REVOLUTIONIZING HYBRID PARALLELIZATION THROUGH DATA COMMUNICATION TECHNIQUES

Asare K. Nkansah[1], Tommy Franczak[2], Thomas Marrinan[3], and Michael E. Papka[2]

[1]University of Kentucky, [2]Northern Illinois University, [3]University of St. Thomas, [123]Argonne National Laboratory

INTRODUCTION

Problem: Fluid dynamics simulation that was designed for use on single processor and not optimized for high performance computing environment.

Solution: Through the use of a step-by-step optimizations and hybrid parallelization, we can improve the performance of large-scale simulations.

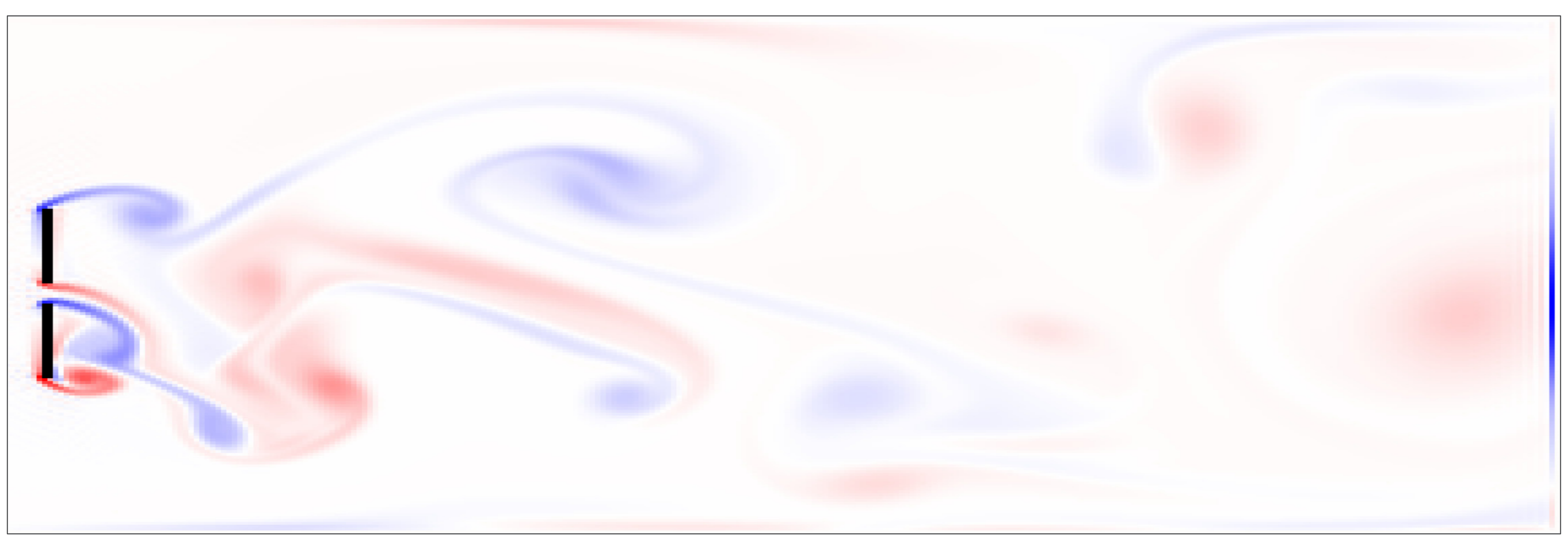

Fig. 1: Visualization of fluid vorticity produced from a single time step of the 2D Lattice-Boltzmann Method Computational Fluid Dynamic simulation (LBMCFD) (blue = spinning clockwise, red = spinning counter-clockwise). Simulation code provides developers with a compact, self-contained program with sensible visual output.

DISTRIBUTED AND SHARED MEMORY PARALLELISM

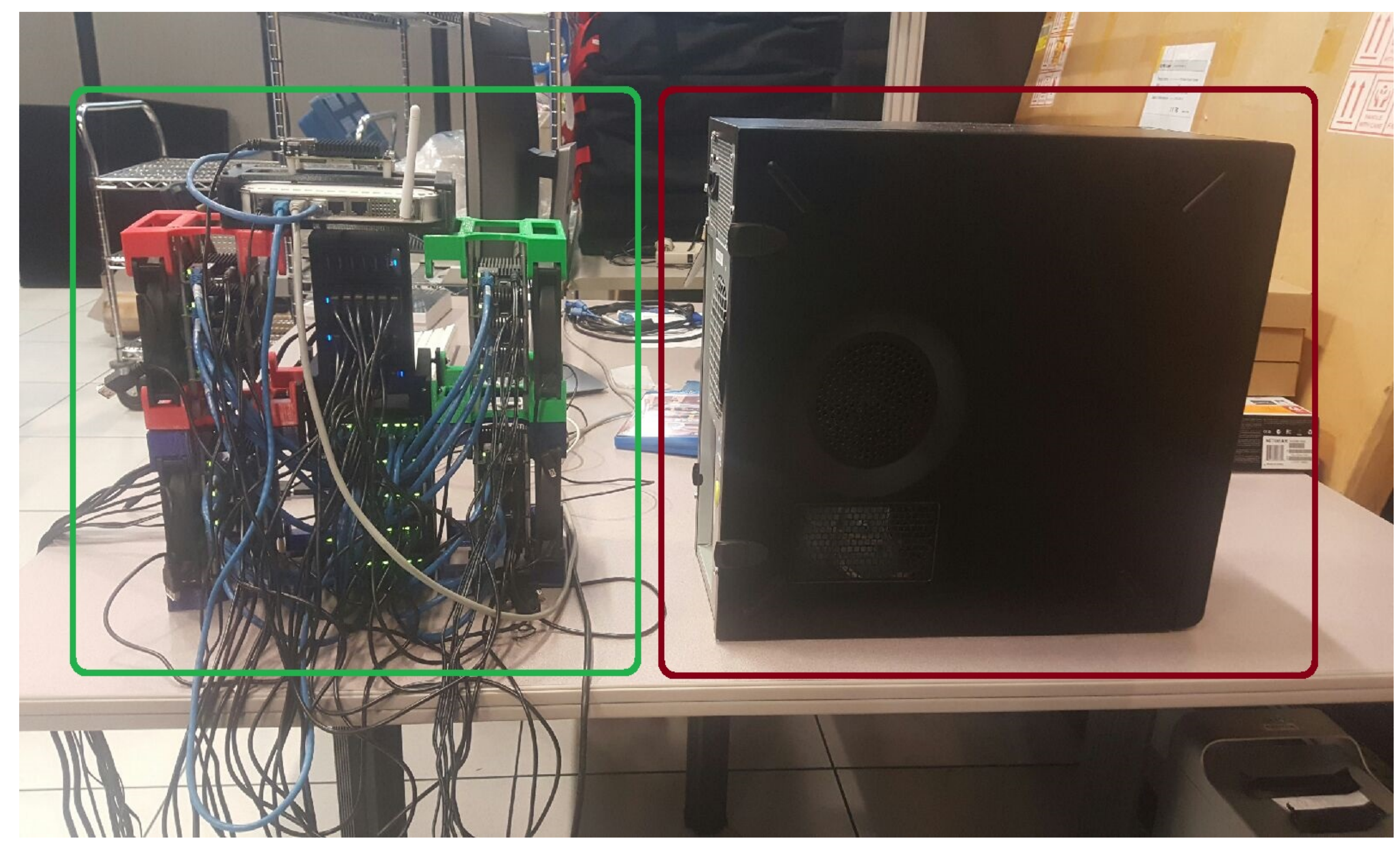

Fig. 2: Custom, mini-cluster comprised of 16 credit-card sized compute nodes, with a total of 288 cores. The mini-cluster (highlighted inside the green box) juxtaposed next to a traditional desktop computer (highlighted inside the red box). Capable of on-board parallelism by offloading programs onto the 16-cores of an Epiphany co-processor.

MAJOR ACCOMPLISHMENTS

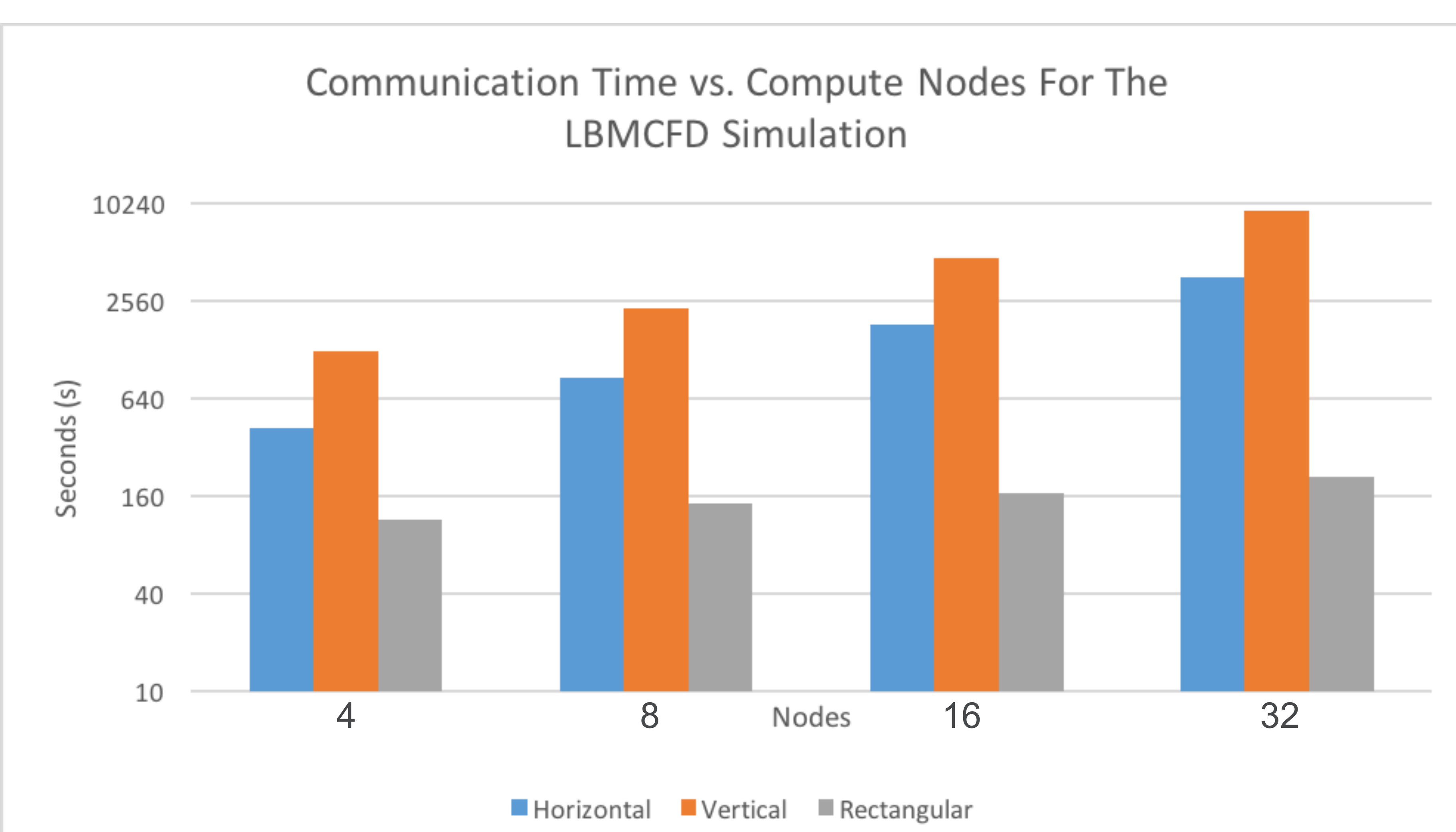

Fig. 3: Average execution time of a pure MPI 4525 x 2263 LMBCFD simulation with 60,000 time-steps based on communication pattern. The simulation used 4 MPI rank per node on a varying number of compute nodes.

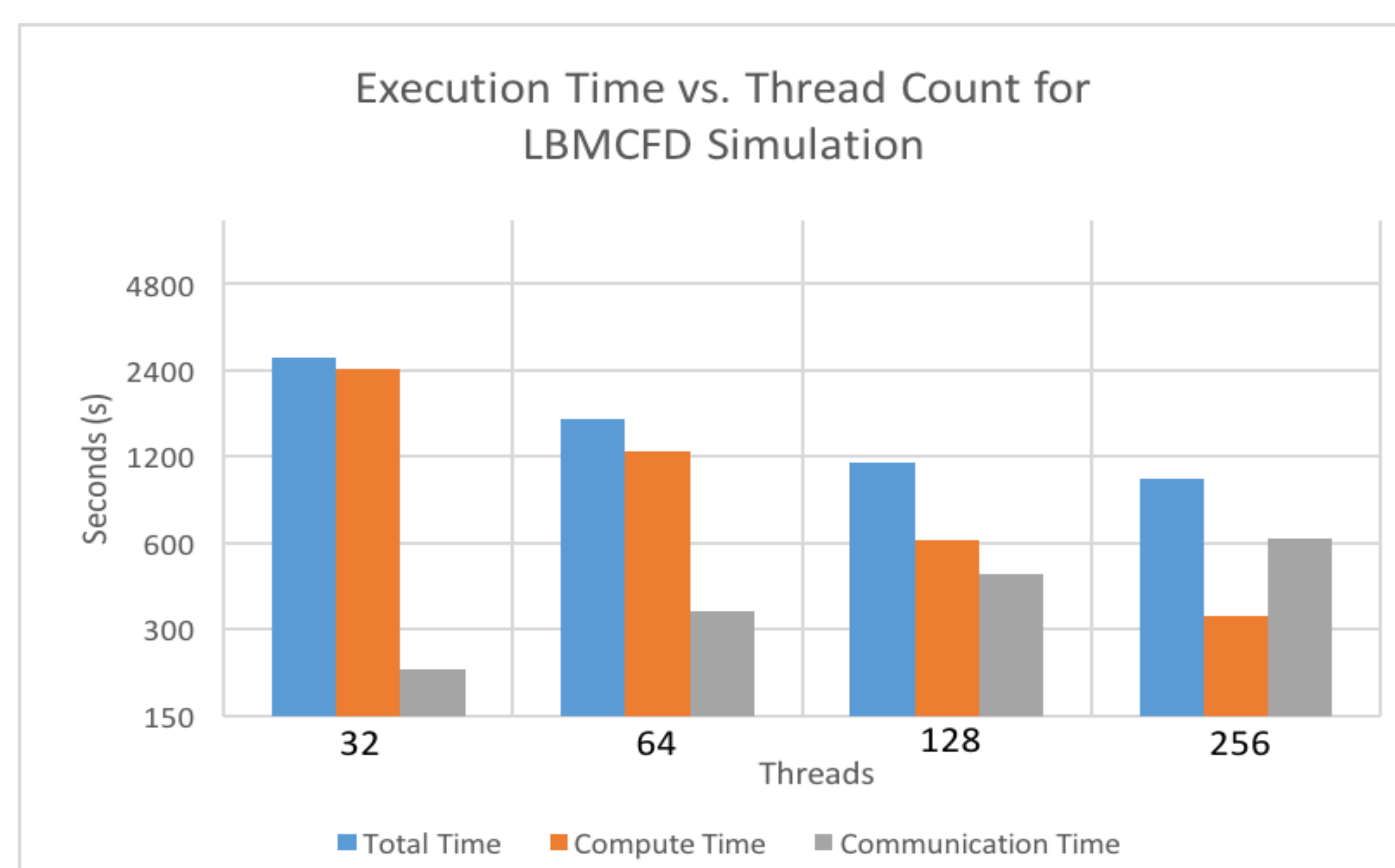

Fig. 4: Average compute vs. communication time of a hybrid 4525 x 2263 LMBCFD simulation with 60,000 time-steps using the rectangular communication pattern. The simulation used 1 MPI Rank per node and 8 threads per rank.

CONCLUSION

- For pure distributed memory parallelism, the simulation using the rectangular grid pattern proved to be the most efficient method of communication compared to the vertical and horizontal communication patterns.
- For shared memory parallelism, increasing the amount of threads used in the program had direct impact on the overall performance of the simulation by substantially decreasing the elapsed time.
- For the hybrid parallelized simulation, the results proved that combining both distributed and shared memory parallelism techniques is superior to either individually.

OUTLOOK

- The results of this research proved that the performance of hybrid parallelism is at optimum efficiency when working in congruence.
- We will further study:
    - Other possible methods of communication between ranks and threads
    - The scalability of these methods on other parallel-capable devices
    - Application to other complex computational-based programs

ACKNOWLEDGMENT: This research was supported by and used resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility operated under contract DE-AC02-06CH11357.