Demonstration of a context-switch method for heterogeneous reconfigurable systems

Arief Wicaksana, Alban Bourge, Olivier Muller, Frédéric Rousseau

Demonstration of a Context-Switch Method for Heterogeneous Reconfigurable Systems

Arief Wicaksana, Alban Bourge, Olivier Muller, Frédéric Rousseau
TIMA Laboratory - Grenoble-INP/UGA/CNRS
46 avenue Félix Viallet, 38000 Grenoble, FRANCE
{arief.wicaksana,alban.bourge,olivier.muller,frederic.rousseau}@imag.fr

I. Introduction

Nowadays, FPGAs are integrated in high-performance computing systems, servers, or even used as accelerators in System-on-Chip (SoC) platforms. Since the execution is performed in hardware, FPGA gives much higher performance and lower energy consumption compared to most microprocessor-based systems. However, the room to improve FPGA performance still exists, e.g. when it is used by multiple users. In multi-user approaches, FPGA resources are shared between several users. Therefore, one must be able to interrupt a running circuit at any given time and continue the task at will. An image of the state of the running circuit (context) is saved during interruption and restored when the execution is continued. The ability to extract and restore the context is known as context-switch.

In the previous work [1], an automatic checkpoint selection method is proposed for circuit generation targeting reconfigurable systems. The method relies on static analysis of the finite state machine of a circuit to select the checkpoint states. States with minimum overhead will be selected as checkpoints, which allow optimal context save and restore. The maximum time to reach a checkpoint will be defined by the user and considered as the context-switch latency. The method is implemented in C code and integrated as plugin in a free and open-source High-Level Synthesis tool AUGH [2].

II. Demonstration

In this demonstration, we present the context-switch method [1] implemented in heterogeneous reconfigurable systems using a network-connected framework. SoC-FPGA platforms with a CPU tightly-coupled with an FPGA are being used in the framework. The demonstration framework consists of two FPGA-SoC platforms, a server with tool-chain installed and a network storage disk. We use two different platforms in the framework, a ZC706 Evaluation Board from Xilinx and an Arria V SoC Development Kit from Altera, to show the genericity of our method even for FPGAs from different vendors. The server provides a tool-chain suite of corresponding platforms for configuration purpose and the network storage is implemented with NFS protocol.

A graphical and a non-graphical applications will be used to perform context-switch on the boards. These applications will consume test vectors as input and perform the computations. For the graphical application, an output video will be prepared for each board so attendees can follow from the screen. For the non-graphical application, the flow of the execution will be observed from user's PC connected to the server.

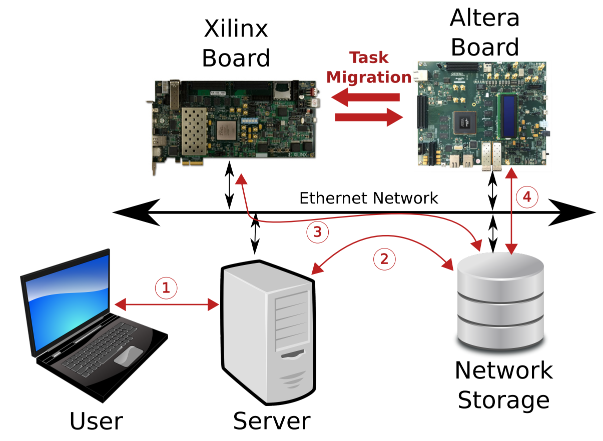

Figure 1. Demonstration Framework and its Typical Execution Flow

Figure 1 shows the general overview of our demonstration framework and the execution flow. A typical execution flow is given below. Other variations of boards and steps are possible.

1) A user connects to *Server* either via local connection or framework's network, put its application and launch the execution.
2) The configuration of the circuit is generated in *Server* and saved in *Network Storage* with its respective input test vectors.
3) The CPU on *Xilinx Board* which detects the configuration file in *Network Storage* and the test vectors will program the FPGA and launch the execution. When there is an interruption, it will retrieve the context from FPGA and save it in *Network Storage*.
4) The CPU on *Altera Board* which detects the bitstream and the context in *Network Storage* will program the FPGA and continue the execution. When there is no other interruption, it will finish the execution and put the results in *Network Storage*.

References

[1] A. Bourge, O. Muller, and F. Rousseau, "Automatic High-Level Hardware Checkpoint Selection for Reconfigurable Systems," in *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, May 2015, pp. 155–158.
[2] A. Prost-Boucle, O. Muller, and F. Rousseau, "Fast and standalone Design Space Exploration for High-Level Synthesis under resource constraints," *Journal of Systems Architecture*, vol. 60, no. 1, pp. 79–93, Jan. 2014.