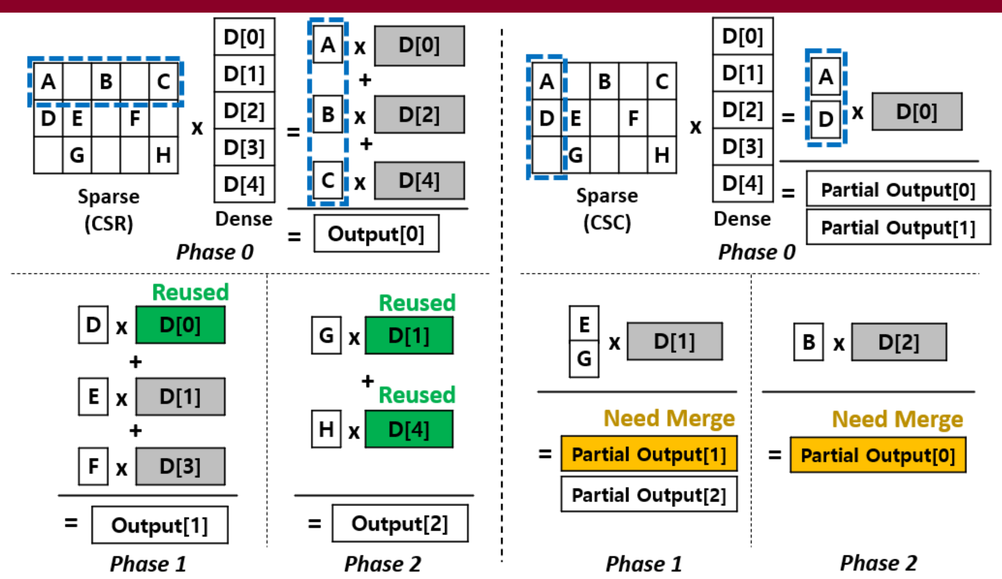
HyMM: A Hybrid Sparse-Dense Matrix Multiplication Accelerator for GCNs

Hunjong Lee[o§], Jihun Lee[o], Jaewon Seo[o], Yunho Oh[o], Myung kuk yoon*, Gunjae Koo[o]
[o]Korea University, *Ewha Womans University, [§]hunjong@korea.ac.kr

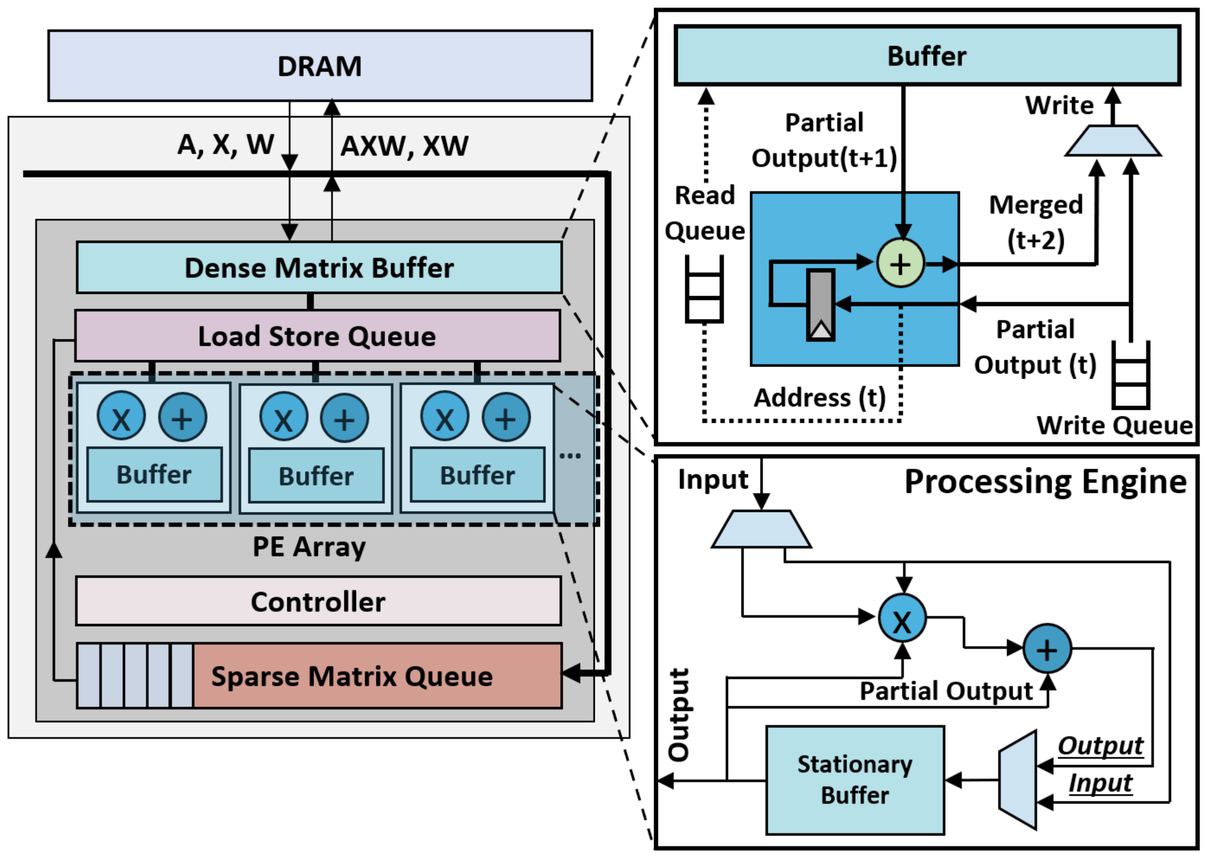
Motivation

- Graph Characteristics
  - Most graph adjacency matrices follow a power-law distribution
- Inefficiency of Existing GCN Accelerators
  - Struggle with rigid dataflows(Row-wise product, Outer product)
  - Missing graph characteristics that can utilize data locality
- We enhance the performance of GCN inference by using hybrid dataflow suitable for graph characteristics

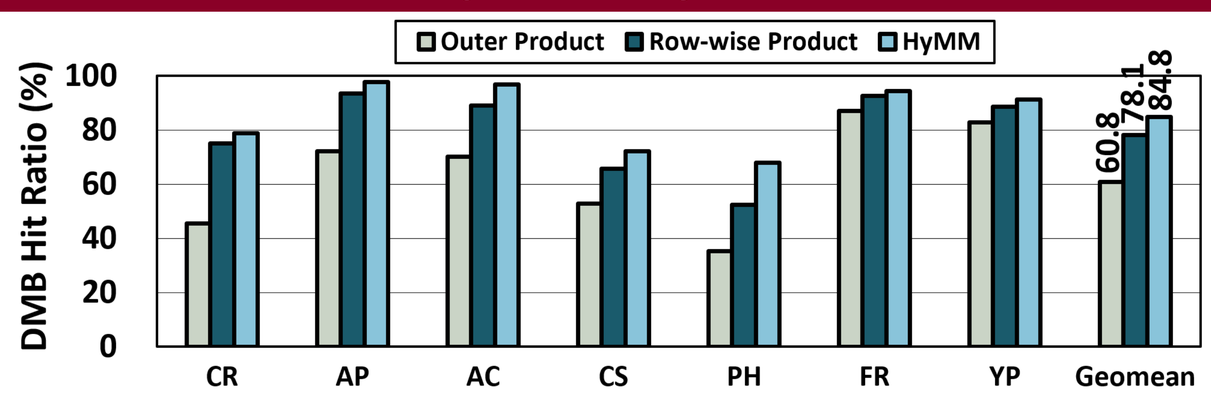
SpDeMM Dataflows



HyMM Architecture



On-chip Memory Hit Ratio

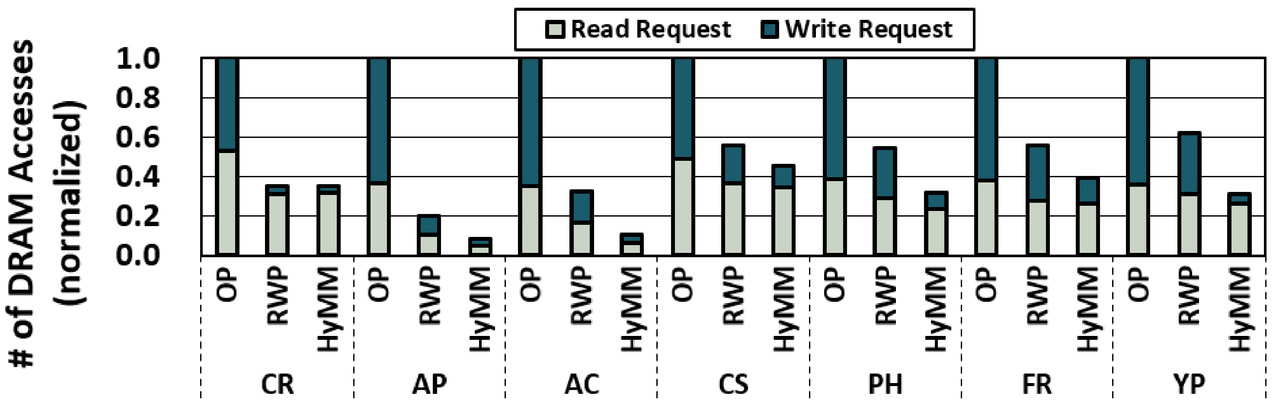


Evaluation Setup

Component	*Configuration*	*Area(mm^2)*	
		7nm	*40nm*
PE Array	16 MAC	0.006	0.21
DMB	256 KB	0.077	2.39
SMQ	16 KB	0.008	0.254
LSQ	128 Entries, 68B/Entry	0.009	0.292
Others	-	0.004	0.129
Total	-	*0.106*	*3.215*

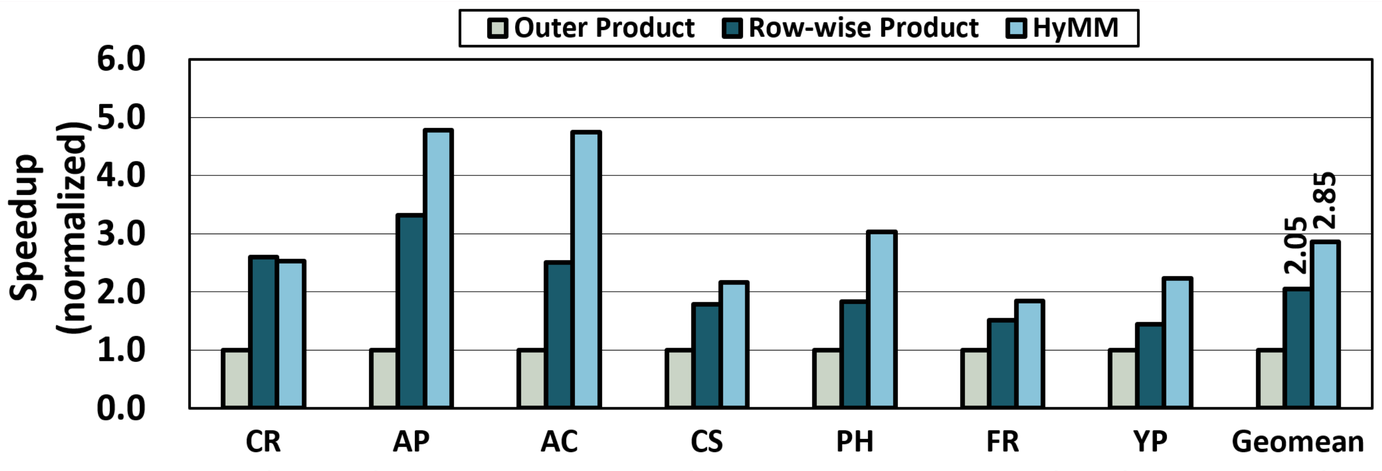
- Python based cycle-accurate simulator
- We use Synopsys Design Compiler with ASAP 7nm and CACTI 7.0

DRAM Access Breakdown



- By exploiting data locality efficiently with the hybrid dataflow architecture, **HyMM** can effectively reduce off-chip accesses

Performance Comparison



- Row-wise product reduces execution time by up to 2.05x compared to the outer product on average
- **HyMM** achieves a maximum performance improvement of 4.78x over the outer product for AP
- **HyMM** effectively employs a divide-and-conquer strategy by appropriately partitioning the graph datasets to apply each dataflow architectures selectively

Conclusion

- **HyMM** – a GCN accelerator with hybrid SpDeMM dataflow
- With low-cost pre-processing, hybrid dataflow mitigate drawbacks of each dataflows
  - Maximizing on-chip memory hit ratio with tiling scheme
  - Exploit near memory accumulator for partial outputs
- **HyMM** achieves 4.78x and 3.09x improvements in performance and utilization compared to baseline dataflows

This work was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (MSIT) of the Government of South Korea (NRF-2021R1C1C1012172)

Computer System Architecture Lab