

Erlang: Good News, Bad News, How to Win Big

Erik Stenman

CTO

Kreditor Europe AB

KREDITOR

Säljande betalningslösningar

Creative Payment Solutions

Erlang is Gaining Popularity



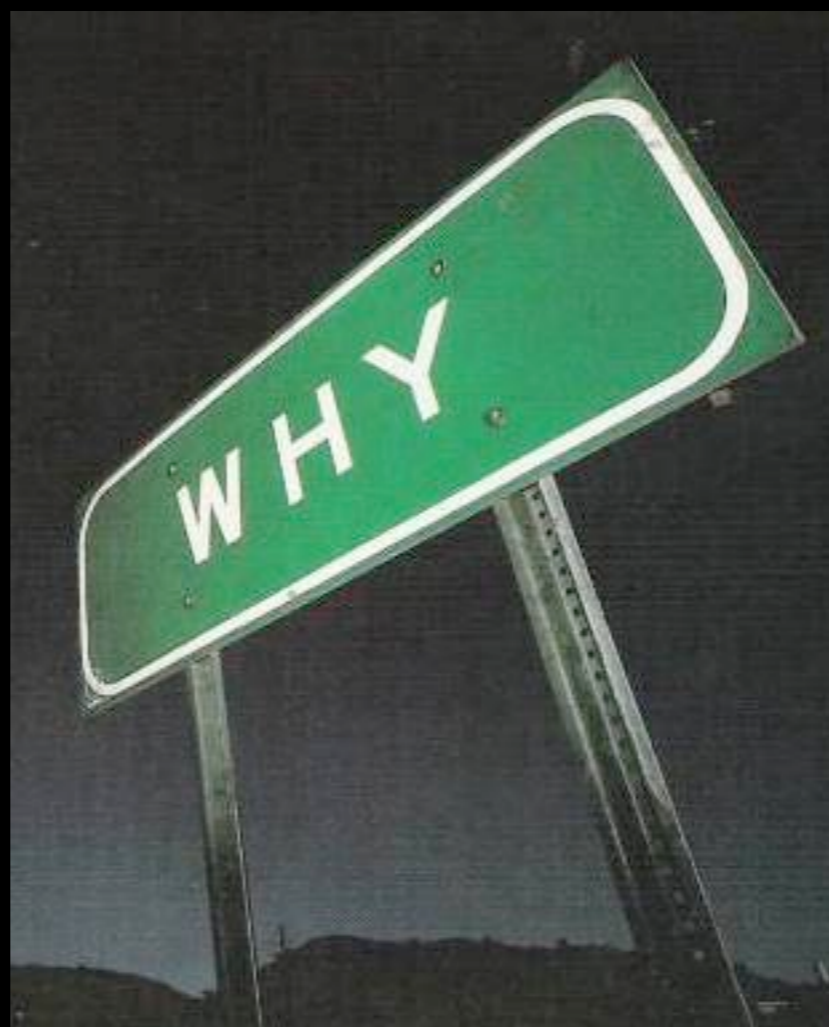
- “Erlang for Concurrent Programming”, by Jim Larson, **Google** in **ACM**:
“Designed for concurrency from the ground up,
the Erlang language can be a valuable tool to help solve concurrent problems.”
(<http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=556>)
- New **Facebook** Chat Feature Scales to 70 Million Users Using Erlang
(<http://highscalability.com/new-facebook-chat-feature-scales-70-million-users-using-erlang>)
- **Amazon's** SimpleDB is built in Erlang
(<http://www.satine.org/archives/2007/12/13/amazon-simplydb/>)
- There are new cool databases such as **CouchDB** and **Scalaris**
- **Dr.Dobb's** 3/12: “It's Time to Get Good at Functional Programming”
“Erlang: A No-Compromises Approach”
(<http://www.ddj.com/development-tools/212201710;jsessionid=AGLEX11IQPUNQQSNDLRSKHSCJUNN2JVN?pgno=2>)



KREDITOR

Säljande betalningslösningar

Why?



KREDITOR

Säljande betalningslösningar

The Erlang Advantage

- **E**asy to get fault-tolerant systems.
- **R**apid development.
- **L**ow-maintenance, easy upgrade.
- **A**bility to leverage multi-core.
- **N**etwork programming is easy.
- **G**ood way to get great programmers.



If you're not cheating, you're not trying hard enough...

KREDITOR

Säljande betalningslösningar

Easy to Build Fault-tolerant Systems

- **Erlang** was designed from the ground up with the purpose of making it easy to develop fault-tolerant systems.
- **Erlang** was developed by **Ericsson** with the telecom market in mind.
- **Erlang** supports processes, distributed systems, advanced exception handling, and signals.
- **Erlang** comes with **OTP**-libraries (**Open Telecom Platform**), e.g. *supervisors* and *generic servers*.

KREDITOR

Säljande betalningslösningar

Rapid Development

- **Erlang** has a number of features to support rapid prototyping and fast development:
 - Automatic memory management.
 - Symbolic constants (atoms).
 - An interactive shell.
 - Dynamic typing.
 - Simple but powerful data types.
 - Higher order functions and list comprehensions.
 - Built in (distributed) database.

KREDITOR

Säljande betalningslösningar

Low-maintenance and Easy Upgrade

- **Erlang** has a number of features that makes it easy to maintain and upgrade:
 - Hot code loading.
 - Distribution.
 - Interactive shell.
 - Simple module system.
 - No shared state.
 - Virtual machine.

KREDITOR

Säljande betalningslösningar

Ability to Leverage Multi Core

- The concept of processes is an integral part of Erlang.
- The Erlang Virtual machine (**BEAM**) has support for *symmetric multiprocessing*.
- No shared memory -- easier to program.
- As Joe Armstrong is found of saying:
“Each year your sequential programs will go slower.
Each year your concurrent programs will go faster.”

KREDITOR

Säljande betalningslösningar

Network Programming is Easy

- Distributed **Erlang** solves many network programming needs.
- Setting up a simple socket protocol is a breeze.
- The binary- (and now bit-) syntax makes parsing binary protocols easy.
- There are simple but powerful libraries for HTTP, XML, XML-RPC and SOAP.

KREDITOR

Säljande betalningslösningar

Good Way to Get Great Programmers

- **Nice paradox:**
The lack of **Erlang** programmers makes it easier for us to find great programmers.
- There are many great C and Java programmers, I'm sure, but they are hidden by hordes of mediocre programmers.
- Programmers who know a functional programming language are often **passionate** about programming.
- Passionate programmers makes **Great Programmers™**.

KREDITOR

Säljande betalningslösningar

Erlang - Background

- Developed by the Computer Science Lab at Ericsson.
- Problem domain - Telephone exchanges.
 - High availability
 - Highly concurrent
 - Real time
 - Distributed
 - Continuous operation
 - In-service upgrades

KREDITOR

Säljande betalningslösningar

Erlang Design Goals*

- How can we build software systems that are as reliable as hardware systems.
- How can we make programming almost as easy as assembling hardware.

* This is my own reconstructed view of what the goal might have been, I was not in any way involved in the design of Erlang.

KREDITOR

Säljande betalningslösningar

Erlang - Insight

To make a fault-tolerant
system you need at
least
two
computers.

KREDITOR

Säljande betalningslösningar

Real - Insight

Actually,
to make a fault-tolerant
system you really need
at least
three
computers*.

* This is an insight about consensus algorithms shown by Leslie Lamport in "The Byzantine Generals Problem" (1982) and in more detail in "Lower Bounds for Asynchronous Consensus" (2004).

KREDITOR

Säljande betalningslösningar

Erlang - Background

- The Erlang designers realized “To do fault tolerant computing we need at least two isolated computers.”
- This lead to concurrent programming with pure message passing and no shared state.
 - Large number of (isolated) processes
 - Communication through message passing
 - No mutable state in processes
 - Pure functional programming

KREDITOR

Säljande betalningslösningar

Erlang - OTP

- A huge part of the success of **Erlang** comes from the standard library **OTP** (**Open Telecom Platform**).
- **OTP** extends the fault tolerance in **Erlang** by providing standard patterns (or behaviours in **Erlang** lingo) for building telco-grade systems.
 - Supervisors, with restart polices
 - Generic servers
 - Generic state machines
 - Logging

KREDITOR

Säljande betalningslösningar

Why not use Erlang?

- The main reasons that I have heard of are:
 1. Politics – **Erlang** is not C/Java, company policy.
 2. One provider – Concern that **Ericsson** will stop supporting **Erlang**.
 3. Lack of programmers – **Erlang** is still not mainstream how can we ensure we get qualified staff?
- When starting a new company, 1 is (usually) not a problem.
- I can't see 2 happening, and **Erlang** is open source anyway.
- When setting up in Stockholm, 3 is not a problem.

KREDITOR

Säljande betalningslösningar

When not to Use Erlang

- Don't try to build a stand-alone GUI.
 - Don't expect to do fancy text handling out of the box.
 - A fancy web-GUI is not easy to do yet.
 - Fast file crunching is not Erlang's forte.
- But Erlang can easily interface with other languages and applications, so use Erlang as the glue to write the robust server core, and plug in the missing parts.

KREDITOR

Säljande betalningslösningar

Erlang Downsides

- It is not mainstream*
 - There are not many third party libraries yet
 - On the upside: There is no DLL-Hell (or Jar-Hell).
 - Even if we can find great programmers it is hard to find enough programmers who know Erlang or who even want to use Erlang.
 - On the upside: Having great programmers is...
...well, just great!

KREDITOR

Säljande betalningslösningar

* A big advantage according to Paul Graham in "Beating the Averages".

Making Erlang More Mainstream

- I have a cunning plan.
- We set up an independent language ranking firm, paid by the **Erlang** community of course.
- They rank **Erlang** as an AAA language.
- Every manager faced with a language decision can then make the non-decision of choosing the triple AAA language **Erlang**.

KREDITOR

Säljande betalningslösningar

Conclusion

- Erlang has many novel features, such as
 - first-class processes,
 - built-on distribution
 - hot-code loading,
 - multi-core support,
 - and a high level of abstraction.
- This makes Erlang very suitable for building high-availability (web-) servers.

Questions?

KREDITOR

Säljande betalningslösningar