Improving Data Partitioning Performance on OpenCL-based FPGAs

Zeke Wang Bingsheng He
Nanyang Technological University, Singapore

Wei Zhang
Hong Kong University of Science and Technology

***Abstract*—We investigate the performance of relational database applications on recent OpenCL-based FPGAs. As a start, we study the performance of data partitioning, a core operation widely used in relational databases. Due to the random memory accesses, data partitioning is time-consuming and can become a major bottleneck for database operators such as hash joins. We start with the state-of-the-art OpenCL implementation which was originally designed for the CPU/GPU, and find that such an implementation suffers from lock overhead and memory stalls. To resolve those overheads, we develop a simple yet efficient multi-kernel approach to leverage two emerging features in Altera OpenCL SDK, namely *task kernel* and *channel*. We evaluate the proposed design on a recent Altera Stratix V GX FPGA. Our results demonstrate that our proposed approach can achieve roughly 10.7X speedup over the state-of-the-art OpenCL implementation.**

I. Introduction

Recently, FPGAs have become an attractive and effective means of accelerating relational database applications, due to their energy efficiency and high throughput. A lot of fruitful research efforts have been devoted to this direction (e.g., [1], [2], [5]). However, most of those previous studies are programmed with low-level hardware description languages (HDL) like Verilog and VHDL. The programmability issues of HDL raise serious concerns on code development and maintenance. FPGA vendors such as Altera and Xilinx have started to develop OpenCL SDKs to address the programmability issues, and the research of relational databases on OpenCL-based FPGAs is still a largely open and challenging problem. As a start, we study the performance of data partitioning with the OpenCL features supported by Altera OpenCL SDK. Given an input table, the data partitioning operation is to divide the input into a number of partitions according to some partitioning criteria (for example, a hash function). Due to the random memory accesses, data partitioning is time-consuming and can become a major bottleneck for database operators. We start with the state-of-the-art OpenCL implementation for data partitioning [3], [4]. We find that, the performance is far from ideal, because of the severe lock overhead and memory stalls.

To resolve lock overheads and memory stalls, we develop a simple yet efficient multi-kernel approach to leverage two emerging features in Altera OpenCL SDK, namely *task kernel* and *channel*. With the channel (FIFO buffer), data partitioning is designed with a producer-consumer paradigm. We evaluate the proposed design on an Altera FPGA. Our results demonstrate that our proposed approach can achieve roughly 10.7X speedup over the state-of-the-art OpenCL implementation.

II. Design and Implementation of Partitioning

The proposed architecture of multi-kernel partitioning comprises of one *Data_in* kernel in the producer stage and multiple *Data_out* kernels in the consumer stage to balance the throughput unbalance between producer stage and consumer stage. Each *Data_out* kernel has the dedicated channel to receive the tuple from the *Data_in* kernel. Another potential benefit of the multi-kernel design is that the local memory (on-chip memory blocks) can be distributed into each *Data_out* kernel. It is very useful since the local memory is roughly distributed in the FPGA chip. We have also developed a cost model to determine the suitable number of *Data_out* kernels so that the throughput balance between the producer and consumer stages is achieved and the overall performance is greatly improved.

III. Results and Conclusion

Our experiments were conducted on the FPGA board is the Terasic's DE5-Net board which includes 4GB 2-bank DDR3 device memory, and an Altera Stratix V GX A7, with the Altera OpenCL SDK version 14.0. The input data is a relation (i.e., table) with the tuple format of <key, payload>, which is a common storage format in column-based databases. Both keys and payloads are 4-byte randomly generated integers.

The proposed data partitioning approach is implemented based on task kernels and channels in Altera OpenCL SDK 14.0. Our experimental result shows that the performance of partitioning is greatly improved with the proposed optimizations, and our proposed multi-kernel approach is 10.7 times faster than the original lock-back implementation, and the performance scales well for increasing data sizes (16MB, 32MB, 64MB, 128MB, and 192MB).

IV. Acknowledgement

This work is in part supported by a MoE AcRF Tier 2 grant (MOE2012-T2-1-126) in Singapore.

References

[1] J. Casper and K. Olukotun. Hardware acceleration of database operations. In *FPGA*, 2014.

[2] C.Dennl, D.Ziener, and J.Teich. Acceleration of sql restrictions and aggregations through fpga-based dynamic partial reconfiguration. In *FCCM*, 2013.

[3] J. He, M. Lu, and B. He. Revisiting co-processing for hash joins on the coupled cpu-gpu architecture. *Proc. VLDB Endow.*, 2013.

[4] J. He, S. Zhang, and B. He. In-cache query co-processing on coupled cpu-gpu architectures. *Proc. VLDB Endow.*, 2015.

[5] L. Woods, Z. Istvn, and G. Alonso. Ibex-an intelligent storage engine with support for advanced sql off-loading. In *VLDB*, 2014.