



PhD Position

Infrastructure and language kernels for IoT Systems

Description

Over the last years, the RMOD team of INRIA Lille and the CAR theme of IMT Lille Douai have been working together on creating tiny language core. For example, Guillermo Polito demonstrated in his PhD a fully reflective kernel that fits into 80 kb of memory and that it is possible to have hyper specialized kernels down to 11 kb. We have also worked on remote debugging (PhD of N. Papoulias) and dynamic code updates (PhD of P. Tesone) of such kernels. All of these works are prototyped in Pharo. More recently, RMOD have been working on advanced probes mechanisms (M. Denker) and a solid remote debugging infrastructure (D. Krudiaskov).

The goal of this PhD is to revisit the architecture around such mini-kernels for building IoT applications. The following tasks are foreseen :

- improve the tools to : edit, compile, debug, deploy and update such kernels on IoT devices. Learning how to debug remotely and dynamically update such IoT systems using the Pharo environment. This task will be in cooperation with M. Denker and D. Krudiakov on remote debugging for IoT and G. Polito for the kernel edition and tooling,
- define some language extensions to manage groups of IoT devices to program them at once. Managing hundreds or even thousands of IoT devices is a challenging task. We want to explore different solutions to help deploying and updating groups of IoT devices using some registration mechanism in a cloud server or some groups/roles based approaches for example,
- expressing the architecture of IoT applications. We would like to explore also how to express IoT architectures and what are the abstractions that should be offered to developers such as expressing event-driven architectures with declarative ECA (Event-Conditions-Actions) rules. But we will study a couple of typical IoT applications.
- dynamically update an IoT application. An IoT application needs to adapt itself because unreachable or faulty devices or the diminution of available bandwidth. We would like that the whole application can reconfigure itself in such situations as Guillaume Grondin proposes it in its PhD.
- Lightweight virtual machines. Virtual machines in the IoT context are very powerful for incremental deployment or dynamic updates. Although they are tuned for speed,

they consume space compared to a non VM-based program. In this task, we would like to investigate what is the minimal memory consumption that we can reach for a VM usable for IoT. To do so we will use a standard VM such as Cog (the open-source virtual machine of Pharo) and degrade it. We will use it since our minimal kernels are running on it.

Note that ZweiDenker GmbH is interested in collaboration on the IoT management cloud infrastructure.

Application

To apply, please send us :

- a CV,
- a copy of your Master diploma
- a copy of your Master thesis
- 2 (two) reference letters, with the contact details of the referents
- links to videos of demos of your experiments and/or simulations

The application materials should be sent no later than August 30th 2017 by email to Prof. S. Ducasse stephane.ducasse@inria.fr. Email subject must start with : [PhD-RMoD-CAR-2017].

Bibliography

- G Polito, Stéphane Ducasse, N Bouraqadi, L Fabresse, M Mattone. Virtualization Support for Dynamic Core Library Update. Onward!, Oct 2015, Pittsburg, USA.
- Guillermo Polito. Virtualization Support for Application Runtime Virtualization and Extension. Ph.D. Thesis 2015. Co-delivrée par l'Université de Lille et l'École des Mines de Douai.
- Extended results of Tornado : A Run-Fail-Grow approach for Dynamic Application Tailoring. Commanditeur : École des mines de Douai, France. 50p, July 2014
- Nick Papoulias, Noury Bouraqadi, Luc Fabresse, Stéphane Ducasse and Marcus Denker, Mercury : Properties and Design of a Remote Debugging Solution using Reflection, Journal of Object Technology, 14, 1 :1-36, 2015
- Madcar : an abstract model for dynamic and automatic (re-)assembling of component-based applications G Grondin, N Bouraqadi, L Vercouter. International Symposium on Component-Based Software Engineering, 360-367.