Distributed Matrix Computation (DMaC)

A Distributed System for Matrix Multiplication

Ryan Bazzell & Jiwoon Yim

March 17, 2025

Introduction

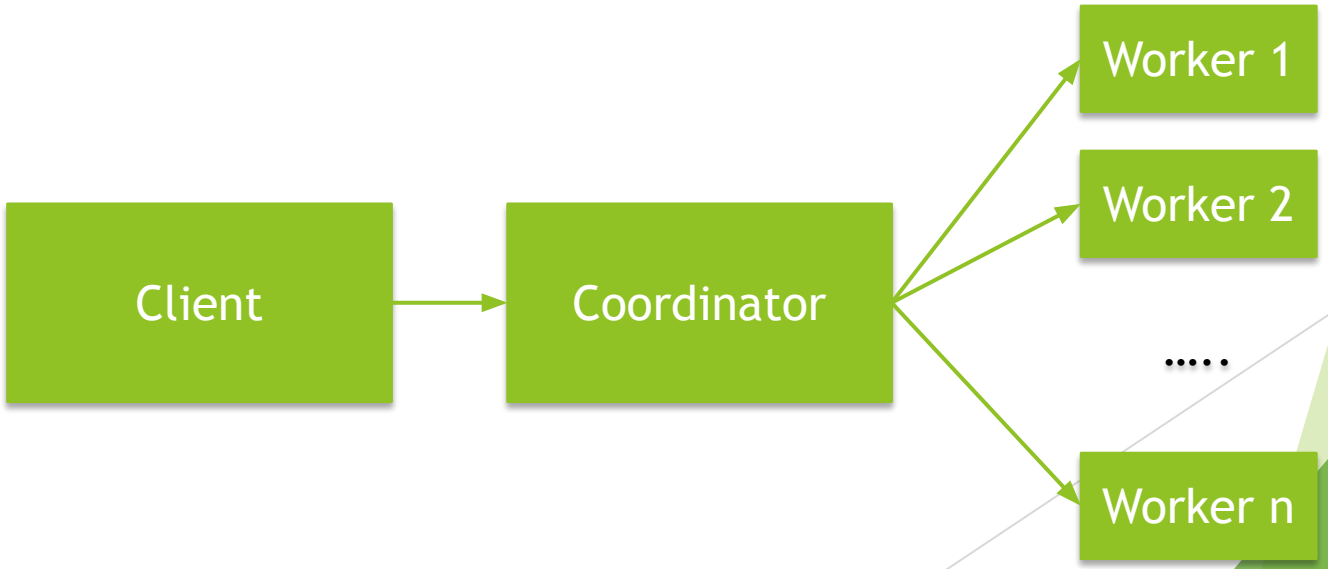
- Distributed matrix multiplication
  - Strassen's Algorithm
- Heterogeneous nodes
- Coordinator-worker architecture
- Client for User-Interface
- Scalable and transparent

Motivation

- Single-threaded shortcomings
- Time consuming large-scale matrix multiplications
- Improve performance
- Provide scalability

System Architecture

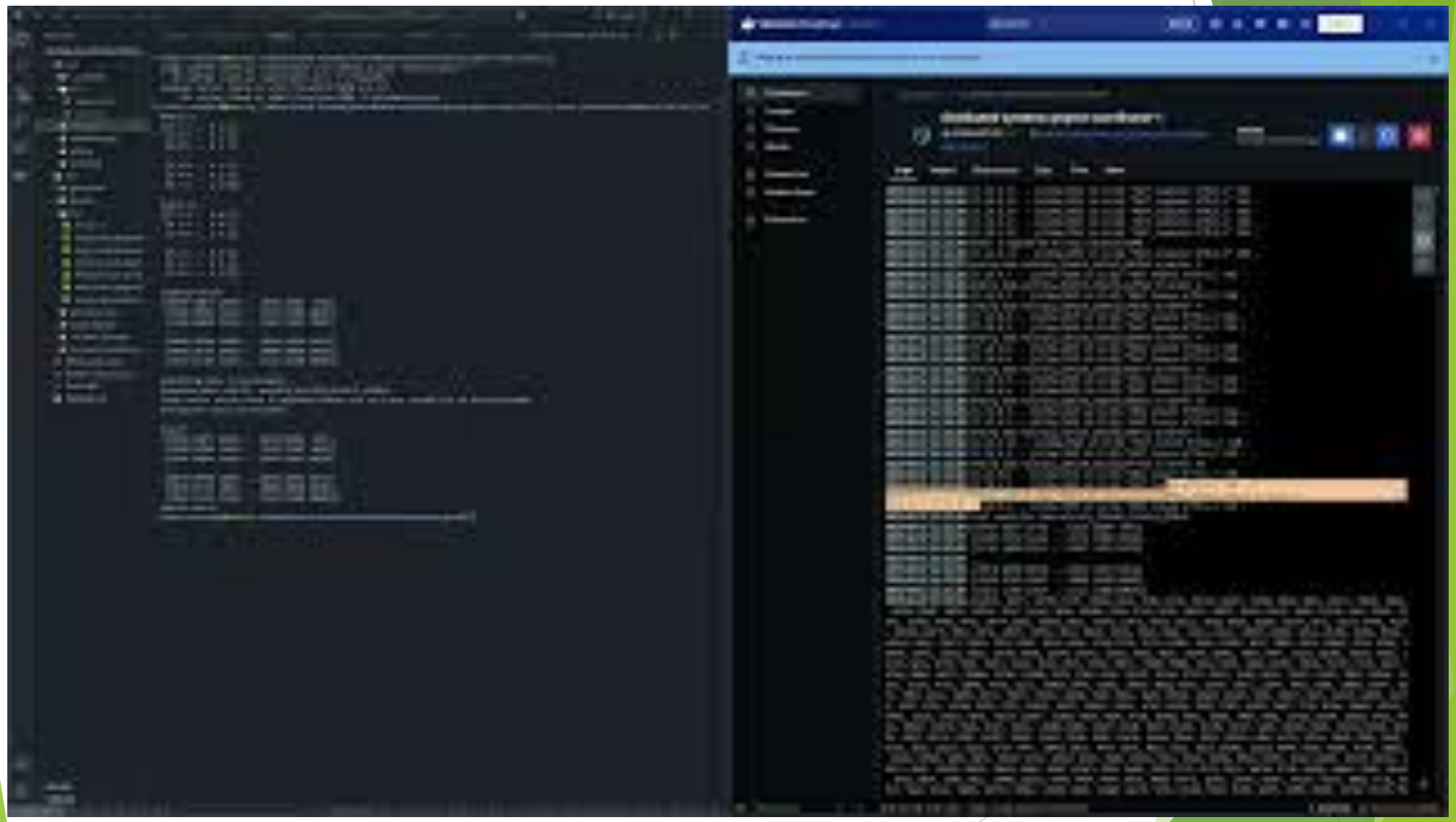
- Client: Submits jobs and verifies results
- Coordinator: Distributes tasks, collects results
- Workers: Compute using Strassen's Algorithm



Implementation Details

- Failure handling with retry mechanisms
- Python with Flask, NumPy, and threading
- Docker simulates a distributed environment
- **Strassen's Algorithm** for matrix multiplication ($O(n^{2.807})$ vs. $O(n^3)$)

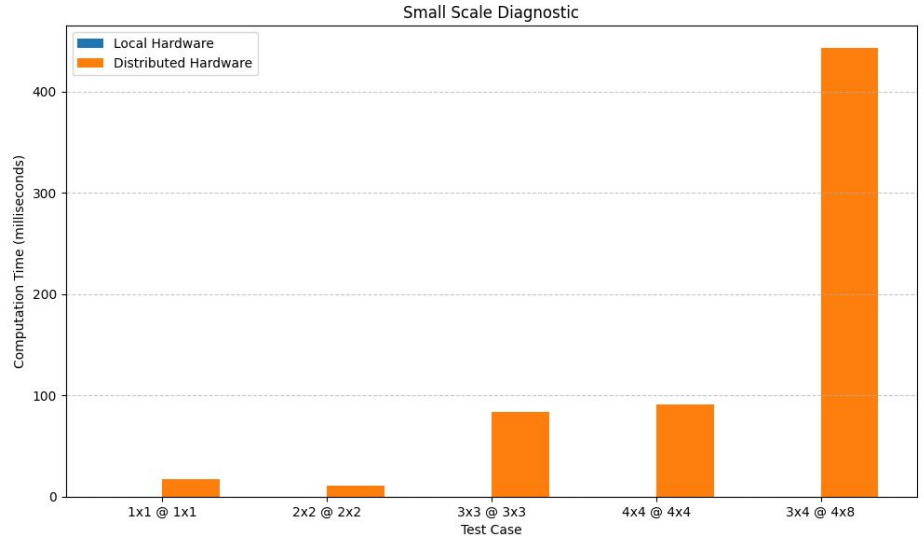
Demo

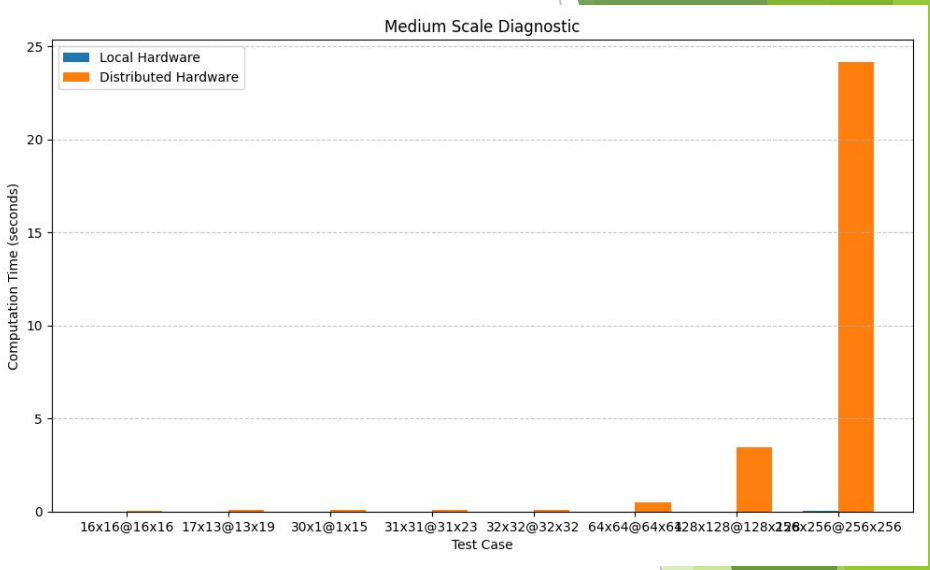


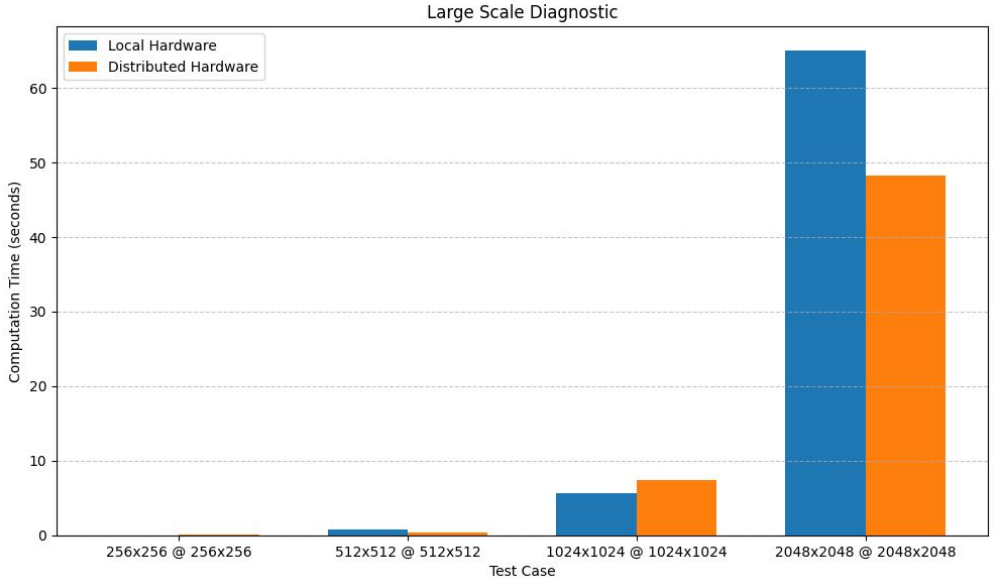
Simulation & Testing

- Performance benchmarks and test cases
  - Simulated heterogeneous setup with Docker (Alpine, Debian, Fedora)
  - Three scales: Small (4 workers), Medium (10 workers), Large (20 workers)
  - Compared local vs. distributed performance
- Metrics
  - Accuracy
  - Performance
  - Scalability
  - Transparency

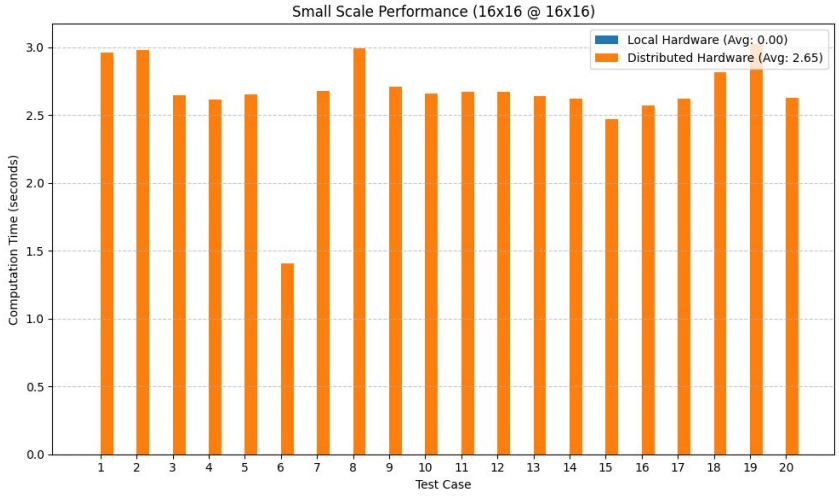
Results - Diagnostics

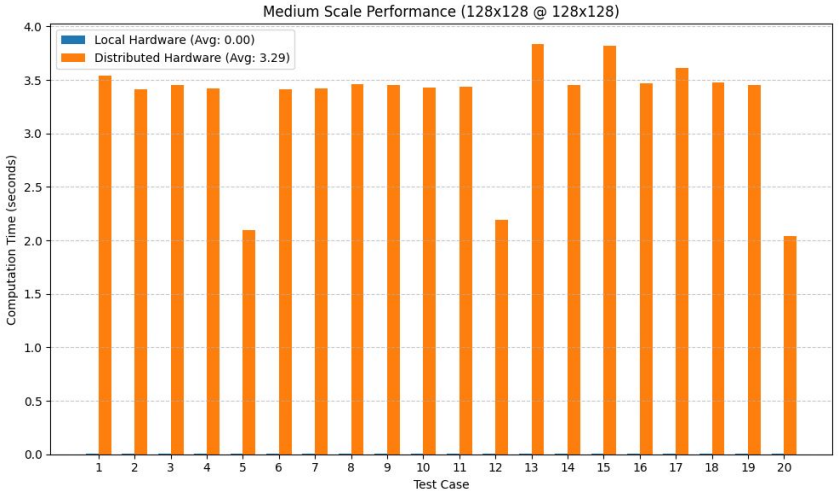


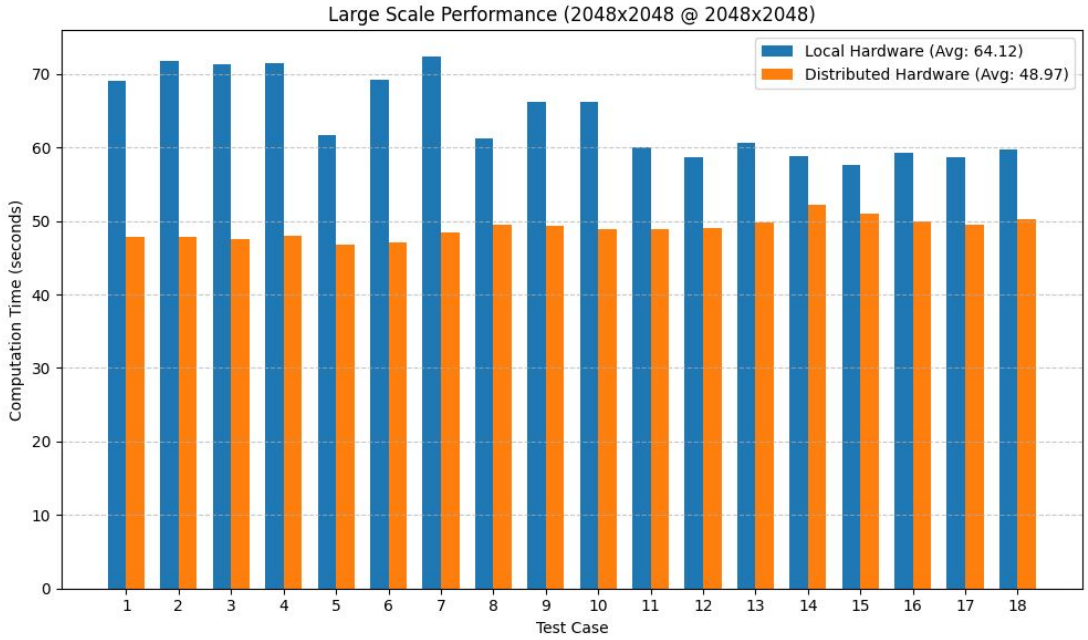




Results - Performance

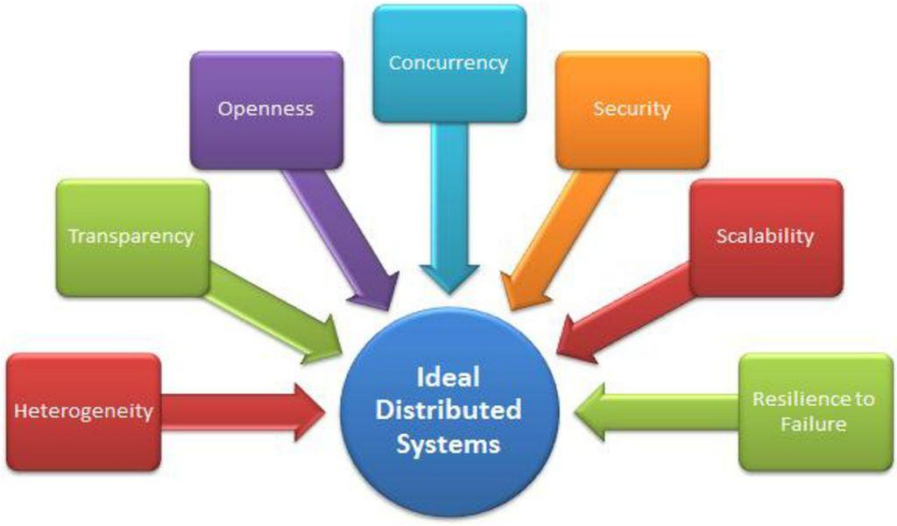






Distributed System Principles

- Scalability: Works across different worker counts
- Transparency: Hides distribution from user
- Resource Sharing: Tasks split across nodes



Lessons Learned

- Communication Overhead
- Single Point of Failure
- Effectiveness of Programming Tools
- Efficient Distributed Algorithms
- Fault Tolerance

Future Work

- Addressing the Single Point of Failure
- Improving Communication Efficiency
- Implement Worker Fault Tolerance

Conclusion

- Distributed computing for matrix multiplications
- Offers scalability, though overhead remains a challenge
- Future improvements can enhance real-world applicability

References

- API — Flask Documentation (3.0.x). 2024. url: https://flask.palletsprojects.com/en/stable/api/.
- Ankur Mallick, Malhar Chaudhari, and Gauri Joshi. “Fast and Efficient Distributed Matrix-vector Multiplication UsingRateless Fountain Codes”. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and SignalProcessing (ICASSP). 2019, pp. 8192–8196. doi: 10.1109/ICASSP.2019.8682347.
- Oded Schwartz and Noa Vaknin. “Pebbling Game and Alternative Basis for High Performance Matrix Multiplication”.In: SIAM Journal on Scientific Computing 45.6 (2023), pp. C277–C303. doi: 10.1137/22M1502719. eprint: https://doi.org/10.1137/22M1502719. url: https://doi.org/10.1137/22M1502719.
- Volker Strassen. “Gaussian elimination is not optimal”. In: Numerische Mathematik 13.4 (Aug. 1969), pp. 354–356. doi:10.1007/bf02165411. url: https://doi.org/10.1007/bf02165411.
- Serhii Zybin, Vladimir Khoroshko, Volodymyr Maksymovych, and Ivan Opirskyy. “Effective Distribution of Tasks inMultiprocessor and Multi-Computers Distributed Homogeneous Systems”. In: International Journal of Computing (June2021), pp. 211–220. doi: 10.47839/ijc.20.2.2168.