Exploring the Scalability of Distributed Computing Networks

PURDUE UNIVERSITY FORT WAYNE

Jacob Bushur

Advisor: Dr. Chao Chen

Department of Electrical and Computer Engineering

Introduction

Cancer research, vaccine development, and climate science are a few examples of the many fields that require large amounts of computational power. One way that researchers acquire the necessary power is through distributed computing networks. By breaking large problems into small pieces, researchers can use computers scattered across the globe to solve a single problem. ***However, what are the limits of this system architecture? What is the marginal benefit of breaking a problem into smaller pieces? Is there a point where utilizing more computers fails to improve execution time?***

To answer these questions, two distributed computing networks are created. The first network consists of 16 Raspberry Pi single-board computers, and the second network consists of the computers in the Purdue University Fort Wayne (PFW) Engineering, Technology, and Computer Science (ETCS) labs. Both networks utilize HTCondor and the Berkeley Open Infrastructure for Network Computing (BOINC). The scalability of distributed computing networks is then investigated by observing the relationship between the execution time of a matrix multiplication program and the number of cores utilized. Finally, the Purdue Fort Wayne distributed computing network is established and made available to faculty and students to aid future research endeavors. It will also be connected to the World Community Grid to support global research projects such as vaccine and cancer research using BOINC [1].

Methodology

Configuring the Raspberry Pi Network:

First, each device in the Raspberry Pi network depicted in Fig. 1 was configured with Ubuntu Linux, HTCondor, and BOINC. HTCondor was configured such that the Raspberry Pis served as execute nodes (computers that would process jobs), and an HP Pavilion laptop was configured as both the central manager (the computer responsible for managing the network) and the submit machine (the computer that submits jobs for the network to run) [2]. BOINC was configured to contribute to the World Community Grid when the Raspberry Pis were idle (when the CPU usage was below a certain threshold) [3]. However, neither HTCondor nor BOINC were permitted to run while a user was logged into and using a Raspberry Pi. Therefore, the network would always perform some work without interfering with any user activity.

Testing the Raspberry Pi Network:

After configuring the Raspberry Pis, the distributed computing network was tested with a matrix multiplication program. This program, written in C, multiplied two square matrices filled with floating-point numbers. Matrix sizes of 1024x1024, 2048x2048, and 4096x4096 were tested using 1, 2, 4, 8, 16, 32, and 64 cores. The execution time was recorded using the log files produced by HTCondor.

Configuring and Testing the PFW Network:

A second distributed computing network was then created using ETCS building lab computers such as those depicted in Fig. 2 and a server from the Purdue System Cloud. The network was configured similarly to the Raspberry Pi network and tested with the same matrix multiplication program.

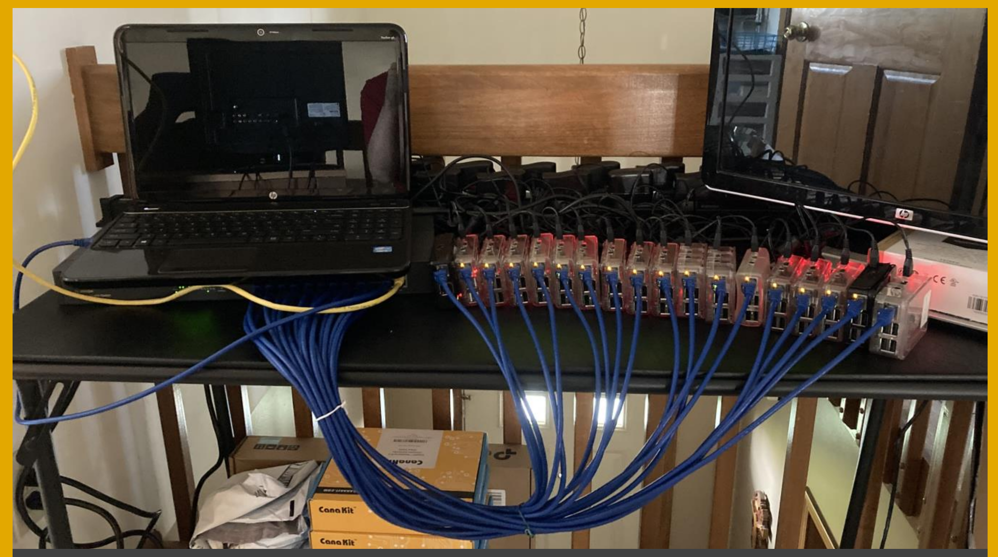

Fig. 1. The Raspberry Pi Distributed Computing Network.

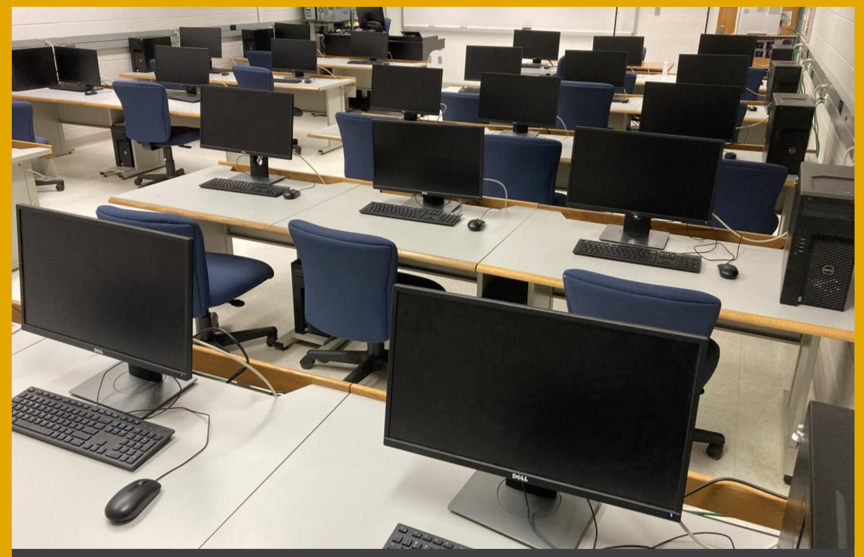

Fig. 2. A Purdue Fort Wayne ETCS Computer Lab.

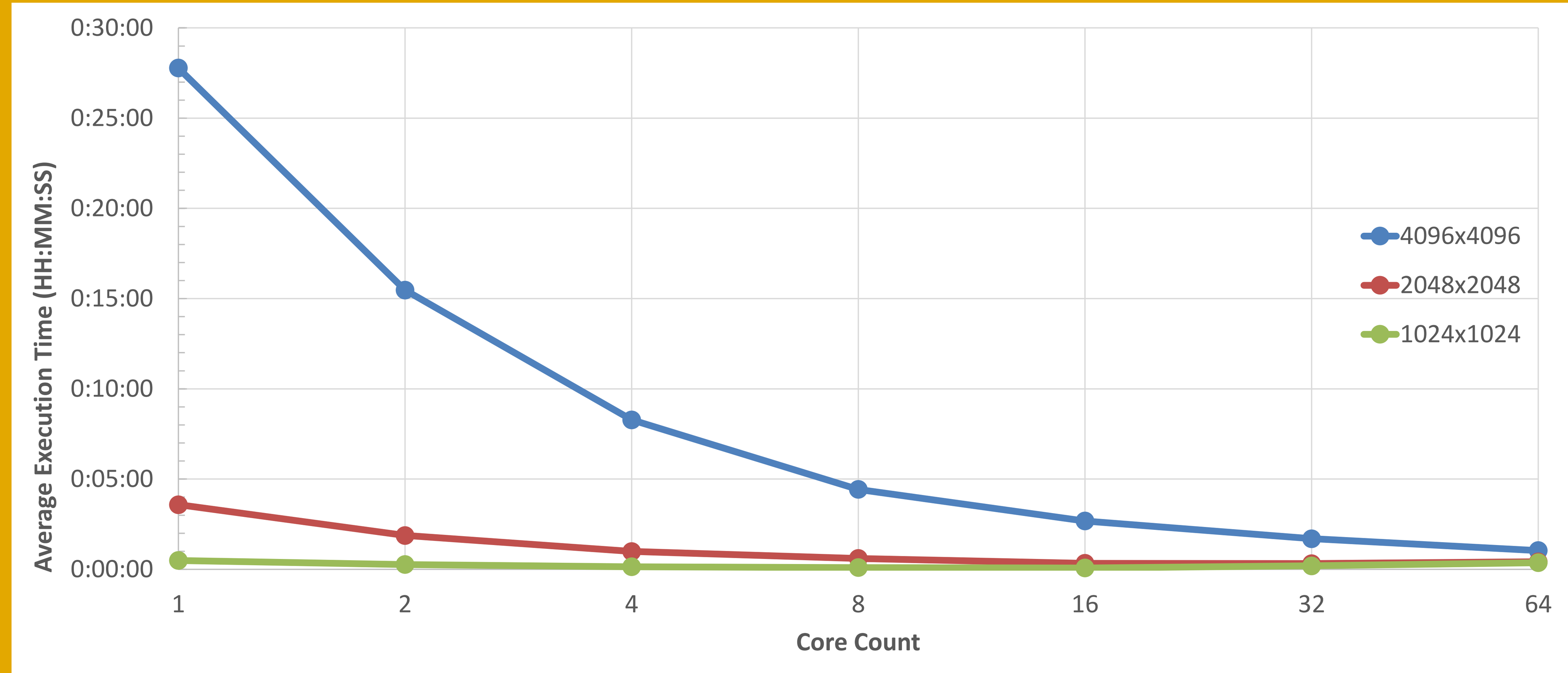

Fig. 3. Average Execution Time vs. Core Count on the Raspberry Pi Distributed Computing Network. The fastest execution times for the 1024x1024, 2048x2048, and 4096x4096 matrices utilized 16, 32, and 64 cores, respectively.

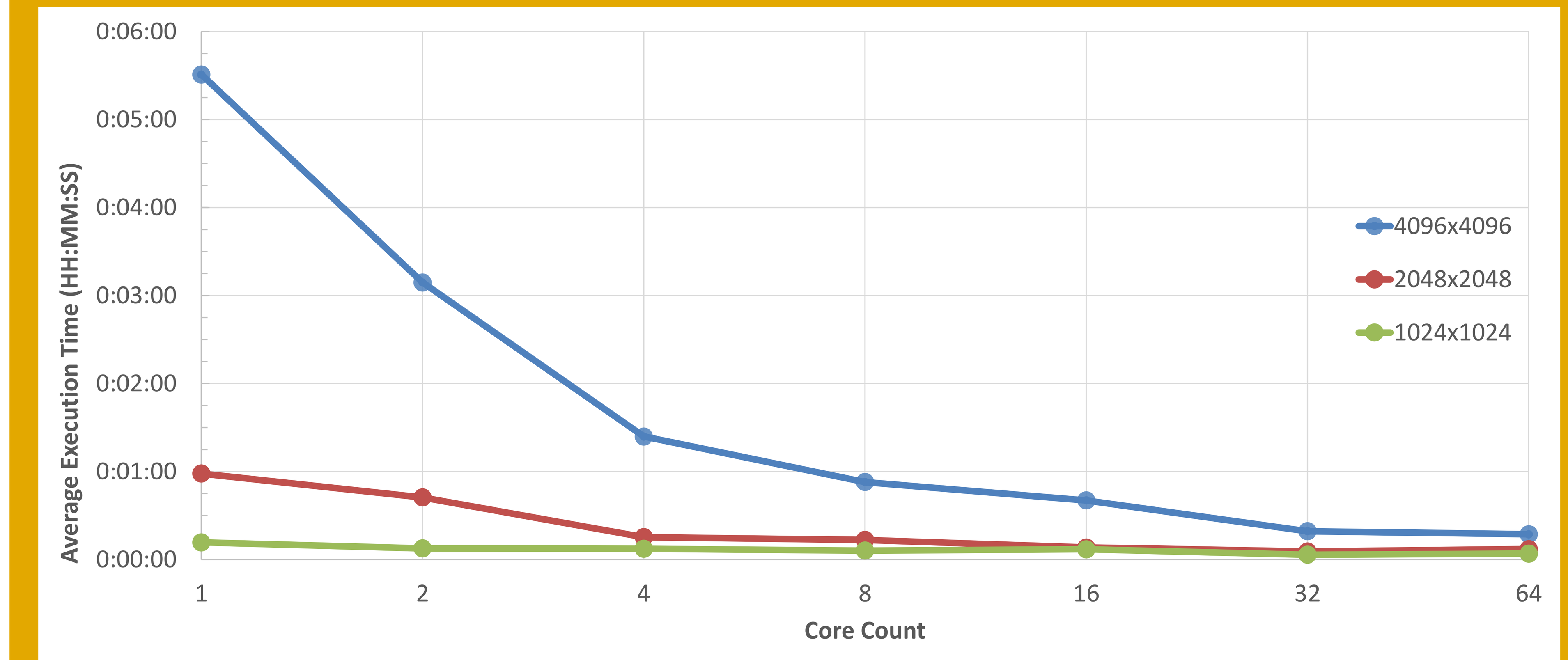

Fig. 4. Average Execution Time vs. Core Count on the Purdue Fort Wayne Distributed Computing Network. The fastest execution times for the 1024x1024, 2048x2048, and 4096x4096 matrices utilized 16, 32, and 64 cores, respectively.

Results

Observations:

Intuitively, if the number of cores is doubled, the time to compute the matrix product should be halved. Furthermore, more cores should also invariably result in a lower execution time. However, these assumptions are incorrect. As illustrated by the 4096x4096 matrices in Fig. 3 and Fig. 4, utilizing more cores resulted in a lower execution time, but doubling the number of cores did not precisely halve execution time. Additionally, the fastest trials for 2048x2048 and 1024x1024 matrices utilized 32 and 16 cores, respectively, on both networks instead of the anticipated 64 cores.

Explanations:

HTCondor's overhead most plausibly explains these observations. When a batch of jobs is submitted to the network, HTCondor must match each job with the appropriate resources, send the program and any necessary data, and receive the results. Therefore, the total time required for a job is given by (1).

$$T = M + S + E + R \quad (1)$$

Where T is the total job time, M is the time to match the job with the appropriate resources, S is the time to send any data to the execute machines, R is the time to return any results to the central manager, and E is the time to run the job.

Therefore, doubling the number of cores may halve E, S, and R. However, M is constant, resulting in an overall job time improvement of less than half. Additionally, if E is small enough, the overhead associated with using more cores and processing more jobs increases the total batch execution time.

Conclusion

In conclusion, two distributed computing networks were created. One network consisted of 16 Raspberry Pi single-board computers, and the other network consisted of the PFW ETCS lab computers. Both networks were tested using a matrix multiplication program. Defying intuition, the tests demonstrated that utilizing more cores will not always yield a faster job execution time. Nevertheless, the networks pool the power of individual computers into a significant computational resource. To access the PFW distributed computing network for future research, faculty can easily submit work by remotely logging into lpvfececon01.pfw.edu. Additionally, the network will be connected to the World Community Grid to support global projects such as vaccine and cancer research using BOINC.

References & Acknowledgements

[1] Krembil Research Institute, "World Community Grid," *www.worldcommunitygrid.org*. https://www.worldcommunitygrid.org/ (accessed Mar. 09, 2022).

[2] University of Wisconsin-Madison, "HTCondor - Home," *research.cs.wisc.edu*. https://research.cs.wisc.edu/htcondor/index.html (accessed Mar. 09, 2022).

[3] University of California, Berkeley, "BOINC," *Berkeley.edu*. https://boinc.berkeley.edu/ (accessed Mar. 09, 2022).

Special thanks to Dr. Chao Chen, Dr. Guoping Wang, the Chapman family, Michael Pressler, and Purdue Fort Wayne Information Technology Services for making this project possible.