Master Thesis

Improving performance of heterogeneous manycores with on-chip pattern mining

Keywords: processor architecture, data mining, pattern sampling, multicore, energy consumption, performance task and data placement, FPGA design, RISC-V

Laboratory: INRIA – IRISA, Rennes, France
Team: CAIRN and LACODAM
Place: Rennes or Lannion

Supervisors: Olivier Sentieys (CAIRN), Alexandre Termier (LACODAM), Cédric Killian (CAIRN)
Contacts: Olivier.Sentieys@inria.fr, Alexandre.Termier@irisa.fr, Cedric.Killian@irisa.fr

Context

Most of nowadays consumer electronics devices such as smartphones, tablets or set-top boxes are powered by Multi-Processor System-on-Chip (MPSoC). MPSoC integrate on a single die multiple generalist computation cores, memories and specialized or reconfigurable accelerators, allowing to improve performance as well as to reduce energy consumption. MPSoC are also named~~s~~ as Heterogeneous Multicores. The main components of the MPSoC are connected together using a Network-on-Chip (NoC). Analysts predict that future MPSoC will feature thousands of computation units after 2020.

In such highly parallel architecture, the affectation of a computation unit to a task (task placement), as well as the storage of a piece of data in one of the memories (data placement), are key factors for run-time performance and energy consumption. Allocating too much data that are very frequently accessed by different cores to a single memory is likely to create congestion on the NoC around that memory, increasing execution time. Oppositely, placing a low priority task alone on a computation unit while the system is not on a high load will prevent to offline that computation unit, which would have allowed to reduce energy consumption. Another important issue is to put data in physical memories close (in terms of NoC wiring) to the computation units hosting the tasks processing these data, in order to reduce NoC usage, memory transfer and cache miss, hence reducing energy consumption and execution time.

Such placement decision can usually not be completely taken in advance, and have to be made dynamically at run time. Existing placement mechanics are designed to be simple to implement and fast to execute, however they have very limited information on execution history, which can lead to inappropriate placement. This study is similar to the principle of hardware prefetching in cache, which is critical for improving performance of high-end processor systems [LKV12].

A promising solution is to analyze application execution history,~~;~~ traffic in the NoC, or access to the shared memory, to detect patterns in the execution and exploit these patterns to take better placement decisions. We have shown that when analyzing post-mortem execution data with data mining methods, it was possible to detect behaviors leading to subpar performance [LTP13, LTP14].

Discovering potentially complex correlations is handled by Pattern Mining algorithms, whose goal is to explore a huge combinatorial space efficiently. There are several ways to reduce the number of patterns output to a manageable size, one of the most drastic being pattern sampling. The idea is to only compute a set of frequent patterns, and to have statistical guarantees that these patterns are a “good” sample of the complete set of patterns. Pattern sampling algorithms are a promising approach for performing a pattern-based analysis of data streams [DLD17, Cha+14, Bol+11, AZ09]. In recent work we have shown that an FPGA accelerator for pattern sampling can outperform a state-of-the-art implementation on a server class CPU [GST19]. We also had contributions to sampling on high-rate data streams and would like to continue in this direction by embedding such hardware pattern miner into a multicore architecture as a mean to improve its performance.

Objectives

The objective of this Master thesis is to improve placement decision during execution using data mining techniques. This means designing novel data mining algorithms that can analyze execution information on-chip, using as few resources as possible, and provide relevant information to take better placement decisions. This on-chip miner can be seen as a smart hardware monitor that can help a supervisor, e.g., the Operating System, to take decision to improve performance and energy efficiency. Example of decision could be to change voltage/frequency, migrate tasks, perform just-in-time (JIT) compilation to modify code parallelism or version, migrate tasks from HW to SW or from a core to another, or change parallelism on the HW.

A second objective of this work, related to the field of computer architecture, is to determine which type of data should be collected in the context of a heterogeneous multicore (i.e, processor cores, memory banks, and hardware accelerators), which type of decision should be taken and what would be the cost overhead of implementing pattern mining algorithms in hardware. Sense, process and react (in real time and on-chip) is a key technique for improving performance of emerging and future multicore systems. First, on-chip hardware and/or software performance monitors must be defined to collect data such as cache miss, memory access, temperature, network contention, etc. Then, different parameters of the architecture can be modified depending on the results of pattern mining. Finally, a study on the complexity of accelerating in hardware the data mining algorithms has to be carried out. We do not want this hardware monitor to be more complex than the core itself!

The last objective of this work will be to propose a solid evaluation method in order to estimate the gains of the proposed techniques. This methodology will include the choice of a variety of workloads to run which may benefit from the dynamic data or task placement strategies proposed, as well as measures of performance gains in various areas, such as execution time or energy consumption.

Architecture-level simulation will be performed on a multicore simulator, such as sniper [SniperSim] with which we have some experience.

The candidate is required to have a strong interest for algorithms, as well as a solid programming background in C/C++. A prior data mining experience will be appreciated, but is not necessary. On the computer architecture side, the candidate is required to have a good knowledge in computer architecture in general and in multicores in particular. Skills on architecture design, and especially on high-level synthesis of hardware from C/C++ will be appreciated.

Note on the bibliography: for the bibliography part of this Master thesis, some pattern sampling techniques will be studied from the literature as well as our previous work in FPGA accelerators. The project will consist in collecting some data from multicore simulation (e.g. the traffic on the NoC) and to apply some analysis using the previously studied pattern sampling algorithms to see if some patterns can be extracted (e.g. are there some patterns like multicast/broadcast?) and used to improve performance of the same code executed on the multicore with another configuration.

References

[LTP13] Sofiane Lagraa, Alexandre Termier, Frédéric Pétrot: Data Mining MPSoC Simulation traces to identify concurrent memory access patterns, DATE 2013

[LTP14] Sofiane Lagraa, Alexandre Termier, Frédéric Pétrot: Scalability bottlenecks discovery in MPSoC platforms using data mining on simulation traces, DATE 2014 (best paper award)

[LKV12] Jaekyu Lee, Hyesoon Kim, and Richard Vuduc. 2012. When Prefetching Works, When It Doesn’t, and Why. ACM Trans. Archit. Code Optim. 9, 1, Article 2 (March 2012), 29 pages.

[JS+15] B. Janssen, F. Schwiegelshohn, M. Koedam, F. Duhem, L. Masing, S. Werner, C. Huriaux, A. Courtay, E. Wheatley, K. Goossens, F. Lemonnier, P. Millet, J. Becker, O. Sentieys, and M. Hubner, “Designing Applications for Heterogeneous Many-Core Architectures with the FlexTiles

Platform,” in 15th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS), 2015.

[HCS15] C. Huriaux, A. Courtay, and O. Sentieys, “Design Flow and Run-Time Management for Compressed FPGA Configurations,” in IEEE/ACM Design, Automation and Test in Europe (DATE), Grenoble, France, 2015, pp. 1551-1554.

[GST19] Mael Gueguen, Olivier Sentieys, Alexandre Termier: Accelerating Itemset Sampling using Satisfiability Constraints on FPGA. DATE, 2019, pp. 1046-1051.

[AZ09] Mohammad Al Hasan and Mohammed J. Zaki. “Output Space Sampling for Graph Patterns”. In: Proc. VLDB Endow. 2.1 (Aug. 2009), pp. 730–741.

[Bol+11] Mario Boley et al. “Direct Local Pattern Sampling by Efficient Two-step Random Procedures”. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD’11. 2011, pp. 582–590.

[Zha+13] Yan Zhang et al. “An FPGA-Based Accelerator for Frequent Itemset Mining”. In: ACM Trans. Reconfigurable Technol. Syst. (TRETS) 6.1 (May 2013), 2:1–2:17.

[Cha+14] Supratik Chakraborty et al. “Distribution-Aware Sampling and Weighted Model Counting for SAT”. In: arXiv:1404.2984 (Apr. 2014).

[DLD17] Vladimir Dzyuba, Matthijs van Leeuwen, and Luc De Raedt. “Flexible constrained sampling with guarantees for pattern mining”. In: Data Mining and Knowledge Discovery (Mar. 2017). (Visited on 06/15/2017).

[SniperSim] https://snipersim.org//w/The_Sniper_Multi-Core_Simulator

Mis en forme : Français (France)

Mis en forme : Anglais (États-Unis)

Mis en forme : Anglais (États-Unis)

Mis en forme : Anglais (États-Unis)