Designing Hardware Acceleration using Dependency and Communication Graphs

xvrana20@stud.fit.vutbr.cz

Abstract

When designing application for time consuming tasks usual method of software acceleration is parallel processing. This allows balancing the load between the resources provided by the processor thus reducing needed time. Since CPUs are designed to handle large set of diverse tasks. This flexibility reduces maximum processing power. If the task demands more computation resources we can use hardware acceleration. Technologies like FPGA and ASICs provide much higher processing power and can reduce CPU load.

For hardware accelerated design we need to decide which operations are better to keep in software and which should be transferred to hardware. That means we need to know the time complexity of the operations. However if formerly software module needs to communicate with other components, transferring it to hardware can increase overhead of this communication. This can result in very little if any performance improvement, even increasing the time needed to perform the task.

The presentation will show building the software application graph model with vertices modelling the modules and edges showing data dependencies and communication. Both vertices and edges are assigned a weight showing time resources and communication intensity. The vertices are then colourized based on the decision if the module will stay in software or will be implemented in hardware. With the graph colourized the vertices and edges will be assigned new weights. The total weight of the graph will show, if the new implementation is viable or the components should be distributed differently.