Publishable Summary

Project Context and Objectives

Every eighteen months during the last thirty years has seen the power of the computer that can be built on a silicon chip double – this has now come to a halt. Instead, chip manufacturers build multiple computers – or cores – on each chip: nearly all PCs are now 'dual' or 'quad' core, and the number of cores it is possible to put on each chip is growing exponentially.

Building software for these multicore systems requires radically new software development technologies that can exploit the platform. Instead of programming a single core, the cores have to be programmed to work together in a coordinated way, and in a way that scales with the numbers of cores. Many expect 100,000-core platforms to become commonplace, and the best predictions are that core failures on such an architecture will be common, perhaps one an hour. Hence we require a programming model that is not only highly scalable but also reliable.

The overall project objective is to develop the first ever scalable concurrency-oriented programming infrastructure and its associated tool set, and hence aims to reduce development times of multicore solutions while delivering increased reliability. Our platform builds on the Erlang language and Open Telecom Platform (OTP) libraries that have concurrency and robustness designed in. Currently Erlang/OTP has inherently scalable computation and reliability models, but in practice scalability is constrained by aspects of the language, virtual machine and toolset.

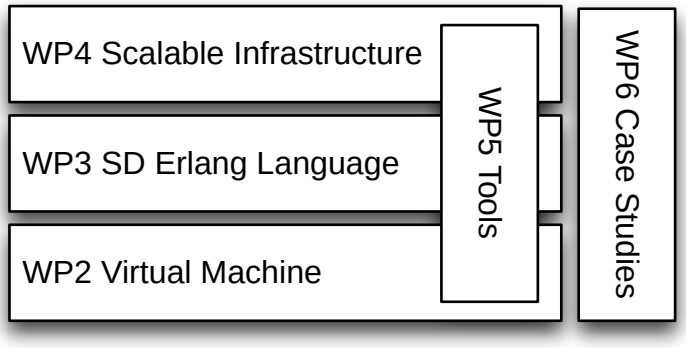

Figure 1 RELEASE Technical Work Package Relationships

Figure 1 shows the work packages and how they inter-relate. We are attacking the scalability challenges at three levels:

- We are evolving the Erlang virtual machine so that it can work effectively in large-scale multicore systems (WP2).
- We are evolving the language to Scalable Distributed (SD) Erlang, and adapt the OTP framework to provide constructs to control how computations are spread across multicore platforms, and coordination patterns to allow SD Erlang to effectively describe computations on large platforms, while preserving performance portability (WP3).
- On top of the language and the virtual machine we are developing a scalable Erlang infrastructure to integrate multiple, heterogeneous clusters (WP4).

To exploit such large platforms, programmers need to be able to understand how their programs are behaving in practice. We will build tools to enable programmers to profile and visualize their SD Erlang applications; to refactor Erlang programs to run scalably and efficiently under SD Erlang; and to debug SD Erlang systems (WP5).

We will demonstrate the effectiveness of the RELEASE approach in two case studies (WP6). EDF will port the Sim-Diasca simulation framework to SD Erlang on the Blue Gene parallel computing platform. Erlang Solutions will build a heterogeneous cloud-based continuous integration framework service using SD Erlang and evaluate it with a representative real-world scenario such as an online chat service.

Work performed since Project Start

In this first year, work has progressed according to plan and the objectives set for the reporting period were met. The results so far can be seen as work in progress towards the two showcases and their different components.

- We have investigated Erlang scalability issues at both Virtual Machine and Language levels.
- To facilitate our scalability investigations we have established an open source repository of scalability benchmarks for Erlang (BenchErl: http://www.softlab.ntua.gr/release/bencherl/), and started to develop larger-scale case-studies.
- We have improved the scalability of the Erlang VM, e.g. reducing contention by introducing lock-free data structures.
- We have designed SD Erlang to address language-level scalability issues in Erlang/OTP, and started the implementation of our design.
- We are developing improved Erlang profiling tools using both DTrace and Percept.
- We have produced a deployment and management inf rastructure (CCL) for exploiting multiple clusters/cloud resources.

Expected Final Results

RELEASE will deliver both conceptual and technological advances in the development of reliable massively parallel general-purpose software. The conceptual advances will be to scale the radical concurrency-

oriented programming paradigm to build reliable general-purpose software, on massively parallel commodity hardware.

Potential Impact and Use

The scientific impact of the project will be a major advance in the technology for developing and effectively using massively-parallel reliable general-purpose software. Specifically we will develop the first ever scalable concurrency-oriented programming infrastructure and its associated tool set. This will contribute to the ICT work programme drive for *increasing competitiveness of European industry and enabling Europe to master and shape future developments in ICT*, and to both the *Digital Agenda* and the *HiPEAC Roadmap*.

The project has significant potential market impact as the Erlang user base is growing rapidly, and is being adopted by high-profile companies like Amazon, Yahoo! and Facebook. The project will enable developers to exploit large parallel systems and to reduce the time to market of software products. In the project we plan to develop to two new products and to contribute Open Source software, tools, and infrastructure.

The project has socio-economic impact by enhancing the competitiveness of the European ICT sector, and thus stimulating economic growth and job creation. Thus one of the ultimate aims can be seen to be the creation of *more and better jobs*.

Project logo:

Project website: **http://www.release-project.eu/**

Contact: Phil Trinder (Coordinator): P.W.Trinder@hw.ac.uk