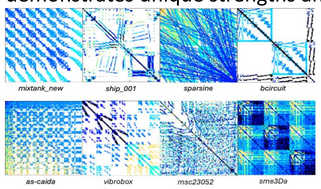
Misam: Using ML in Dataflow Selection of Sparse-Sparse Matrix Multiplication

Sanjali Yadav, Bahar Asgari
University of Maryland College Park, Department of Computer Science

Introduction: Need for Dynamic Dataflow Selection

- Sparse matrix-matrix multiplication (SpGEMM) is a crucial kernel in several applications.
- The inherent sparsity of data introduces significant performance challenges on modern computers.
- State-of-the-art hardware accelerators tackle these challenges by optimizing specific sparsity patterns through fixed dataflow schemes: Inner Product, Outer Product, and Row-wise Product.
- Each dataflow demonstrates unique strengths and weaknesses, resulting in varied performance across sparsity patterns.



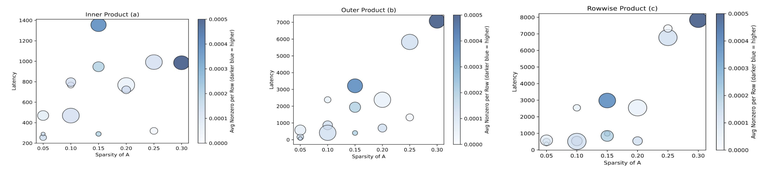
SuiteSparse matrices with diverse patterns

Design Aspect ↓ \ Dataflow →	Inner	Outer	Row-wise
Psum Granularity	✓	✗	✓
Input Format	✗	✗	✓
Index Intersection	✗	✓	✓
Input Reuse (B)	✗	✓	✗
Output Reuse (C)	✓	✗	✓
High Input Sparsity	✗	✓	✓

SpGEMM dataflow impact various design aspects.“ ✗ " indicates a potential challenge and “✓" indicates no issue.

Challenges: Selecting the Optimal Dataflow

- Prior work has started to support multiple dataflows for varied sparsity patterns. However, there are still challenges:
  - Trapezoid[1]: No built-in method to select optimal dataflow.
  - Flexagon[2]: Uses an offline profiling approach.
- Prior work lack a mechanism that dynamically predicts the optimal dataflow for diverse matrix sparsity patterns:
  1. What techniques can be employed to achieve accurate prediction?
  2. What type of data is required to inform this prediction?



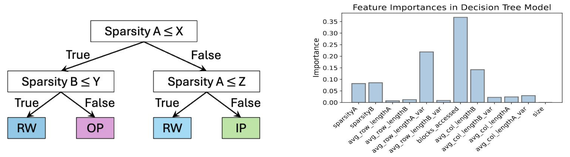
Optimal dataflow varies with features such as sparsity levels in matrices A (x axis) and B (bubble size) or nonzero per-row in matrix A (color depth)

Key insights: Using Machine Learning in Dataflow Selection

- The characteristics of this problem are well-suited to common machine learning (ML) techniques for data classification:
  given the features of the input matrices, we can categorize them into classes corresponding to different dataflows.

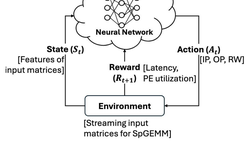
Decision Tree Model

- Supervised learning algorithm characterized by a hierarchical tree structure, which models decisions and their potential outcome.
- To identify features to represent sparsity of a matrix and generate a dataset, which labels the features with optimal dataflow.



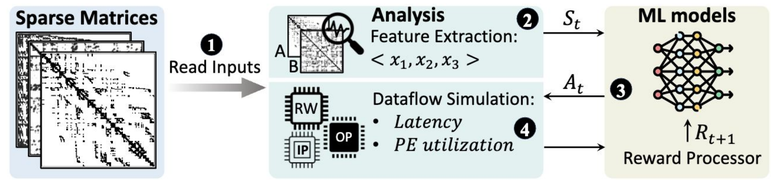
Reinforcement Learning Model

- Unsupervised ML algorithm where an agent interacts with an environment by taking actions to earn rewards.
- The agent establishes an optimal policy that associates environmental states with actions that maximize the expected cumulative reward.
- Reinforcement learning model optimizes for both performance and resource utilization.



Misam Workflow:

1. Read compressed sparse input matrices and analyze them.
2. Extract features from these inputs, which serves as states (i.e., S_t) for the RL model and input features for the decision trees.
3. Predict the optimal dataflow based on the state.
4. Simulate the chosen dataflow and gather latency and PE utilization metrics from the execution.

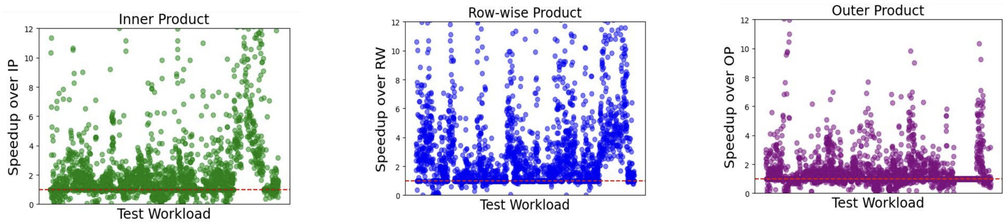


Experimental Setup

- Hardware Platform:
  - Decision Tree Model: AMD EPYC 7302 with 128GB of DRAM with a bandwidth of 204.8 GB/s
  - RL model: NVIDIA GeForce RTX 4090 with 24GB global memory with a bandwidth of 1.01 TB/s, 72 MB L2 cache and 128 KB shared memory per SM.
- Simulator: Cycle accurate C++ simulator
- Dataset: SuiteSparse Matrices and Synthetic Random Matrices

Main Results: Decision Tree Model

- Our k-fold cross-validation experiments demonstrated an **accuracy of 90%** across the dataset, translating to average (geomean) speedups of **1.93× over IP, 1.27× over OP and 2.28× over RW**.
- Small storage requirement of **512B**.
- The fine-tuned model for Trapezoid’s dataflows demonstrated **85% accuracy**.



A correct prediction achieves a speedup of one relative to the best dataflow and greater than one compared to others; a speedup below one indicates a suboptimal dataflow choice.

Main Results: Reinforcement Learning Model

- On average, achieved geometric mean speedups of **3.77× over IP, 2.48× over OP, and 4.46× over RW**.
- The RL model offers higher accuracy at the cost of a **32KB** storage requirement.

Conclusions:

- Misam introduces a lightweight novel techniques to predict the optimal dataflow for SpGEMM kernel.
- Misam identifies an optimal representation of matrix sparsity to use as input for machine learning models.
- Misam integrates its ML models with the Trapezoid hardware accelerator to demonstrate the solution's practicality.

Future Directions:

- Include additional compression formats such as DIA, COO, and ELL-Pack.
- Target deployment into a self-reconfigurable hardware system.

References:

[1] Y. Yang, J. S. Emer, and D. Sanchez. Trapezoid: A versatile accelerator for dense and sparse matrix multiplications. In 2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA),pages 931–945, Los Alamitos, CA, USA, jul 2024. IEEE Computer Society

[2] E. Qin, A. Samajdar, H. Kwon, V. Nadella, S. Srinivasan, D. Das, B. Kaul, and T. Krishna, “Sigma: A sparse and irregular gemm accelerator with flexible interconnects for dnn training,” in 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2020, pp. 58–70.