In-Memory Data Analytics on Coupled Architectures

Bingsheng He

1 Background on Coupled Architectures

Modern CPU has evolved into powerful processor with multi-core and high frequency processing capabilities. However, due to the manufacturing limitations and thermal issues, it becomes increasingly difficulty to further improve the computing capability of CPU. Moreover, the increasing complex applications require processors to be able to process workloads with varying computing complexity and pattern. Accelerators are designed to bridge the gap between CPU and requirements in reality. Graphics Processing Unit (GPU) is specifically designed for manipulating graphic and image processing. Field Programmable Gate Array (FPGA) is specialized with configurable capability to deliver combinational functions customized by users.

1.1 Why coupled

GPUs have emerged as promising hardware co-processors to speedup various applications, such as scientific computing [33] and database operations [17]. With massive thread parallelism and high memory bandwidth, GPUs are suitable for applications with massive data parallelism and high computation intensity.

One hurdle for the effectiveness of CPU-GPU co-processing is that the GPU is usually connected with the CPU with a PCI-e bus, as illustrated in Figure 1 (a). On such discrete architectures, the mismatch between the PCI-e bandwidth (e.g., 48 GB/sec) and CPU/GPU memory bandwidth (e.g., dozens to hundreds of GB/sec) can offset the overall performance of CPU-GPU co-processing. As a result, it is preferred to have coarse-grained co-processing to reduce the data transfer on the PCI-e bus. Moreover, as the GPU and the CPU have their own memory controllers and caches (such as L2 data cache), data accesses are in different paths.

Recently, vendors have released new coupled CPU-GPU architectures, such as the AMD APU and Intel Ivy Bridge. An abstract view of the coupled architecture is illustrated in Figure 1 (b). The CPU and the GPU are integrated into the same chip. They can access the same main memory space, which is managed by a unified memory controller. Furthermore, both processors share the L2/L3 data cache, which potentially increases the cache efficiency.

Table 1 gives an overview of AMD APU A8-3870K, which is used in our study. We also show the specification of the latest AMD GPU (Radeon HD 7970) in discrete architectures for comparison. The GPU in the coupled architecture has a much smaller number of cores at lower clock frequency, mainly because of chip area limitations. On current AMD APUs, the system memory is further divided into two parts, which are host memory for the CPU and device memory for the GPU. Both of the two memory spaces can be accessed by either the GPU or the CPU through the zero copy buffer. This study stores the data in the zero copy buffer to fully take advantage of co-processing capabilities of the coupled architecture. The current zero copy buffer is relatively small, which can be relaxed in the future coupled CPU-GPU architecture [2].

There have been some studies (like MapReduce [6] and key-value stores [20]) on the coupled architecture. Most studies have demonstrated the performance advantage of the coupled architecture over the CPU-only or the GPU-only algorithm. This study focuses on hash joins, and goes beyond existing studies [6, 20] in two major aspects. Firstly, we revisit the design space of hash joins on the coupled architecture and develop a cost model that can guide the decisions for co-processing. We conjecture that the design space and cost models are also applicable to those studies [6,20]. Secondly, we quantitatively show the advantage of co-processing on the coupled architecture, in comparison with that on the discrete architecture.

1.2 Existing products

1.3 Abstract model

2 Related Work

2.1 Coupled Architectures

2.1.1 Databases

2.1.2 Data Processing

2.1.3 Architecture Design

2.2 In-memory Databases

2.2.1 CPU-based Databases

2.2.2 GPU-based Databases

3 Tentative Proposals

3.1 Pipelined Design

3.2 In-cache Design

4 Author's Bio

Dr. He Jiong received his bachelor degree from East China University of Science and Technology (2007 2011) in China. After that, he got the PhD degree from Nanyang Technological University (2011 2016) in Singapore, where his major research interests focus on High Performance Computing (HPC) and database systems. In particular, he interests in applying HPC techniques (like GPGPU) to accelerate the performance of relational query processing and other data-related applications. He has conducted a comprehensive and systematic study on combining HPC and relational database systems by exploiting the power of emerging new hardware such as coupled CPU-GPU architectures, which has inspired the database community in building high-performance and energy-efficient data processing systems based on the next generation hardware. In 2013, he was awarded with the VLDB travel grant for his research about hash joins on coupled CPU-GPU architectures. Moreover, his research work has been published in prestigious international proceedings such as VLDB/PVLDB and ACM SIGMOD.

5 Author's CV