Graph Neural Networks Acceleration with Adaptive Dataflow Architectures & FPGA

The University of Manchester

Advanced Processor Technologies Research Group

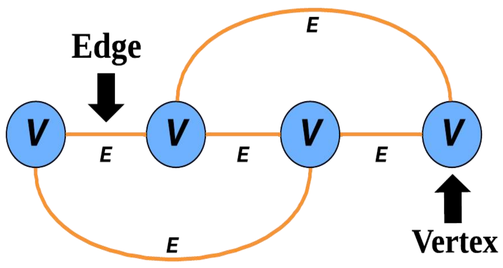
Zheyu Liu

Supervisor: *Christos-efthymios Kotselidis*

I. Introduction

• What is Graph?

Graph is a non-Euclidean data structure used to represent complex relationships between entities, consisting of numerous vertices and edges.

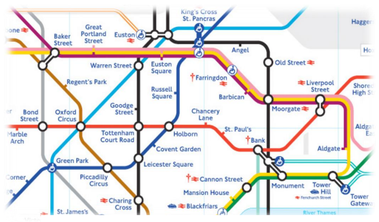


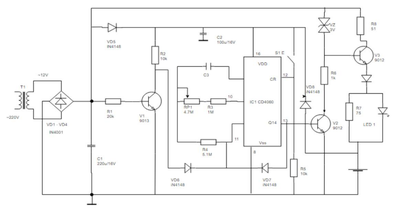
G = (V, E)

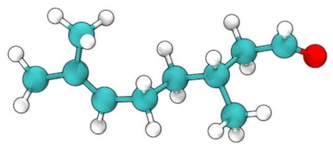
Edge: Relationship

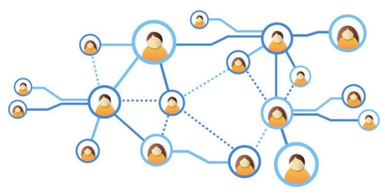
Vertex: Entity

• Graph Examples









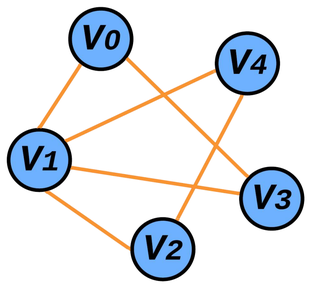
• Graph in Memory

Vertex in a graph contains its unique information, such as a person's height, weight and hobbies. All the features of a vertex can be represented as a numercial ***Embedding Vector*** with N elements.

Embedding Vector

V_0	$V_{(0)}F_{(0)}$	$V_{(0)}F_{(1)}$	$V_{(0)}F_{(2)}$	···	$V_{(0)}F_{(N-1)}$
V_1	$V_{(1)}F_{(0)}$	$V_{(1)}F_{(1)}$	$V_{(1)}F_{(2)}$	···	$V_{(1)}F_{(N-1)}$
V_2	$V_{(2)}F_{(0)}$	$V_{(2)}F_{(1)}$	$V_{(2)}F_{(2)}$	···	$V_{(2)}F_{(N-1)}$
⋮				⋮	
V_{M-1}	$V_{(M-1)}F_{(0)}$	$V_{(M-1)}F_{(1)}$	$V_{(M-1)}F_{(2)}$	···	$V_{(M-1)}F_{(N-1)}$

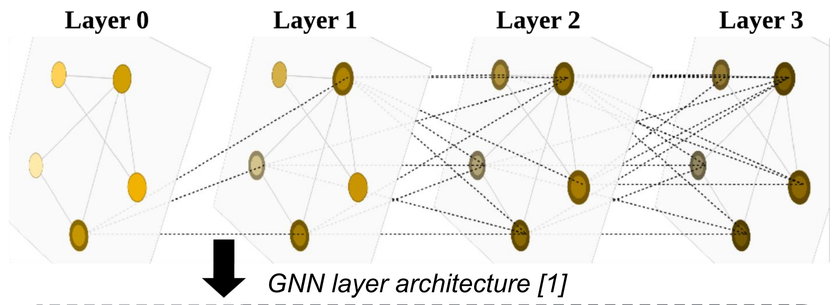
Connectivity information of vertices is usually stored in form of an ***Adjacency Matrix***.



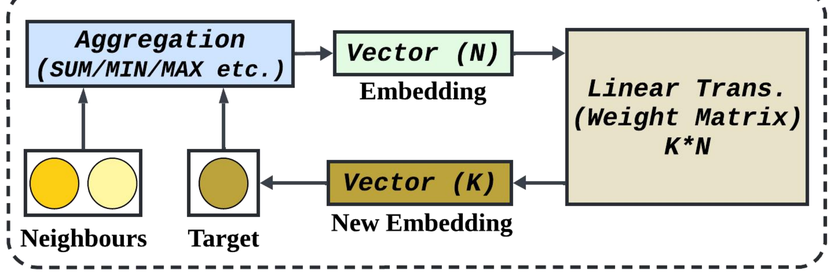
	V0	V1	V2	V3	V4
V0	0	1	0	1	0
V1	1	0	1	1	1
V2	0	1	0	0	1
V3	1	1	0	0	0
V4	0	1	1	0	0

• Apply Neural Network to Graph

The message passing framework establishes the data flow in a graph neural network (GNN) which primarily comprises ***Aggregation*** and ***Linear Transformation***.

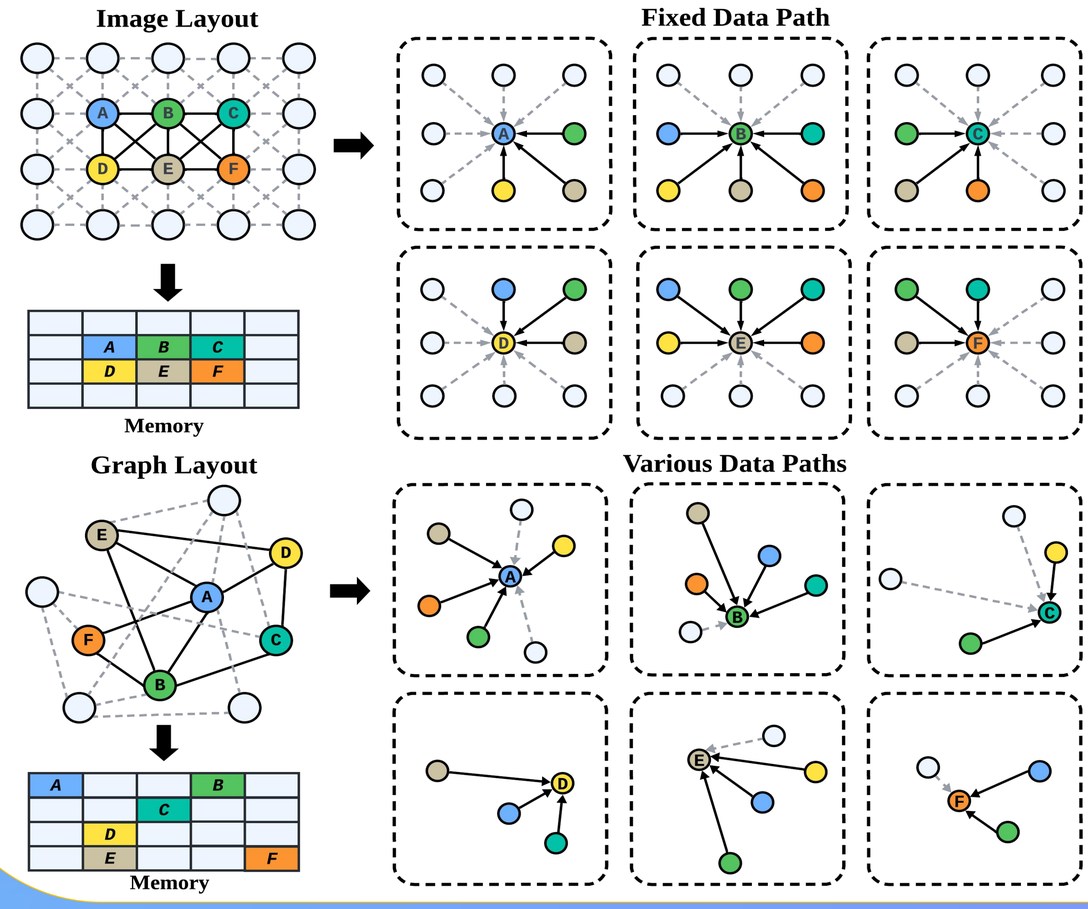


GNN layer architecture [1]

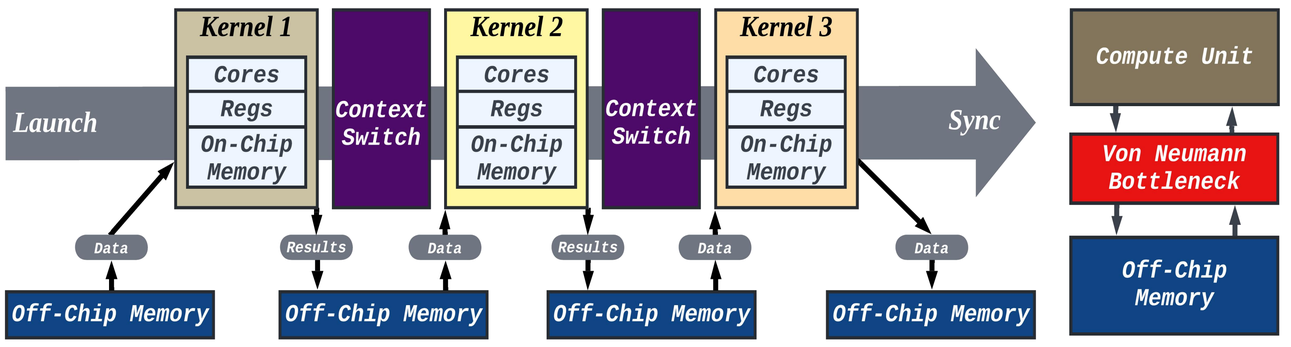


GNNs are showing promising applications in diverse domains such as traffic prediction, fraud detection and drug discovery.

II. Problems of Running GNNs on GPU



GPUs are equiped with cache and very high bandwidth off-chip memory to alleviate the Von Neumann bottleneck, enabling faster data transfer between the processing units (core) and memory.



Operators	L2 Hit % (A100)
torch.gather	62.75
torch.index_add_	82.09
torch.index_select	70.02
torch.matmul($\leq 10^8$ el.)	88.58
torch.matmul($> 10^8$ el.)	89.56
torch.transpose	93.59

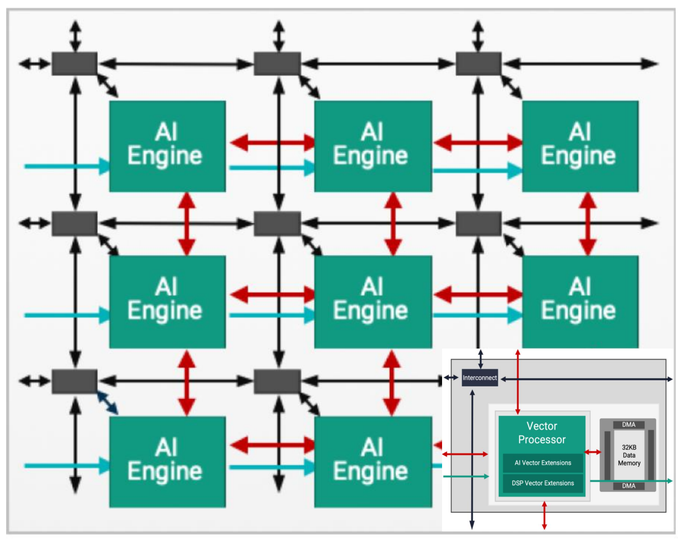
A100 Cache Hit Rate on ML Micro-operations [2]

Due to frequent random memory accesses, the cache hit rate for operations related to GNN is significantly reduced.

- ***Sparse Connection Among Vertices***
- ***Irregular Memory Access Pattern***
- ***Unbalanced Workload***
- ***Complex Vertex Dependency***
- ***Hard to Batching***

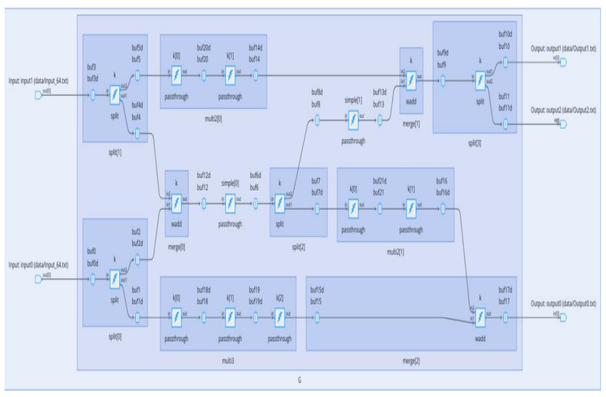
III. GNNs on Reconfigurable Archiecture Acceleration

• Xilinx Adaptive Dataflow Architecture



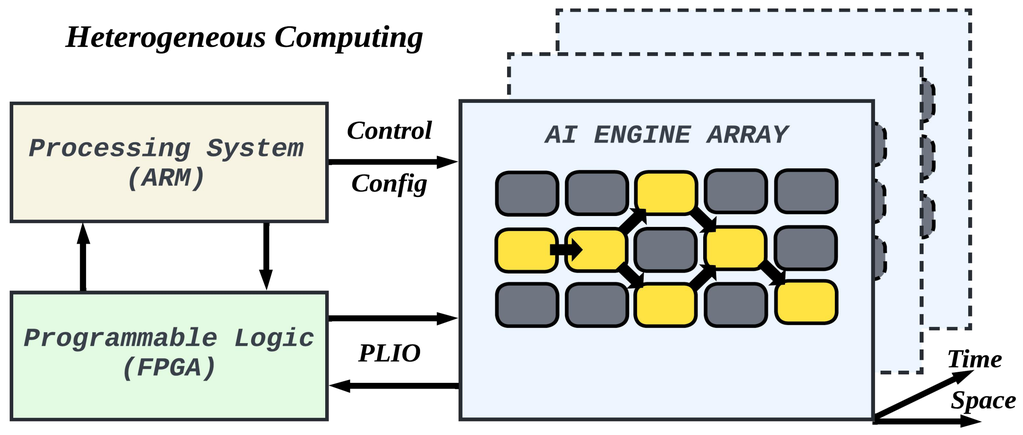
Xilinx AI Engine Array [3]

- ***Data-Driven Computing***
- ***Implicit Parallelism***
- ***Distributed On-Chip Memory***
- ***Reconfigurable SIMD Pipeline***
- ***Graph based Programability***



Dataflow Graph View [4]

• GNN Adaptive Acceleration



FPGA is a fine-grained reconfigurable architecture capable of efficiently performing data preprocessing tasks and implementing specialized hierarchical caches to hide memory access latency.

The PS unfolds GNN computation along both temporal and spatial dimensions, dynamically configuring new computation graphs.

[1] Sanchez-Lengeling, B., Reif, E., Pearce, A., & Wiltschko, A. B. (2021). A Gentle Introduction to Graph Neural Networks. Distill. Google Research. Retrieved from https://distill.pub/2021/gnn-intro/
[2] Hosseini, R., et al. (2023). Exploring the Use of Dataflow Architectures for Graph Neural Network Workloads. In: High Performance Computing. ISC High Performance 2023. Retrieved from https://doi.org/10.1007/978-3-031-40843-4_48
[3] AMD AI Engine Technology. Retrieved from https://www.xilinx.com/products/technology/ai-engine.html
[4] AMD. (n.d.). AI Engine Kernel and Graph Programming Guide (UG1079). Retrieved from https://docs.amd.com/r/en-US/ug1079-ai-engine-kernel-coding?tocId=vRsLf1MNvoWTQ8W~C3d_Jg