Compiler for Mapping Stream Processing Applications onto Real-Time Heterogeneous Multiprocessor Systems

S.J. Geuns, B.H.J. Dekens, P.S. Wilmanns, J.P.H.M. Hausmans, M.J.G. Bekooij
University of Twente, Department of EEMCS, Enschede, The Netherlands
s.j.geuns@utwente.nl

ABSTRACT

Heterogeneous multiprocessors system are employed for power-efficiency reasons in wearable software defined radios. These systems are hardware cost-effective and deliver a superior performance compared to their homogeneous counterparts. However these systems are notoriously hard to program without tool support, which makes it is desirable that programming is simplified with the help of an optimizing multiprocessor compiler for stream processing applications.

This demonstration shows our multiprocessor compiler for mapping real-time stream processing applications onto our real-time heterogeneous multi-core system. The applications are described as sequential programs and are compiled into parallel task graphs. Buffer capacities are computed using dataflow analysis techniques given the real-time constraints of the application. Our multi-core system contains 16 MicroBlaze processor cores as well as two hardware accelerators and is prototyped on a Xilinx Virtex-6 FPGA. A connection-less communication ring is used for inter-processor communication. Our system is equipped with an analog RF front-end, which enables us to demonstrate PAL-video reception and decoding.

1. REAL-TIME MULTIPROCESSOR COMPILER

The input of our compiler Omphale [2] is a sequential program written in the coordination language OIL. The OIL language has the look and feel of the C-programming language but excludes the use of pointers, recursion and dynamic memory allocation at the top level. As a result the compiler can always perform dependency analysis at array granularity and can compute memory usage at compile time. OIL programs can contain arbitrary if-statements, while-loops and array index expressions such that stream processing applications with modes can be described.

The compiler translates an OIL program into a parallel tasks graph. It is guaranteed by construction that this task graph is deadlock free because a sequential OIL program is by definition deadlock free. Besides the parallel task graph a structured VPDF model [2] is generated for real-time analysis. Such model allows for dynamic applications and can be efficiently analysed due to its structure.

2. HETEROGENEOUS MULTI-CORE PLATFORM

The main target of our compiler is our heterogeneous multi-core system Starburst [1]. This system contains 16 MicroBlaze processor cores and 2 hardware accelerators. The hardware accelerators process streams and communicate via FIFOs with tasks that run on the MicroBlaze cores.

Only starvation-free schedulers are employed in our multi-core system which simplifies dataflow analysis. By making use of starvation-free schedulers the analysis problem can be linearised without a significant loss in accuracy of the analysis results [3].

Interprocessor communication is realised using a low-cost communication ring that supports point-to-point communication. The applied slot reservation mechanism of this ring enables the derivation of application level performance guarantees. Slots which are reserved for one processor but are not used by it can be claimed by other processors which makes this ring work-conservering.

While the system is optimized for stream processing applications it is also suitable for Pthread applications by making use of a software layer that enables to abstract from the memory model of the hardware [4].

The multiprocessor system is extensively used as an experimental platform for the real-time systems course at the University of Twente. Besides stream processing applications also graphics applications and games have been developed for this system.

3. DEMONSTRATED APPLICATIONS

The capabilities of the platform are shown using a real-time PAL video decoder application that processes a live video stream. This application consists of 16 tasks of which two tasks are executed by a hardware accelerator. Furthermore, the capabilities of our compiler are demonstrated by making use of several didactical applications described in OIL.

4. REFERENCES

[1] B. Dekens, P. Wilmanns, M. Bekooij, and G. Smit. Low-cost guaranteed-throughput communication ring for real-time streaming mpsocs. In *Design and Architectures for Signal and Image Processing (DASIP)*, 2013.

[2] S. Geuns, J. Hausmans, and M. Bekooij. Automatic dataflow model extraction from modal real-time stream processing applications. In *Proc. Int'l Conf. on Languages, Compilers, Tools and Theory for Embedded Systems (LCTES)*, 2013.

[3] J. Hausmans, S. Geuns, M. Wiggers, and M. Bekooij. Dataflow analysis for multiprocessor systems with non-starvation-free schedulers. In *Proc. Int'l Workshop on Software and Compilers for Embedded Systems (SCOPES)*, 2013.

[4] J. Rutgers, M. Bekooij, and G. Smit. Portable Memory Consistency for Software Managed Distributed Memory in Many-Core SoC. In *Reconfigurable Architectures Workshop (RAW)*, 2013.