Skyway: Accelerate Graph Applications with a Dual-Path Architecture and Fine-Grained Data Management

Mo Zou (邹沫), Ming-Zhe Zhang (张明喆), Ru-Jia Wang (王茹嘉),
Xian-He Sun (孙贤和), Xiao-Chun Ye (叶笑春),
Dong-Rui Fan (范东睿), Zhi-Min Tang (唐志敏)

Research Objectives

- This paper is about graph processing acceleration.
- Motivated by the three facts that 1) not all data arrays in graph processing show poor locality; 2) low data reuse in the cacheline; and 3) under-utilized memory system, we aim to optimize current memory hierarchy.
- In this work, we propose to enhance graph processing applications by leveraging fine-grained memory access patterns with a dual-path architecture on top of existing software-based graph optimizations.
- The proposed Skyway architecture is able to improve the overall performance by reducing the memory access interference and improving data access efficiency with a minimal overhead.

Research Method

- Skyway architecture consists of two primary components:
  - At the cache side, we modify the conventional cache hierarchy to include a direct path with a small Property Buffer (PBuf), which supports fine-grained random memory accesses.
  - At the main memory side, we revisit the memory array and row buffer design to include the Duplication Row (DRow) to mitigate row buffer conflicts. DRow provides extra buffer space next to the row buffer to eliminate the interference between multiple accesses.
- As a result, Pbuf and DRow can work together seamlessly to improve the utilization of the overall memory system bandwidth without breaking the data locality.

Research Results

- Skyway improves the performance by 29% on average and up to 86% in the best case over the LRU baseline.
- Compared to the state-of-the-art GRASP, Skyway provides an average performance improvement of 23%.
- Skyway improves the DRAM bandwidth utilization by 2.13x on average and up to 5.87x in the best case.
- Skyway adds an extra storage overhead of 2.6% to LLC and 0.02% to DRAM.
- We leave the work of identifying input graphs and employing Skyway based on graph features automatically in the future work.

Research Conclusions

- Graph processing is important. However, the current memory hierarchical architecture does not support graph processing well due to poor locality and complex access patterns.
- We show that a graph can be represented in three data arrays, while only memory requests to the state array exhibit irregular access patterns.
- Based on such observations, we present Skyway, a data-aware hardware architecture with a fine-grained direct datapath from core to main memory and a memory-side row buffer hardware.
- In doing so, Skyway processes memory requests more efficiently by mitigating the memory access interference as well.
- On a set of graph workloads, Skyway improves application performance by 29% on average over the baseline. Skyway also outperforms existing state-of-the-art hardware optimizations.
- While Skyway is motivated by graph processing, the key idea behind the design can be extended to accelerate any applications with multiple access patterns.