Data-Based Performance Analysis of Linear Algebra Algorithms

Variants Analysis of Linear Algebra Algorithms in the Presence of Performance Variations

Aravind Sankaran Robert van de Geijn Paolo Bientinesi

1. Background

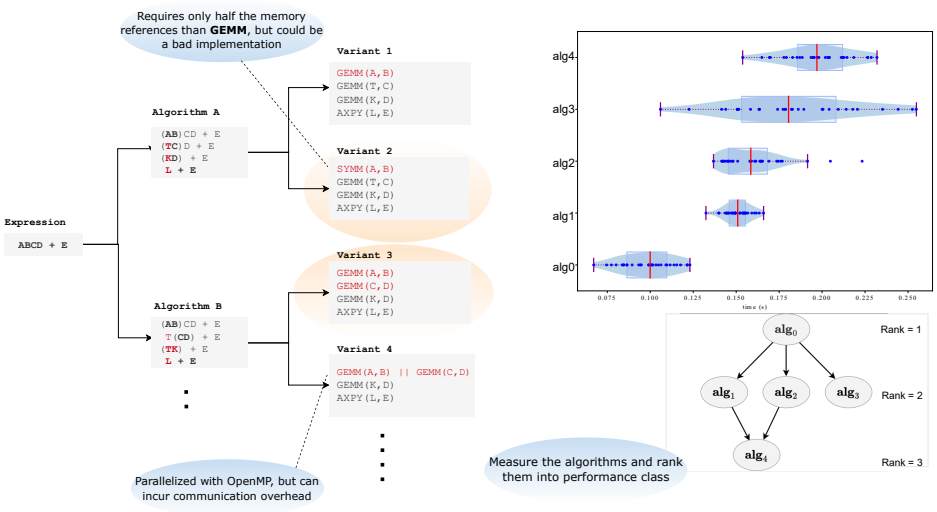
Motivation

- A mathematical expression can be computed in many alternative ways, which although equivalent from a mathematical perspective, differ in terms of performance.
- Popular computing systems such as Matlab, TensorFlow, PyTorch, etc. often choose the sub-optimal variant [1].

Objective

- Data-driven performance analysis of variant implementations and aid in selecting a fast variant.

2. Same expression, but many variants with different performance



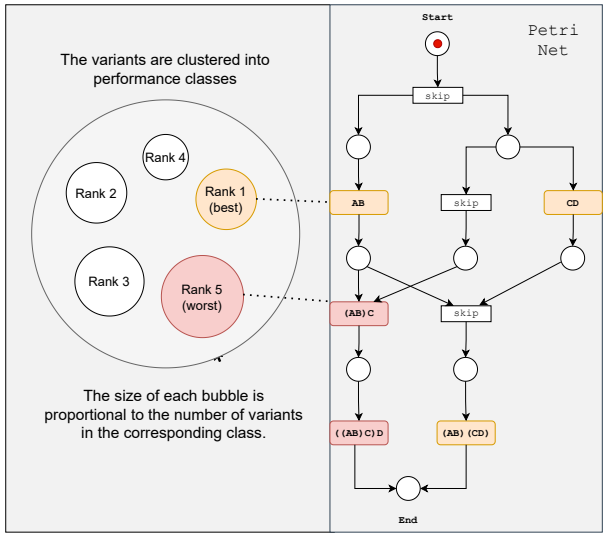
Parenthesization and Kernel choices results in different implementations of a Matrix chain.

3. Deliverables

The increasing complexity of computing systems induces variations in performance measurements. In this work, we develop methods to analyze the variants in the presence of performance variations.

- Methodologies to rank the variants into performance class, while making as little measurements as possible.
- Developed a test to determine if standard performance indicators such as FLOP count can already be used to discriminate the variants. For a given problem, this test is used to determine (and quantify) the need for machine learning based performance models [2].
- The ranks are used in the identification of the root causes of performance variations and to aid in selecting a fast variant.

4. Performance Analysis



Process model of the variants in the best and the worst ranks.

[1] **A. Sankaran**, NA. Alashti, C.Psarras and P. Bientinesi, *"Benchmarking the Linear Algebra Awareness of TensorFlow and PyTorch."*, IPDPS: iWAPT 2022.
[2] **A. Sankaran** and P. Bientinesi, *A Test for FLOPs as a Discriminant for Linear Algebra Algorithms*, IEEE SBAC-PAD 2022