Graphiler: A Compiler for Graph Neural Networks

[1] [2]

Zhiqiang Xie[1,2], Zihao Ye[2], Minjie Wang[2], Zheng Zhang[2], Rui Fan[1]

How to program a GNN?

Message Passing

$$m_e = \phi(x_u, x_v, w_e), (u, e, v) \in \mathcal{E}$$
$$h_v = \rho(\{m_e : (u, e, v) \in \mathcal{E}\})$$
$$x_v^{new} = \psi(x_v, h_v), v \in \mathcal{V}$$

DGL DEEP GRAPH LIBRARY PyTorch geometric ...

UDF (ops = dense tensor operators):

```
def message_udf(edges):
    return ops(edges)
def aggregation_udf(messages):
    return ops(messages)
def update_udf(nodes):
    return ops(nodes)
```

Primitives (ops = graph operations):

```
def message_and_aggregate(graph):
    return ops(graphs)
```

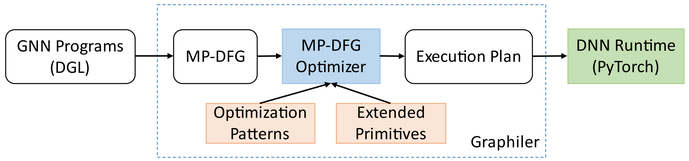
Primitives ← → **UDF**

Performance ← → Flexibility

How to both achieve **high performance** and provide a **flexible programming interface?**

Check out Graphiler!

How Graphiler works?

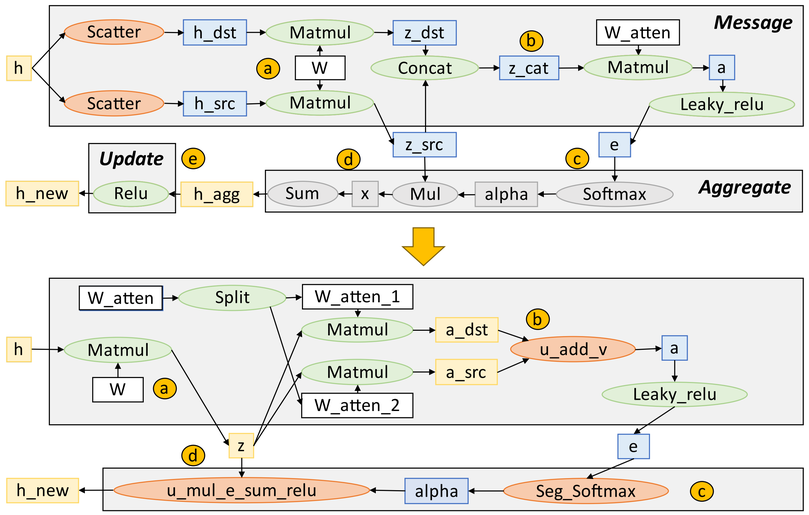


MP-DFG: Extend Data Flow Graph (DFG) using Message Passing (MP) semantics
- Data movement between nodes and edges, tensor and operator type inference

Optimization Patterns (selected):
- **Reorder:** "scatter-compute" -> "compute-scatter" to eliminate redundant computation
- **Split:** "concatenate-multiply" -> "split-multiply-sum" to enable further optimizations
- **Lowering:** Infer and replace graph operations in aggregation UDFs by extended primitives
- **Fusion:** Eliminate redundant computation and I/O by avoiding edge data materialization

Execution Plan: A proper combination of extended primitives for GNNs

MP-DFG Transformation, Graph Attention Network (GAT) as an example:



a: Reorder

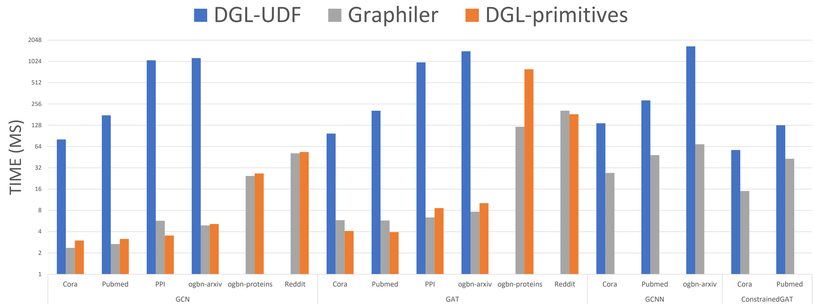
b: Split, Reorder, Fusion

c: Lowering

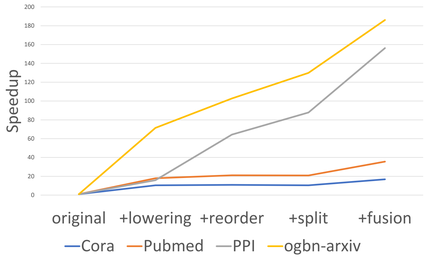
d: Lowering, Reorder, Fusion

e: Fusion

Evaluation



Overall Performance:
- Up to **232x** (GCN), **186x** (GAT) and **24x** (GCNN) on ogbn-arxiv dataset, **3x** (ConstrainedGAT) on Pubmed dataset faster than DGL-UDF
- Competitive to hand optimized implementation DGL-primitives
- Drastic memory saving



Breakdown Analysis (GAT):
- Lowering: Up to **70x** speedup
- Reorder and Split combined: Up to **5.5x** speedup further
- Fusion: Up to **1.7x** speedup further

Future Work

Graphiler is under active development!
- Heterogeneous GNNs
- More optimization passes
- More high performance GNN primitives
- **Your valuable suggestions are more than welcome!**

Please feel free to drop any question and comment to xiezhq@shanghaitech.edu.cn!