A Comparative Study of Two Matrix Multiplication Algorithms Unvder Current Hardware Architectures

MIDWESTERN STATE UNIVERSITY

McCoy College of Science, Math, & Engineering, Department of Computer Science
Author: Samuel Olatunde, Eduardo Colmenares

Background Information

The Matrix Multiplication algorithm serves as a widely utilized and computationally intensive scientific kernel. A notable improvement is the Strassen variant, which produces a more asymptotically optimal $O(n^{2.81})$ algorithm compared to the standard $O(n^3)$ version. Our research aims to compare Single-Level Strassen's algorithm with Naive method, evaluating their performance on CPU and GPU architectures.

Research Questions

- **Algorithmic:** How do the performance characteristics of Naive and Strassen's approaches vary with problem sizes, and what are the trade-offs between them?
- **Architectural:** What insights can studying the GPU as a floating-point accelerator provide, particularly for matrix multiplication algorithms?

Methodology

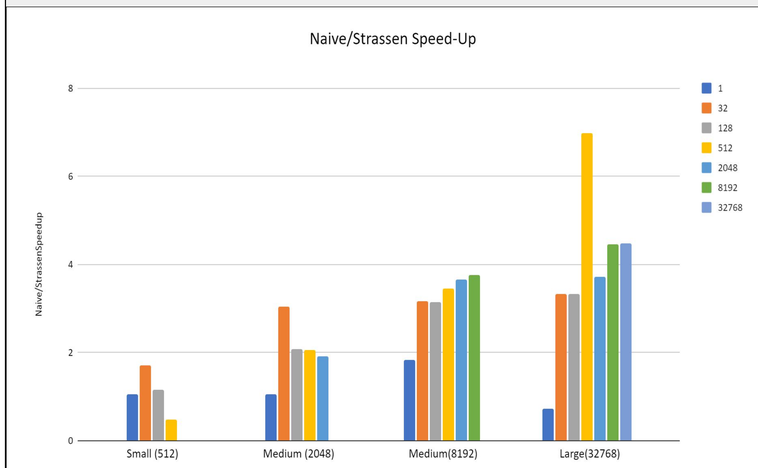
Implementation:
- Developed serial and parallel versions for both algorithms
- Utilized square, dense matrices with dimensions that are powers of 2.
- Employed single precision with integer initialization to address numerical stability.

Testing and Data Collection:
- For serial testing, follow a straightforward approach.
- For parallel testing, vary the number of threads for each data size.
- Record the runtime for each test.

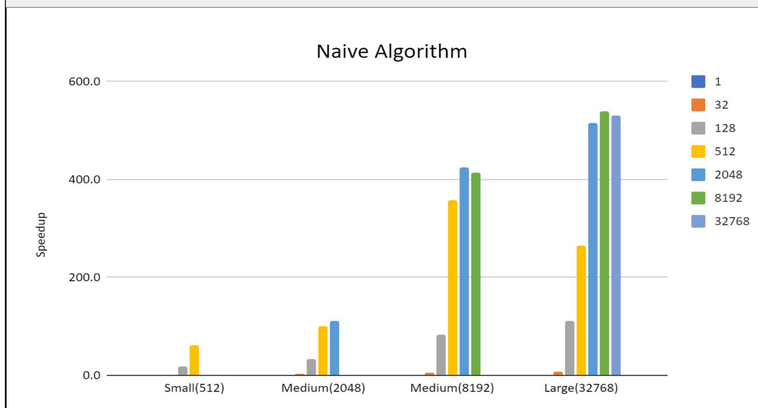
Analyze Data:
- Performance Evaluation Metrics:
  - Runtime (Primary Metric):
    - Speedup (Architectural)
    - Naive-Strassen Speedup (Algorithmic)

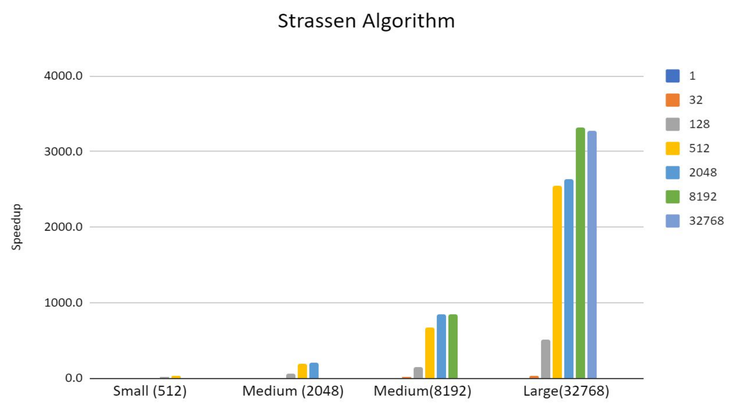
Results

Algorithmic Comparison



Architectural Comparison





Conclusions

Algorithmic:
- Serially, Strassen's algorithm outperforms up to a certain threshold (8K).
- Strassen's algorithm outperforms the naive approach as data size and thread size increase, except for an anomaly at 32K.

Architectural:
- Both algorithms consistently achieve significant speedup across all data sizes.
- GPU accelerates Strassen's algorithm more effectively than the naive approach.
- Both algorithms demonstrate sublinear speedup.

Potential Future Work:
- Implement memory-efficient multi-level Strassen's algorithm.
- Investigate numerical stability.
- Explore kernel fusion techniques.

References

- P. Pacheco and M. Malensek, *An Introduction to Parallel Programming.*, Second Edition. San Francisco Elsevier Science & Technology, 2022, pp. 17–84.
- S. G. AKL and S. D. Bruda, “PARALLEL REAL-TIME OPTIMIZATION: BEYOND SPEEDUP,” *Parallel Processing Letters*, vol. 09, no. 04, pp. 499–509, Dec. 1999.
- “Multi-Stage Memory Efficient Strassen’s Matrix Multiplication on GPU | IEEE Conference Publication | IEEE Xplore,” *ieeexplore.ieee.org*.

Acknowledgements

This research is supported by a travel grant from the McAda Graduate School at Midwestern State University.