FPGA acceleration of Spark applications in a Pynq cluster

Christoforos Kachris
ICCS/NTUA, Athens, Greece

Elias Koromilas, Ioannis Stamelos
ECE, NTUA, Athens, Greece

Dimitrios Soudris
ECE, NTUA, Athens, Greece

In this paper we present a framework for the seamlessly utilization of hardware accelerators in heterogeneous SoCs that are used to speedup the processing of Spark data analytics applications.

The main features of this framework are the following:

- The development of an efficient set of libraries that hide the accelerator's details to simplify the incorporation of hardware accelerators in Spark
- Mapping of the accelerated Spark to a heterogeneous all-programmable MPSoC (Zynq) based on the Pynq platform
- A performance evaluation for a use-case on machine learning (logistic regression) that shows how the proposed framework could achieve up to 3x speedup compared to a high performance x86_64 processor.
- A cluster of 4 nodes based on the Pynq platform that are used to accelerate the Spark Logistic regression algorithm using only a fraction of the energy consumption of the typical high-performance servers.

On top of the Pynq framework, we have efficiently mapped the Spark framework and we have adapted it to communicate with the hardware accelerators located in the programmable logic of the Zynq system. Spark master node is hosted on a personal computer that comes with an Intel i5 x86_64 architecture processor, but also an Intel x86 or ARM system could be used. Worker nodes are hosted on PYNQ's ARM cores. Figure 1 shows the proposed cluster architecture. More workers beyond PYNQ cores' could be used to take advantage of all the available processing resources. A python API is used for each accelerator that is used for the communication with the hardware accelerator. Each Python API is communicating with the C library that serves as the hardware accelerator driver.

On the reconfigurable logic part, the hardware accelerators for the specific application are hosted. The hardware accelerators are invoked by the python API of the Spark application. Therefore, the only modification that is required is the extension of the python library with the new function calls for the communication with the hardware accelerator.

The utilization of hardware accelerators directly from Spark has two major advantages; firstly, the application in Spark remains as it is and the only modification that is required is the replacement of the machine learning library's function with the function that invokes the hardware accelerator. Secondly the invoking of the hardware accelerators from the python API eliminates many of the original layers thus making faster the execution of these tasks. The python API invokes the C API that serves as a hardware acceleration's library.

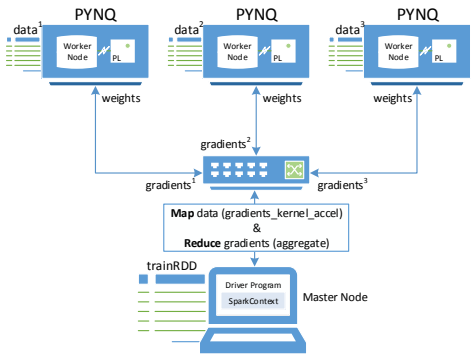

Fig. 1. Spynq architecture (LR MapReduce)

To evaluate the proposed framework, we have developed a hardware accelerator for Logistic Regression (LR) training with BGD and more specifically for the gradients kernel. The hardware accelerator has been implemented using the Xilinx High-Level Synthesis (HLS) tool.

In Spark $gradients\_kernel$ can be parallelized using Map-Reduce, so partial gradients are computed in each Worker, using different chunks of the training set, and then the Master aggregates them and updates w. When the Spark user wants to utilize the hardware accelerator, the only change that needs to be made is the replacement of the Spark *mllib* library with the *mllib_accel* library. Therefore, the user can speedup the execution time of the Spark application with a simple replacement of the libraries that wish to accelerate.

The performance evaluation shows that the cluster of 4 nodes using Pynq can achieve up to 3x speedup compared to the typical high-performance server processors using in data centers and up to 18x lower energy consumption. The architecture and detailed performance evaluation on this platform is shown in [1].

This project has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 687628 - VINEYARD H2020. We also thank Xilinx University Program for the kind donation of the software tools and hardware platforms.

REFERENCES

[1] Christoforos Kachris, Elias Koromilas, Ioannis Stamelos, and Dimitrios Soudris. SPynq: Acceleration of Machine Learning Applications over Spark on Pynq. In *International Conference on Embedded Computer Systems: Architectures, MOdeling, and Simulation (SAMOS 2017)*, July 2017.