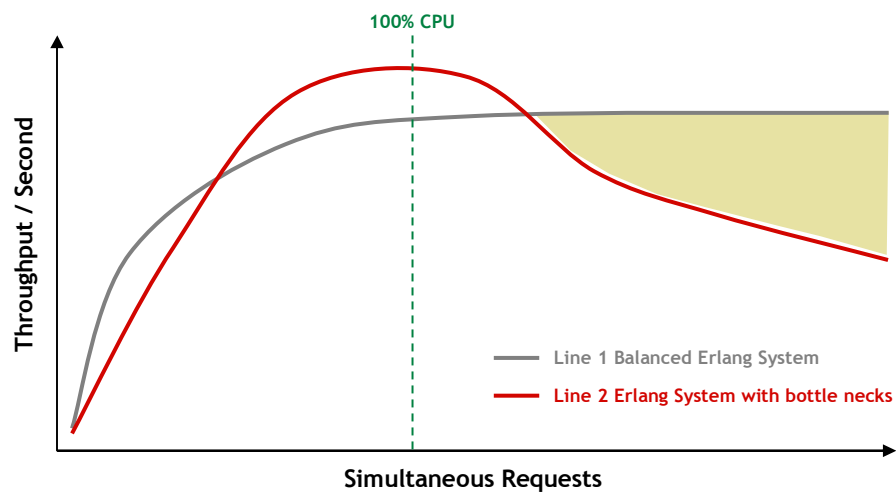


## Share-nothing Architecture - Messaging Gateway



Copyright 2008 - Erlang Training and Consulting Ltd

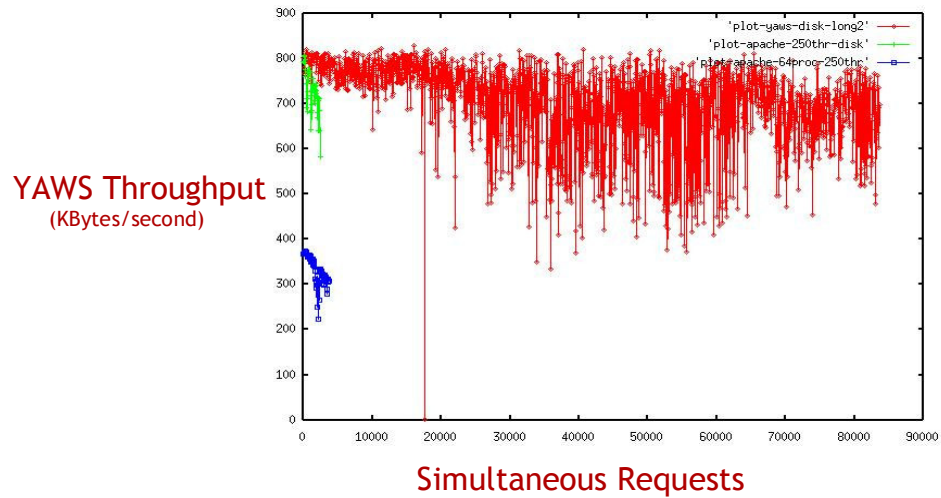
## Erlang Concurrency Under Stress - Pre-SMP



Copyright 2008 - Erlang Training and Consulting Ltd



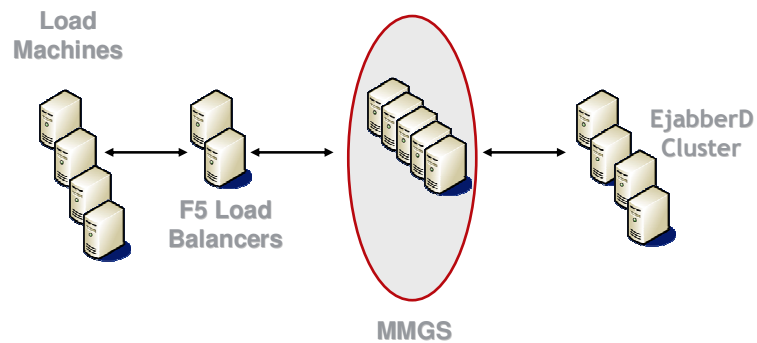
## Erlang Concurrency Under Stress - Pre-SMP



Copyright 2008 - Erlang Training and Consulting Ltd

*Erlang*

## Erlang Concurrency Under Stress - Post-SMP

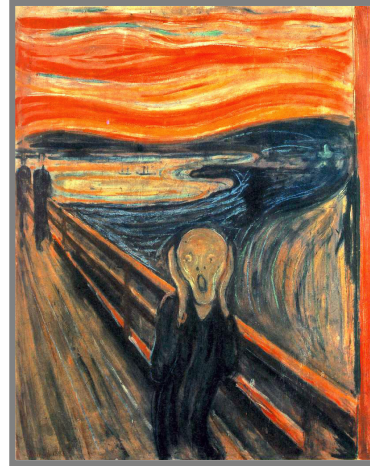


Copyright 2008 - Erlang Training and Consulting Ltd

*Erlang*

## Stress Tests With SMP

- I/O Starvation
- TCP/IP Congestion
- Memory Spikes
- Timeout Fine-tuning
- OS Limitations
- ERTS Configuration Flags
- Shut down Audit Logs

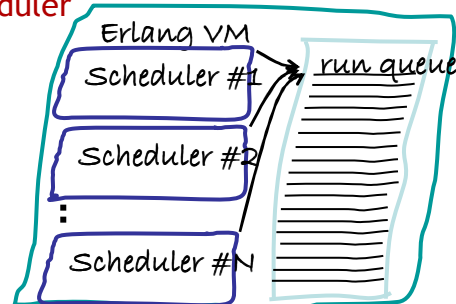


Copyright 2008 - Erlang Training and Consulting Ltd

*Erlang*

## SMP bottlenecks

- The common run-queue
- ETS-tables (when many processes access the same table)
- Message passing (when many processes send to the same process)
- A process can block the whole scheduler
- Memory allocators
- Distribution ports



Copyright 2008 - Erlang Training and Consulting Ltd

*Erlang*

I wrote my  
Erlang system  
in 4 weeks!

Copyright 2008 - Erlang Training and Consulting Ltd



## The Myths of Erlang....

Is it Documented?

Is the developer supporting it?

What visibility does support staff have into what is going on?

- SNMP
- Live Tracing
- Audit Trails
- Statistics
- CLI / HTTP Interface

How much new code was actually written?

Copyright 2008 - Erlang Training and Consulting Ltd





Upgrades  
during runtime  
are Easy!

Copyright 2008 - Erlang Training and Consulting Ltd



## The Myths of Erlang....

**Yes, it is easy for**

- Simple patches
- Adding functionality without changing the state

**Non backwards compatible changes need time time**

- Database schema changes
- State changes in your processes
- Upgrades in distributed environments

**Test, Test, Test**

- A great feature when you have the manpower!

Copyright 2008 - Erlang Training and Consulting Ltd



We achieved  
99.99999999  
availability!

Copyright 2008 - Erlang Training and Consulting Ltd



## The Myths of Erlang....

“As a matter of fact, the network performance has been so reliable that there is almost a risk that our field engineers do not learn maintenance skills”

**Bert Nilsson, Director  
NGS-Programs Ericsson**

*Ericsson Contact, Issue 19 2002*



Copyright 2008 - Erlang Training and Consulting Ltd



## The Myths of Erlang....

99,999 (Five Nines) is a more like it!

- Achieved at a fraction of the effort of Java & C++

Upgrades are risky!

Non Software related issues

- Power Outages
- Networking
- Hardware Faults

## The Myths of Erlang....

99,999 (Five Nines) is a more like it!

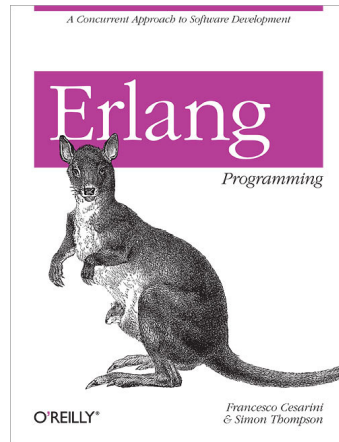
- Achieved at a fraction of the effort of Java & C++

Upgrades are risky!

Non Software related issues

- Power Outages
- Networking
- Hardware Faults

## Shameless Infomercial...



***Out in  
June 2009***

Copyright 2008 - Erlang Training and Consulting Ltd

*Erlang*

# Thank You!

Copyright 2008 - Erlang Training and Consulting Ltd

*Erlang*