

RELEASE: A High-level Paradigm for Reliable Large- scale Server Software

Phil Trinder



UPPSALA
UNIVERSITET



RELEASE



What is RELEASE

- European Research Project
 - EU FP7 STREP
 - Euro 3.6M, ~25 Researchers
- October 2011 – September 2014

Context

Exponential growth in cores

Conventional languages are essentially sequential

Core failure rates constant, so as the number of cores grow exponentially, so do failures!

=> So multicore systems need to be both scalable and robust.

Erlang

A functional language with an **Actor** model of distribution

Widely recognised as a **beacon language** for distributed computing, influencing e.g. languages and frameworks like Scala/Akka, F#, Clojure, Cloud Haskell, ...

RELEASE Aim

- *To scale the radical distributed actor programming paradigm to build reliable general-purpose software, such as server-based systems, on massively parallel machines (10^5 cores).*

RELEASE Aim

- *To scale the radical distributed actor programming paradigm to build reliable general-purpose software, such as server-based systems, on massively parallel machines (10^5 cores).*
- **Doesn't Erlang already provide scalable reliability?** Erlang/OTP has an inherently scalable computation and reliability models, but in practice scalability is constrained
 - VM aspects, e.g. the transitive sharing of connections between all nodes
 - Language aspects, e.g. explicit process placement
 - Tool support

Target Platforms

- We target reliable scalable general purpose computing on stock heterogeneous platforms, i.e.
 - general server-side computation, like a messaging server.
 - standard hardware, operating systems and middleware
 - Not specialised high-performance computing hardware/software stacks.

Objectives

1. To Scale Erlang by co-designing

- an extension, Scalable Distributed (SD) Erlang (WP3)
- the BEAM virtual machine (WP2)

for reliable scalability.

Objectives

1. To Scale Erlang by co-designing

- an extension, Scalable Distributed (SD) Erlang (WP3)
- the BEAM virtual machine (WP2)

for reliable scalability.

Status:

WP2: Improved BEAM concurrency, much in latest Erlang/OTP release

WP3: Designed and Implemented SD Erlang, evaluation in process

Objectives

2. To develop a scalable virtualisation infrastructure (WP4)
 - creates, manages and dynamically scales super-clusters of heterogeneous clusters/cloud instances

Objectives

2. To develop a scalable virtualisation infrastructure (WP4)
 - creates, manages and dynamically scales super-clusters of homogeneous or heterogeneous clusters/cloud instances

Status: Developed

Objectives

3. To develop improved software development technologies for large scale Erlang software (WP5)

- a suite of integrated tools for monitoring, visualization, debugging & refactoring and methodologies for using them

Objectives

3. To develop improved software development technologies for large scale Erlang software (WP5)

- a suite of integrated tools for monitoring, visualization, debugging & refactoring and methodologies for using them

Status: Released a number of profiling tools (Percept2), Dtrace. Developing SD-Erlang specific tools

Objectives

4. To evaluate SD Erlang and associated tools/ methodologies (WP6)

- Case studies:
 - a cloud/cluster deployment tool (WOMBAT)
 - a large simulation (Sim-Diasca).
- Platform: Blue Gene
- Investigate mainstream impact by investigating adding SD Erlang constructs to a popular Actor framework

Objectives/Status

4. To evaluate SD Erlang and associated tools/ methodologies (WP6)

- Case studies:
 - a cloud/cluster deployment tool (WOMBAT) - **under development & evaluation**
 - a large simulation (Sim-Diasca). – **reliability improved, refactored for WOMBAT deployment**
- Platform: Blue Gene - **Erlang port in progress**
- Investigate mainstream impact by investigating adding SD Erlang constructs to a popular Actor framework - **Not due to start**

Overall Strategy

