Master's Thesis Engineering Technology

2016-2017

How to distribute most efficiently a computation intensive calculation on an Android device to external compute units with an Android API?

Lander Beckers

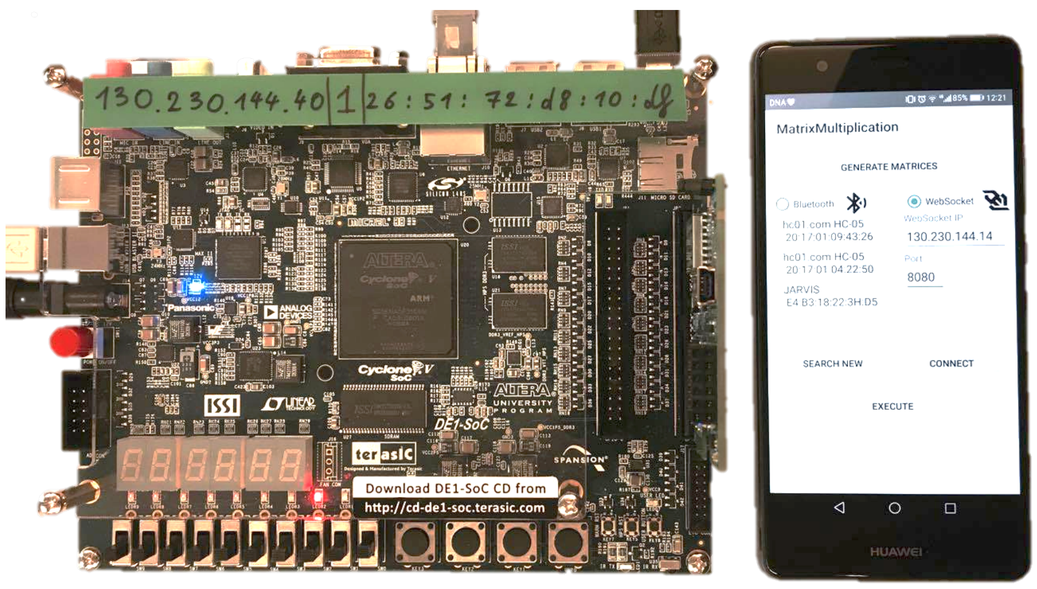
Master of Electronics and ICT Engineering Technology

Henning Lakiere

Master of Electronics and ICT Engineering Technology

Introduction

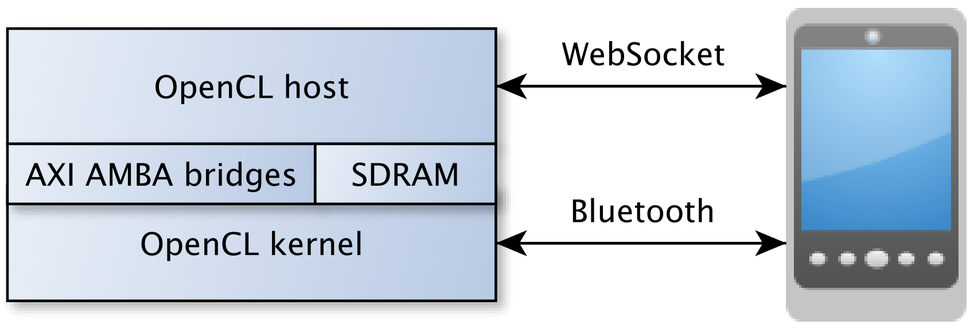
Although *CPUs* increased in speed and throughput during the last few years, computation intensive calculation are still slow on moderate *CPUs*, like smartphones. It would be interesting to compare execution times of a computation intensive calculation on a *CPU* and *FPGA* inside an Android phone and DE1SoC board, respectively. The figure below shows our setup.



The use case researched in this thesis is the transfer of a matrix multiplication from the android phone to a *FPGA*, by the use of Bluetooth or WebSocket. This forms two questions. Firstly, **is it possible to calculate the same matrix multiplication faster on a *FPGA* than on an Android phone?** Secondly, **how do the data transfer times relate to an increasing matrix size?** A final conclusion can be formed combining the answers on these two questions.

Method

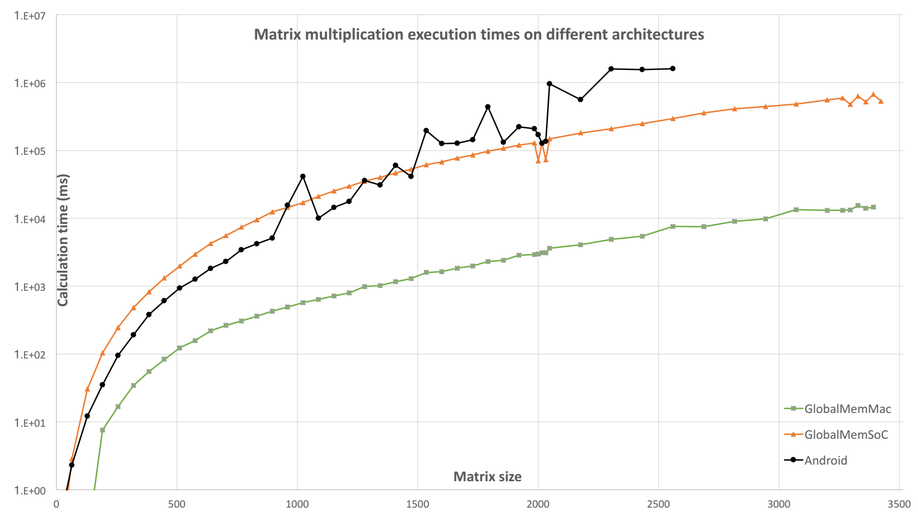
The matrix multiplication data was transferred using Bluetooth or WebSocket, as can be seen in the figure below. While the WebSocket is implemented directly on the HPS of the SoC, the Bluetooth data are received using the TerASIC RFS daughter card and must be transferred from *FPGA* to *CPU* using the AXI AMBA bridges. Figure 2 shows the data communication and *OpenCL* implementation.



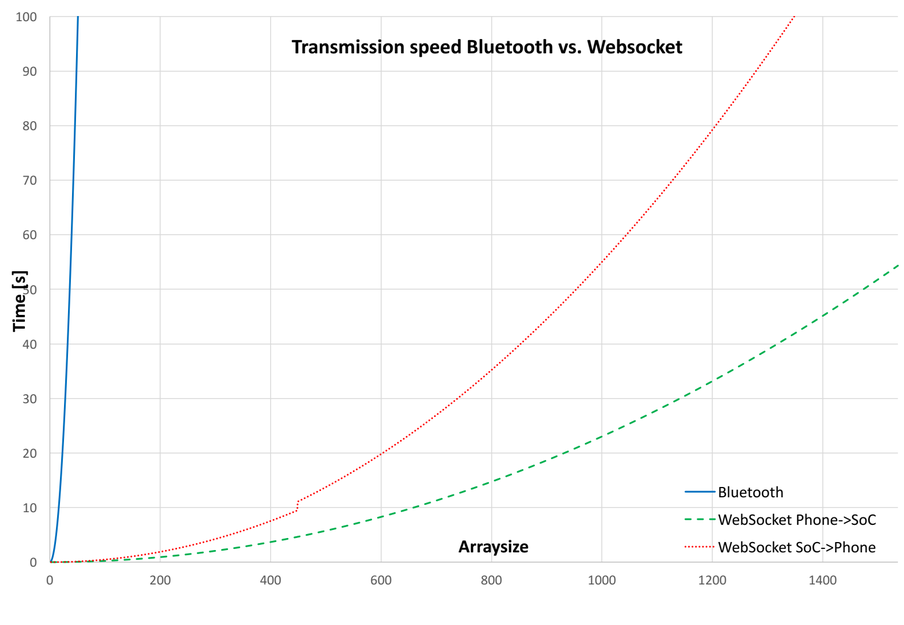
Once the data is transferred into the SoC, *OpenCL* is used to accelerate the matrix multiplication. As can be seen in figure 2, *OpenCL* data is saved into on-chip SDRAM memory and the AXI AMBA bridges are used to communicate with the kernel.

Results

Firstly, matrix multiplication execution times for the *OpenCL* kernel and Android phone are compared, corresponding to their increasing matrix size, the graph below. Although calculating a matrix multiplication using the *OpenCL* kernel on the DE1SoC seems to be slower than multiplying directly on the Android phone, the calculations on the *SoC* progress in a continuous motion. However, when the same *OpenCL* kernel is executed on the MacBook Pro, then the achieved calculation times are significantly faster and they evolve smoother.



Secondly, Bluetooth and WebSocket matrix transfer times are compared in function of their matrix size. Bluetooth is much slower than WebSocket, concluded from the gaph below.



Conclusion

The final conclusion is made out of the next two sub conclusions. **It is possible to win time by transferring the calculation of a matrix multiplication to an external compute unit**, but there can be a significant difference in calculation time depending on the architecture. It would be faster, but more expensive, to replace the DE1SoC by the MacBook Pro.

A more efficient way of improving the system would be using a faster data transfer protocol. Although **WebSocket is 500 times faster than Bluetooth**, transportation is still the slowest link in the chain. Increasing matrix transfer speed will be the main factor to improve the current results.

Supervisors / Cosupervisors: K. Aerts, J. Nurmi, K. Wang

TAMPERE UNIVERSITY OF TECHNOLOGY

UHASSELT

KU LEUVEN