Master/Bachelor thesis proposal
Use deep learning to evaluate performance of parallel ADAS applications with OpenCL

1 Introduction

High performance computing applications are stepping into the automotive area. In the future vehicles, both computation-powerful GPUs and energy-saving FPGAs are integrated to give an optimal leverage on the ECUs. For this heterogeneous system, OpenCL is well suited as the programming framework since it enables the executions of application across multiple devices without changing the source code.

However, due to the huge discrepancies in software implementation and hardware architecture, the performance cannot be smoothly ported between accelerators like GPUs and FPGAs. Typically the optimal execution case on each device can only be found via tuning the parameters exhaustively, which is rather time-consuming.

Based on OpenCL, we want to obtain the software execution features of the devices via micro-benchmarking, a technique used to outline the performance figure of the device. With the device-specific characteristics, it's afterwards possible to schedule different devices to get an optimal execution of on-vehicle ADAS applications.

2 Motivation and Goals

Based on the benchmarkings we have, your goal is to reveal the relationship between the input functional parameters and the final profiling indicators. Since the input sample space is very large, it's non-trivial to get the quantitative relations via naive regression analysis models.

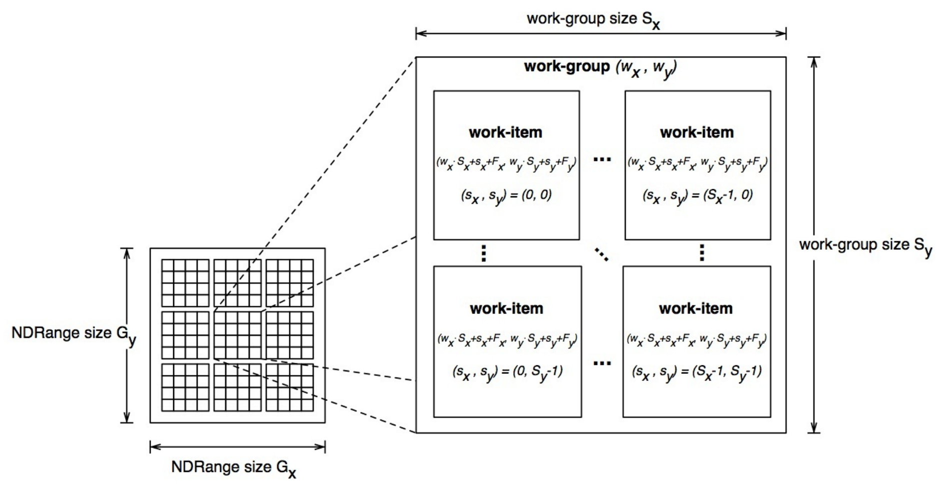
work-group size S_x
work-group (w_x, w_y)
work-item
$(w_x \cdot S_x+s_x+F_x, w_y \cdot S_y+s_y+F_y)$
$(s_x, s_y) = (0, 0)$
work-item
$(w_x \cdot S_x+s_x+F_x, w_y \cdot S_y+s_y+F_y)$
$(s_x, s_y) = (S_x-1, 0)$
work-item
$(w_x \cdot S_x+s_x+F_x, w_y \cdot S_y+s_y+F_y)$
$(s_x, s_y) = (0, S_y-1)$
work-item
$(w_x \cdot S_x+s_x+F_x, w_y \cdot S_y+s_y+F_y)$
$(s_x, s_y) = (S_x-1, S_y-1)$
work-group size S_y
NDRange size G_y
NDRange size G_x

Figure 1: setting of parallel granularity in OpenCL

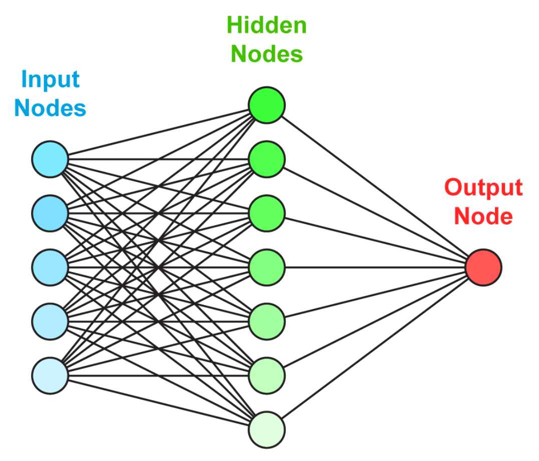

Figure 2: typical neural network

As deep learning is well suitable to process analysis between two groups of multi-dimensional data, we plan to perform the regression analysis via a multi-layer model like the popular CNN and RNN methods. For instance, as seen in figure 1, the number of work item W_I within a work group highly determine the parallel granularity, hence influencing the execution time T, when executing the OpenCL program. Therefore, we can use the neural network shown in figure 2 to train the collected input-output dataset and then predict output with unknown inputs.

3 Your tasks

- Understand the basic concepts in OpenCL and know the principle of neural network.
- Build the performance model using collected data and deep learning algorithms.

4 Requires

- Basic knowledge on programming and algorithm design
- Good knowledge on C++ programming

5 Contact

Xiebing Wang	Kai Huang
Institut für Informatik VI, TUM	Institut für Informatik VI, TUM
Room MI 03.07.059	Room MI 03.07.042
wangxie@in.tum.de	kai.huang@in.tum.de
+49.89.289.18128	+49.89.289.18111