CSR-based Graph Traversal Accelerator on HMC

Cheng Qian†, Bruce Childers‡, Libo Huang†(✉), Qi Yu†, Zhiying Wang†

†*National University of Defense Technology, China*

‡*University of Pittsburgh, USA*

{qiancheng, libohuang, yuqi13, zywang}@nudt.edu.cn†, childers@cs.pitt.edu‡

Introduction

- Graph traversal is adopted in a wide variety of realistic scenarios. However, because of the terrible spatial locality, graph traversal is quite time-consuming
- Conventional prefetch technology and parallel framework do not bring much benefit. Fortunately the well-defined structure graph structure leaves chance for explicit prefetchers. The only problem is to accelerate this process
- We proposed **CGAcc**, a CSR-based [1] graph traversal accelerator on HMC, which deploys three prefetchers working in a pipeline style cooperatively to reduce transaction latency

CGAcc

- CGAcc acts like a master rather than a slave, which means what CPU side needs to do is only send a start request. After the request is accepted by CGAcc, it will continue fetch data until all nodes are accessed
- CGAcc needs the support of several external modules
  - Activation Register, used as a trigger to wake CGAcc up; Continual Register, used to store the current maximum start node index
  - On-chip Caches are used as buffers for vertex, edge and visited array to reduce transaction latency
  - Prefetcher groups work like cycle in pipeline way. Except access registers periodically, VEP receives work list info from VSP and send vertex data to EP; EP receives vertex info from VEP and send edge data to VSP; VSP receives edge info from EP and send new node index data to VEP

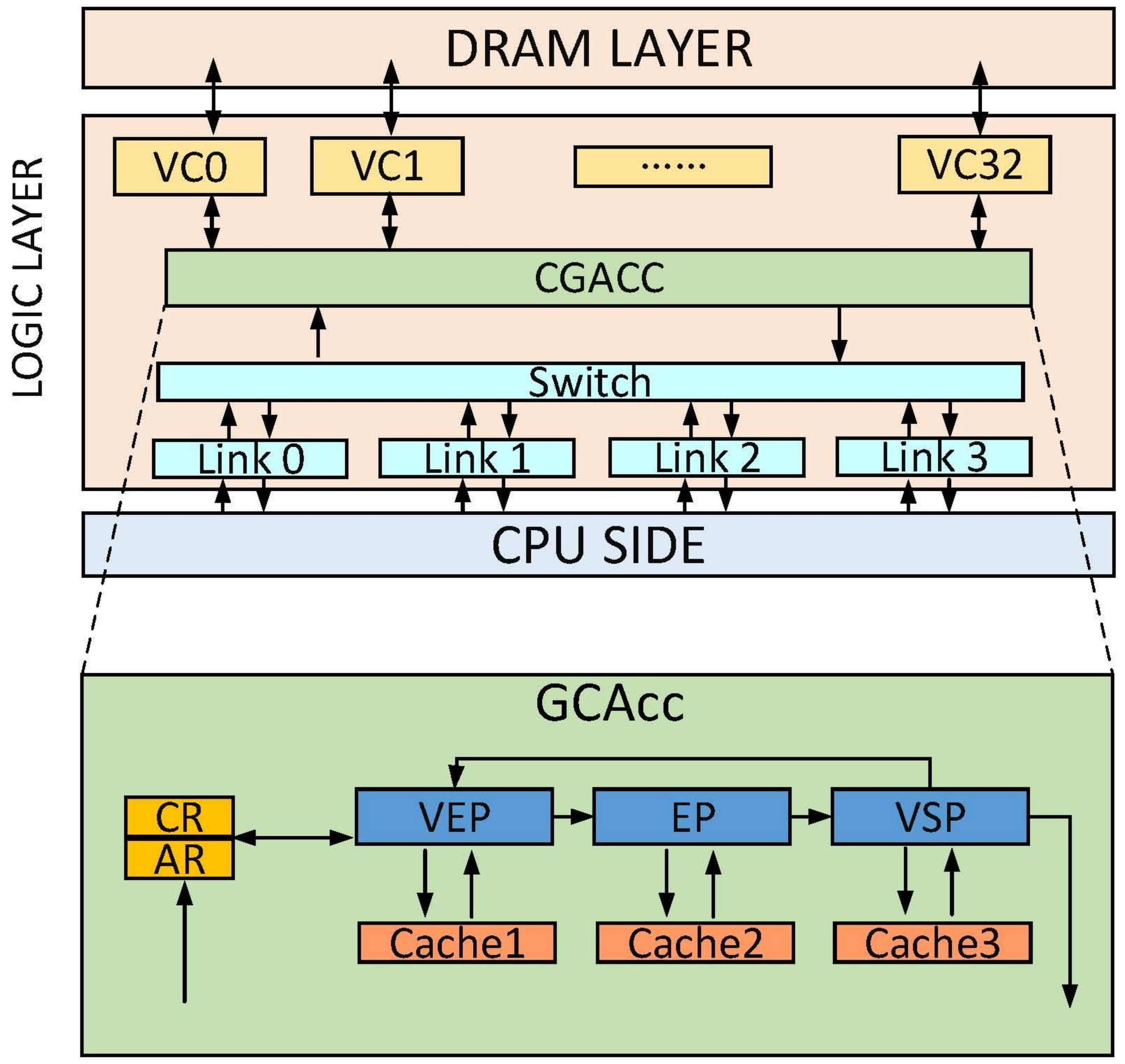

Figure 2. The architecture of HMC and CGAcc

CSR-based Graph Traversal

- A work node index leads to the corresponding two locations (index and index + 1) in vertex array. These two values which fetched from vertex arrays illustrate the range that the data should take from edge array
- Then edge array will be accessed. Similarly, the edge data will also be used as the index for the visited array
- Finally, the visited array will be accessed to determine whether this extended node has been accessed or not, and if not, this node will be pushed into the work list as a new node

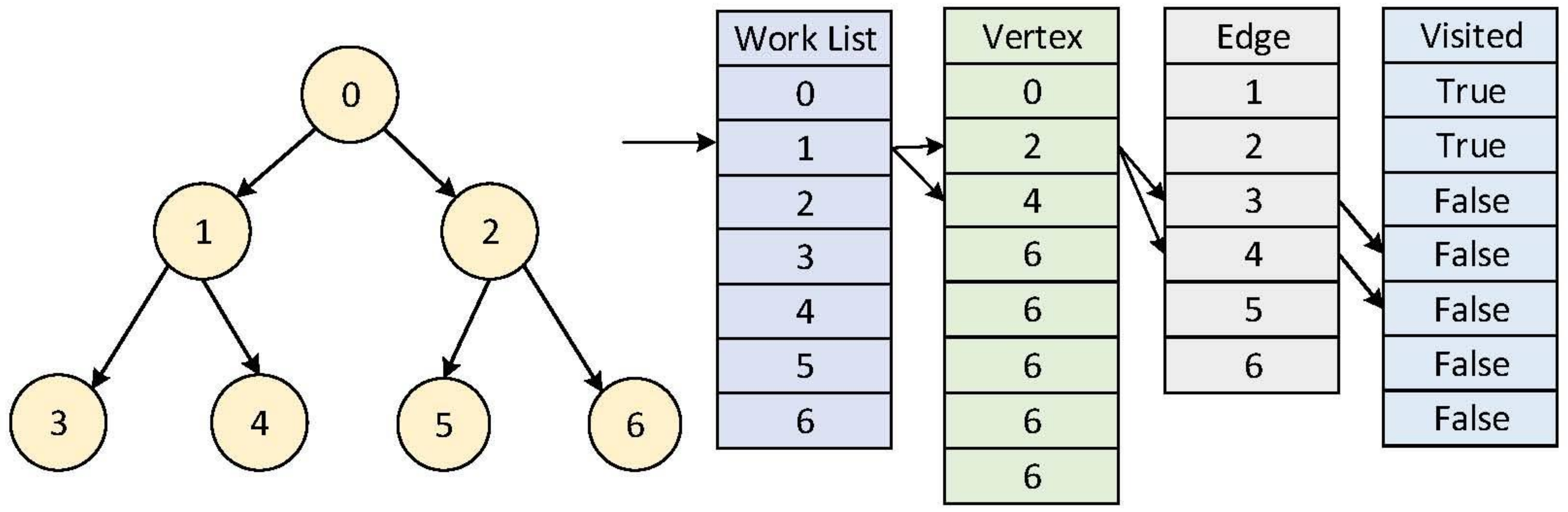

Figure 1. CSR graph traversal work flow.

Results

- We perform the experiments on the cycle-level CasHMC [2] simulator, and make the necessary micro-architectural modifications. We evaluate on several workloads from Graph-BIG [3]
- The results are shown in Figure 3. The average speedup can reach 2.8X. Respectively, for real-world workloads, the average speedup is 2.84X while the speedup shows as 2.70X for synthetic workloads

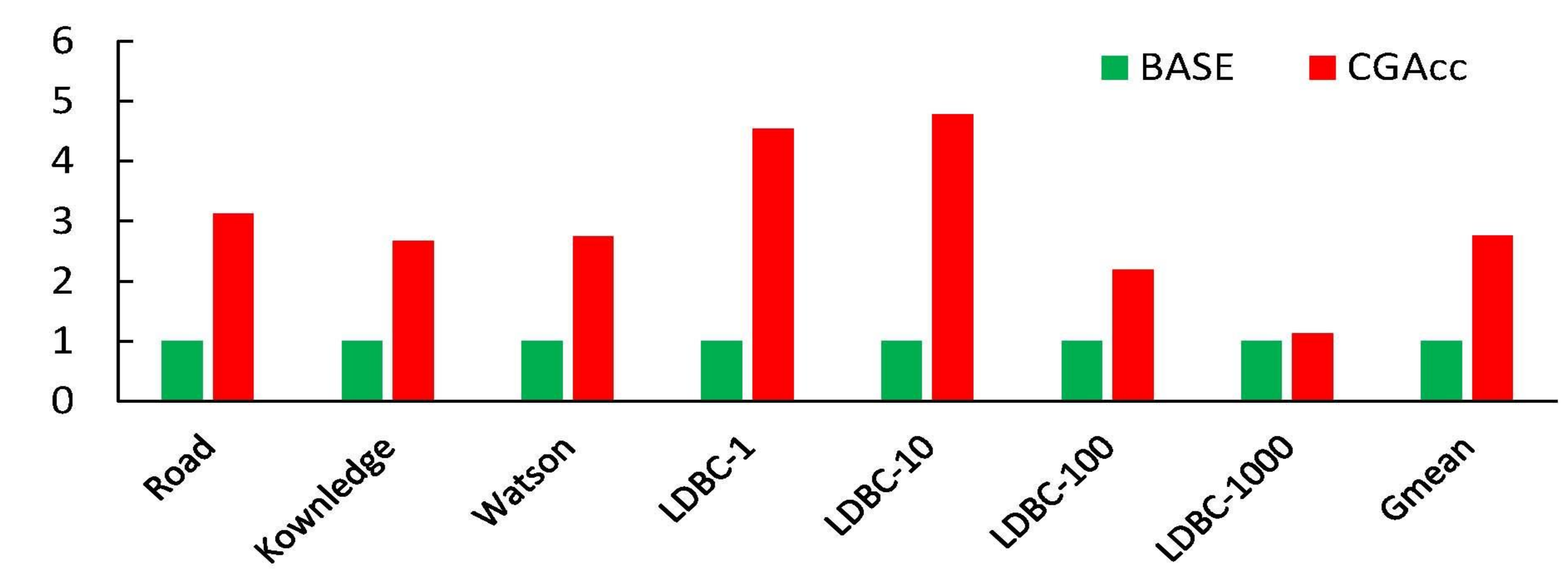

Figure 3. Comparison of performance on several graph workloads.

References

1. E. F. DAzevedo, M. R. Fahey, and R. T. Mills, "Vectorized sparse matrix multiply for compressed row storage format," in International Conference on Computational Science. Springer, 2005, pp. 99–106.
2. D.-I. Jeon and K.-S. Chung, "Cashmc: A cycle-accurate simulator for hybrid memory cube," IEEE Computer Architecture Letters, 2016.
3. Nai, Lifeng, et al. "GraphBIG: understanding graph computing in the context of industrial solutions." High Performance Computing, Networking, Storage and Analysis, 2015 SC-International Conference for. IEEE, 2015.