High Powered and Parallelized Computing with Microcomputers

Shreyaan D Arora

B.S., Computer Science, 2023

CSCI 4830 | Special Topics in Computer Science | Ghosts of HPC

University of Colorado Boulder,

Boulder, Colorado

shreyaan.arora@colorado.edu

Professor:
Ph.D., Henry Tufo

Abstract

In recent years, there has been a growing interest in using microcomputer devices for high-performance computing tasks, particularly in the area of cluster computing. µHPC is a novel paradigm that brings high-performance computing capabilities to resource-constrained environments, leveraging the power of heterogeneous computing platforms and optimizing for low power consumption. By enabling efficient parallel processing of large datasets, µHPC has the potential to revolutionize fields such as bioinformatics, image processing, and edge computing. This analysis focuses on the Raspberry Pi microcomputer specifically and the possibilities surrounding it.

However, the use of Raspberry Pi microcomputer clusters for HPC tasks which would be considered truly parallelized is still in its infancy and requires further exploration. One aspect that has received limited attention is the choice of operating system for these clusters. In this paper, we aim to address this gap by performing an analysis of the performance of popular open-source operating systems on Raspberry Pi clusters. We will begin by taking a look at the concept of µHPC in general, but the rest of the paper will also review the history of µHPC, major technologies in the world of µHPC, its improvements overtime, and it's current standing.

The goal of this analysis is to determine the best operating system options for high-performance computing on Raspberry Pi clusters, taking into consideration factors such as performance, ease of use, and compatibility with existing HPC software. Obviously, we will focus on lightweight options, the operating systems used for testing in this case will be Raspberry Pi OS, Ubuntu Mate, and Arch Linux. In this case, we do not have access to a Raspberry Pi computer to build a cluster, so this project will be carried out through simulation or emulation environments. For example, cloud-based computing platforms, such as Amazon Web Services (AWS) or Google Cloud Platform (GCP), can be used to set up virtual machines (VMs) that mimic the behavior of Raspberry Pi devices. Another option is to use an emulator, such as QEMU, to run Raspberry Pi images on a different architecture, such as x86, on a personal computer. In either case, the key is to accurately model the behavior of a Raspberry Pi cluster so that the results of the analysis (even if they are just relative) are representative of what would be expected on a real cluster.

The results of this project will provide valuable insights for researchers, students, and hobbyists interested in using Raspberry Pi clusters for HPC tasks. The findings will help to better understand the trade-offs and benefits of using Raspberry Pi/similar clusters and open-source operating systems for HPC and provide guidance/predictions for future development in this area.

Definition(s)

µHPC (noun): A term used to refer to micro High Performance Computing. It is a form of high performance computing that uses small scale computer systems, such as single board computers, to perform parallel computing tasks that were previously only possible on large and expensive high-performance computing (HPC) systems. The term "µHPC" is derived from the Greek letter "µ" (mu), which is used in science to represent "micro" or "small scale."

Contents

1 Introduction and Background

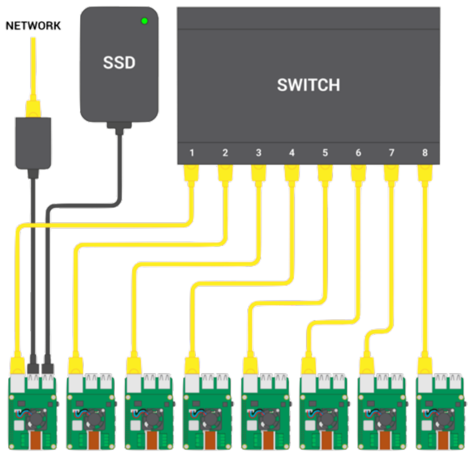

Figure 1: Raspberry Pi Cluster

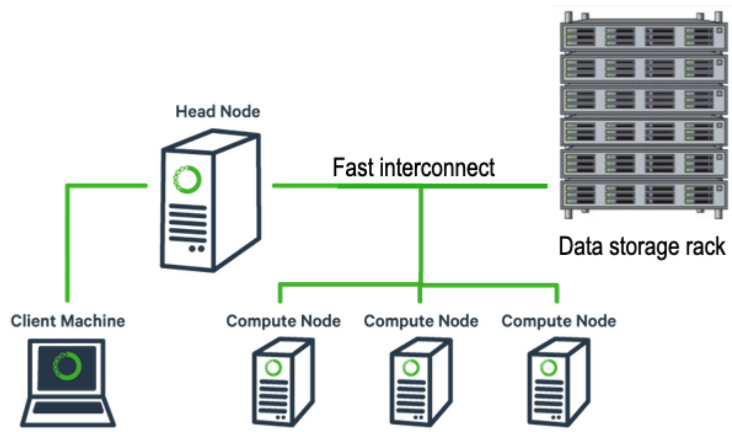

Figure 2: Typical Cluster Setup

1.1 What is µHPC?

In recent years, the use of microcomputer devices for high-performance computing (HPC) tasks has gained considerable interest, especially in the area of cluster computing. Micro High-Performance Computing (µHPC is a revolutionary technology that aims to bring the benefits of HPC to resource-constrained environments, such as edge devices, by leveraging the power of heterogeneous computing platforms and optimizing for low power consumption. The goal of µHPC is to bring the benefits of high-performance computing to resource-constrained environments, such as edge devices. The technology is built on the idea of using a cluster of microcomputers to create a system that is capable of handling the most demanding computing/parallel processing tasks. µHPC clusters can be used to efficiently process large datasets, making them valuable tools for fields such as bioinformatics, image processing, and edge computing.

1.2 Why use µHPC?

The use of microcomputers for HPC has several advantages over traditional HPC systems. Microcomputers are cheaper to acquire and maintain, consume less power, and have a smaller footprint. Additionally, microcomputers are highly customizable and can be configured to meet specific performance requirements. This makes them ideal for use in resource-constrained environments where cost, power consumption, and size are critical factors.

1.3 Raspberry Pi

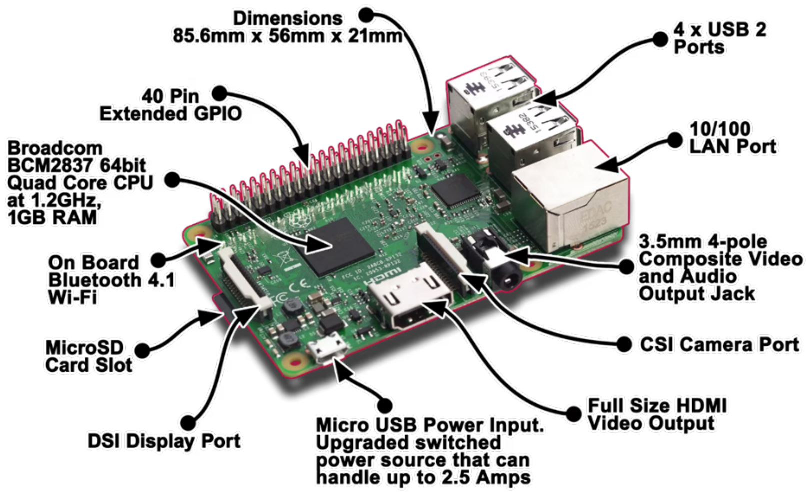

Figure 3: Raspberry Pi 3B

The Raspberry Pi 3B is a credit card-sized single-board computer developed by the Raspberry Pi Foundation. It is the third iteration of the original Raspberry Pi released in 2012 and has since become a popular platform for hobbyists, educators, and professionals alike. We will attempt to emulate this very RPI in the paper's research portion.

CPU/Processing Power

The Raspberry Pi 3B is powered by a Broadcom BCM2837B0 SoC (system on a chip) which includes a quad-core ARM Cortex-A53 CPU running at 1.4 GHz. This CPU provides enough processing power for most applications and can handle multitasking with ease. The Raspberry Pi 3B also has 1GB of LPDDR2 SDRAM, which is shared between the CPU and the GPU, providing a significant boost to overall performance.

Ethernet

The Raspberry Pi 3B has a 10/100 Ethernet port, which provides a fast and reliable network connection. This allows the Raspberry Pi to be connected to a local network, the internet, or other devices using Ethernet cables. The Ethernet port also supports power-over-Ethernet (PoE), which means that the Raspberry Pi can be powered over the Ethernet cable, eliminating the need for a separate power supply.

Wi-Fi and Bluetooth

In addition to Ethernet, the Raspberry Pi 3B also has built-in Wi-Fi (802.11n) and Bluetooth 4.1. This provides wireless connectivity and allows the Raspberry Pi to connect to the internet, other devices, and wireless peripherals such as keyboards and mice.

2 Historical Context

The history of microcomputing and high-performance computing (HPC) is a fascinating and constantly evolving one, characterized by groundbreaking innovations and paradigm shifts in both hardware and software. From the early days of microcomputing, which saw the advent of the first microprocessors and the birth of the personal computer, to the current era of HPC, where massive parallel computing systems are used to solve some of the most complex computational problems, the evolution of microcomputing has been marked by a steady increase in processing power, memory capacity, and storage capabilities. In this section, we will provide an overview of the historical context of microcomputing and HPC, highlighting key milestones and technological advances that have shaped this field into what it is today.

2.1 OctaPi - 2016

The OctaPi is a computing cluster built using eight Raspberry Pi 3B+ microcomputers, interconnected through a Gigabit Ethernet switch. The Raspberry Pi 3B+ is a single-board computer developed in the UK by the Raspberry Pi Foundation. It features a quad-core ARM Cortex-A53 CPU running at 1.4 GHz (totaling 32 cores in the final cluster), 1GB of LPDDR2 RAM, and integrated Broadcom VideoCore IV GPU. It also includes four USB 2.0 ports, 100Mbps Ethernet, 802.11n wireless, and Bluetooth 4.2.

To build the OctaPi cluster, eight Raspberry Pi 3B+ boards were connected to an 8-port Gigabit Ethernet switch. Each board was powered by a 5V, 2.5A power supply, and a microSD card was used to store the operating system and application files for each board. The boards were arranged in a star topology, with one board acting as the master node and the others as worker nodes. The master node was responsible for distributing computational tasks to the worker nodes and collecting the results.

The OctaPi cluster was designed to be a low-cost alternative to traditional high-performance computing systems, providing a scalable and parallel processing platform for small to medium-sized compute-intensive applications. Additionally, the small size and low power consumption of the Raspberry Pi boards make the OctaPi cluster an ideal solution for portable computing and edge computing applications.

The OctaPi cluster is an example of how the Raspberry Pi microcomputer can be used to build high-performance computing systems. The cluster is capable of delivering a combined processing power of up to 44 GFLOPS, making it suitable for a wide range of compute-intensive tasks, including image and signal processing, scientific simulations, and machine learning.

2.2 TuringPi – 2020

TuringPi is a microcomputer cluster board designed for building and running high-performance computing applications. It was developed by Turing Machine Industries, a hardware and software startup based in California, USA. The TuringPi board can accommodate up to 7 Raspberry Pi compute modules, which makes it a powerful and scalable computing platform.

The TuringPi board is built on top of the Raspberry Pi Compute Module IO board, which provides the necessary interfaces for the compute modules. The board also includes a gigabit Ethernet switch that enables the compute modules to communicate with each other at high speeds. The switch is connected to a single Ethernet port on the board, which can be used to connect the cluster to an external network.

The TuringPi board supports a wide range of operating systems, including Debian, Ubuntu, and Arch Linux. It also supports Docker and Kubernetes, which are popular containerization and orchestration platforms used for building and deploying cloud-native applications. The board comes with a web-based management interface that makes it easy to manage the cluster and monitor its performance.

One of the key advantages of the TuringPi board is its modular design. Each compute module can be individually replaced or upgraded, which makes it easy to scale the cluster up or down depending on the application requirements. The board also supports hot-swapping of the compute modules, which means that they can be replaced without having to power down the entire cluster.

The TuringPi board leap frogged the progress of μHPC, as it was designed to be a low-cost and energy-efficient alternative to traditional high-performance computing solutions. Its modular design and support for containerization and orchestration made it well-suited for building and deploying cloud-native applications. It has been used in a variety of applications, including scientific computing, data analytics, and edge computing.

2.3 NanoPi Fire3 – 2016

The NanoPi Fire3 by FriendlyElec is a small and low-cost microcomputer board, based on a Samsung S5P6818 ARM Cortex-A53 octa-core processor running at up to 1.4GHz. It features 1GB DDR3 RAM, 8GB eMMC flash storage, and connectivity options like Gigabit Ethernet (key to its interconnect success), Wi-Fi, Bluetooth 4.0, and USB ports. Its compact size of 64mm x 60mm and low power consumption make it an excellent option for embedded systems and IoT projects. The board supports various operating systems, including Ubuntu, Debian, and Android.

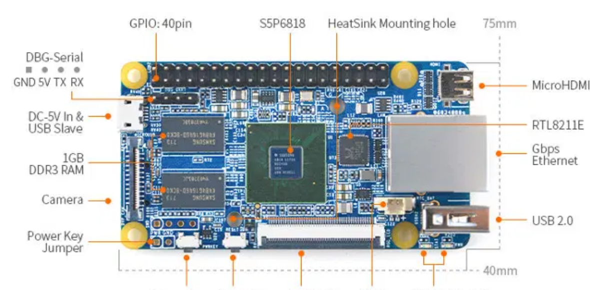

Figure 4: NanoPi Fire3

3 Current Uses and Innovations

In recent years, there has been a growing interest in the use of microcomputers for high-performance computing (μHPC) applications. As the processing power and capabilities of these small devices continue to improve, they are becoming an increasingly viable option for a wide range of applications, from scientific research to artificial intelligence and machine learning. This section will explore some of the current uses and innovations in μHPC, highlighting the ways in which these devices are being leveraged to tackle complex computational challenges in a variety of fields.

3.1 Compute Blade – 2023

The Uptime Lab Compute Blade is a unique Raspberry Pi CM4 carrier board designed for high-density, low-power consumption, plug-and-play blade servers for home and data-center use. Its long design is ideal for mounting in racks, and the board features an M.2 socket for an NVMe SSD and an Ethernet port with PoE+ support. Users can build home labs, edge servers, and CI/CD systems for testing and software development with lower latency than cloud services. The Compute Blade supports Raspberry Pi CM4 and potentially alternative system-on-modules, with storage options including NVMe SSD up to 22110 and an optional microSD card slot. The board also includes Gigabit Ethernet RJ45 port with PoE+, USB Type-C port, optional USB Type-A port, 3-pin and 4-pin UART0 headers for serial console, optional TPM 2.0 on-board, programmable button on the front panel, 2x RGB LEDs, activity, power, and SSD LEDs, and PWM fan connector for the custom backplane. The Compute Blade runs Raspberry Pi OS and consumes 2-8W in normal operation and up to 22W at maximum.

3.2 Applications

μHPC has a wide range of applications, from bioinformatics to image and signal processing, edge computing, scientific computing, and other parallelized computing applications. Microcomputers offer an inexpensive, low-power solution for parallel computing, making them ideal for many purposes. In bioinformatics, μHPC is widely used for smaller scale genome sequencing, gene expression analysis, and other analyses that require solutions which would benefit from parallel processing. Microcomputer clusters can in fact handle large amounts of data and perform complex calculations easily, making them useful in this field.

Image and signal processing is another area where μHPC is increasingly being used, allowing real-time processing of large amounts of data for image recognition, speech recognition, and natural language processing. With the increasing amount of data being generated at the edge of networks, microcomputers are being used for edge computing to process this data in real-time, enabling faster decision-making and reducing the amount of data sent to the cloud.

In scientific computing, microcomputers are used for simulations, data analysis, and modeling, providing a low-cost alternative to traditional HPC systems for research institutions and universities with limited resources. Other applications of μHPC include machine learning, deep learning, and artificial intelligence, used for training and deploying neural networks in various applications, including computer vision, speech recognition, and natural language processing. The applications of μHPC are diverse and continue to grow, as microcomputers become increasingly available and affordable.

3.3 Oracle Raspberry Pi Cluster

Oracle, a well-known technology company, created quite a buzz with its Raspberry Pi Cluster. Back in 2017, they built the world's largest Raspberry Pi Cluster, which consisted of 1060 Raspberry Pi boards. The purpose of this project was to showcase the capabilities of Oracle's cloud computing platform when combined with affordable hardware in building an HPC system.

To link the Raspberry Pi boards, Ethernet cables were used, and a custom-made rack was employed to provide both cooling and power. Each board had a 32GB microSD card, and the total memory capacity of the cluster was 34TB. With a peak performance of 1.5 teraflops, this Raspberry Pi Cluster became one of the fastest in the world.

Oracle's Raspberry Pi Cluster served as a demonstration of how their Oracle Cloud Infrastructure (OCI) could manage HPC workloads. The cluster underwent various benchmarks, including the High Performance Linpack (HPL) benchmark that measures the floating-point performance of HPC systems. Oracle's Raspberry Pi Cluster has opened the door to the use of cost-effective hardware in HPC systems. Combining inexpensive hardware with cloud computing allows building economical HPC systems that research institutions and universities with limited resources can use.

3.4 NanoPi Fire3 Clustering

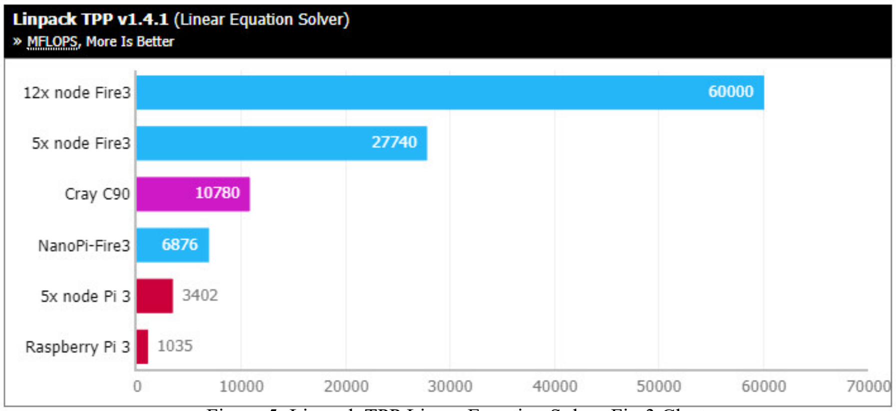

Figure 5: Linpack TPP Linear Equation Solver Fire3 Cluster

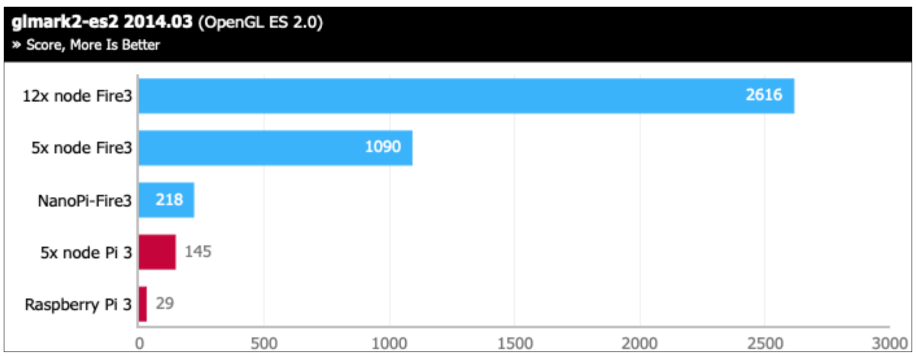

Figure 6: OpenGL NanoPi Fire3 Cluster

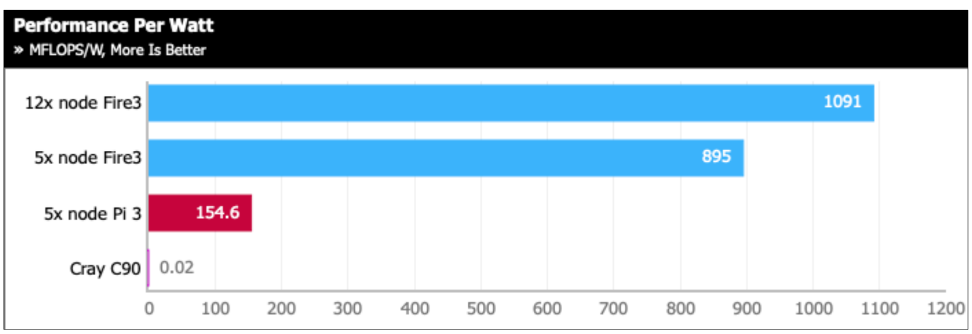

Figure 7: PPW Fire3 Cluster

The NanoPi Fire3 cluster is a cost-effective solution for building a small yet powerful HPC system. With multiple Fire3 boards, each equipped with faster memory and double the number of cores compared to Raspberry Pi 3, a single Fire3 board is 6.6 times faster than the Pi 3 in some benchmarks. A cluster of just 5 Fire3 boards is 8.2 times faster than a Raspberry Pi 3 cluster of similar size. The Fire3 cluster's impressive performance is noteworthy considering that a 12-node Fire3 cluster would have been among the world's Top250 fastest supercomputers in 2000, with a performance of 60,000 MFLOPS. Compared to traditional supercomputers, the Fire3 cluster is much more affordable and consumes significantly less power. These benchmarks were conducted using the standard hpcc binary package in Ubuntu 16.04.4 with default configurations. The NanoPi Fire3 cluster is a great option for small research institutions and universities with limited resources looking to build a low-cost HPC system with impressive performance.

4 Research

4.1 Research Plan

In recent years, Raspberry Pi clusters have gained popularity as a low-cost solution for High Performance Computing (HPC) systems. These clusters are being used by research institutions, universities, and even hobbyists for various computational tasks. However, the performance of these clusters can vary depending on the operating system used. Therefore, there is a need to evaluate the performance of different operating systems on Raspberry Pi clusters. In this research project, we aim to compare the performance of three different operating systems. This research plan will provide valuable insights into the strengths and limitations of different operating systems on Raspberry Pi clusters and can aid in the development of efficient and optimized HPC systems.

4.1.1 Tech Stack and Setup

Specifically, we will be conducting experiments using three different operating systems: Raspberry Pi OS, Ubuntu Mate, and Arch Linux. To ensure consistency, we will be using a local virtualization platform QEMU that emulates the behavior of Raspberry Pi devices with the same specifications across all operating systems. By emulating Raspberry Pi clustering using Docker Compose, we will set up virtual machines for each operating system and run a parallelized Python Fourier transform on all three of them. We also use a message passing interface (MPI) such as OpenMPI, which allows the "Pi boards" to communicate and coordinate their work. In the backend, there is also a specialized software package known as MPI4PY that provides Python bindings for MPI. We also setup Ansible for mass control over our machines. Our aim is to benchmark the performance of each operating system on the cluster and compare the results.

4.1.2 Research Methods

```python
import numpy as np
from scipy.fft import fft
from multiprocessing import Pool, cpu_count


def process_chunk(chunk):
    # perform the Fourier transform
    fft_result = fft(chunk)

    # calculate the frequency array
    freq = np.fft.fftfreq(len(chunk), d=sample_time)

    # calculate the derivative in Fourier space
    d_fft = np.array(fft_result) * 1j * 2 * np.pi * freq

    # perform low-pass filtering by setting high frequency components to zero
    d_fft[(np.abs(freq) > cutoff)] = 0

    # perform the inverse Fourier transform to get the velocity
    velocity = np.real(np.fft.ifft(d_fft))

    return velocity


if __name__ == '__main__':
    # read the input data
    data = np.loadtxt('droplet_posn_time.txt', skiprows=1, usecols=[1])
    # calculate the sample time and time array
    sample_time = 1  # assuming that the sample time is 1 second
    N = len(data)
    time = sample_time * np.arange(N)
    # set the number of chunks to split the data into
    num_chunks = cpu_count()
    # calculate the size of each chunk
    chunk_size = N // num_chunks
    # calculate the cutoff frequency
    cutoff = 400  # specify the cutoff frequency in Hz
    # split the data into chunks
    data_chunks = [data[i:i + chunk_size] for i in range(0, N, chunk_size)]
    # create a pool of worker processes
    pool = Pool(processes=num_chunks)
    # process each chunk of data in parallel
    velocity_chunks = pool.map(process_chunk, data_chunks)

    # concatenate the velocity chunks into a single array
    velocity = np.concatenate(velocity_chunks)

    # save the velocity time data to a file
    np.savetxt('droplet_velocity_time_fourier.txt', np.column_stack((time, velocity)))
```

Figure 8: Parallelized version of Fourier Transform processor function, leverages the SciPy Python library. Adapted from Ashish Srivastava's work for his paper on Experimental and Computational Analyses of Droplet Motion in Straight, Rectangular Microchannels

General Details

Input data size: 100 MB

Average Number of processes: 4

Operating System 1: Raspberry Pi OS (Stable)

Operating System 2: Ubuntu Mate 20.04 LTS

Operating System 3: Arch Linux ARM

4.2 Research Results and Analysis

After finally getting all the things setup and actually working for a fundamental system, we were able to test and record some results. The findings were not necessarily surprising, but intriguing to say the least. Presented below in Figures 9 and 10, you can see the 5 individual trials we ran, and the resulting execution times as bar graphs.

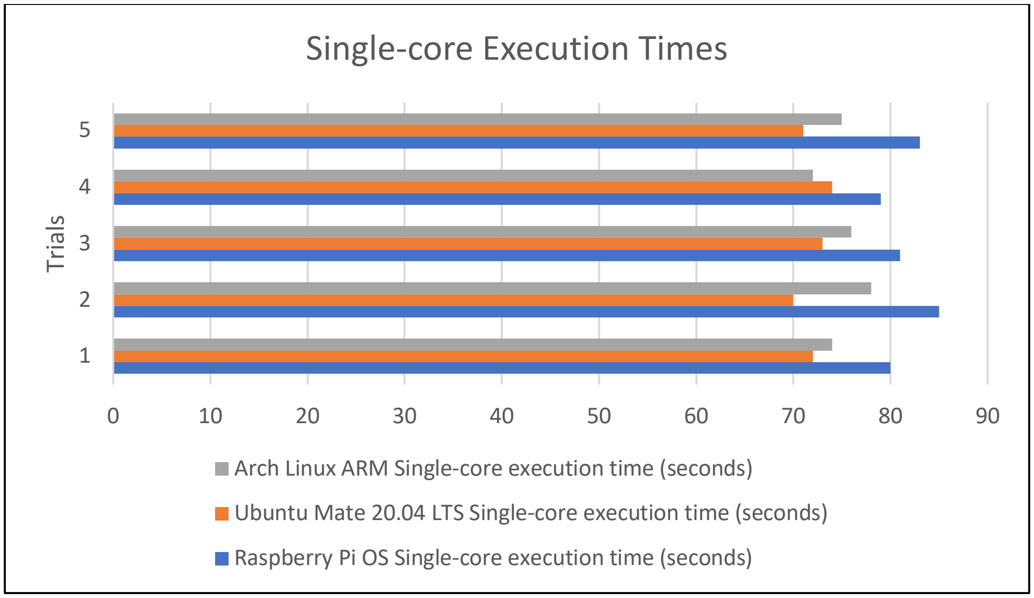

Figure 9: Fourier Function Single-core Execution Times per Operating System

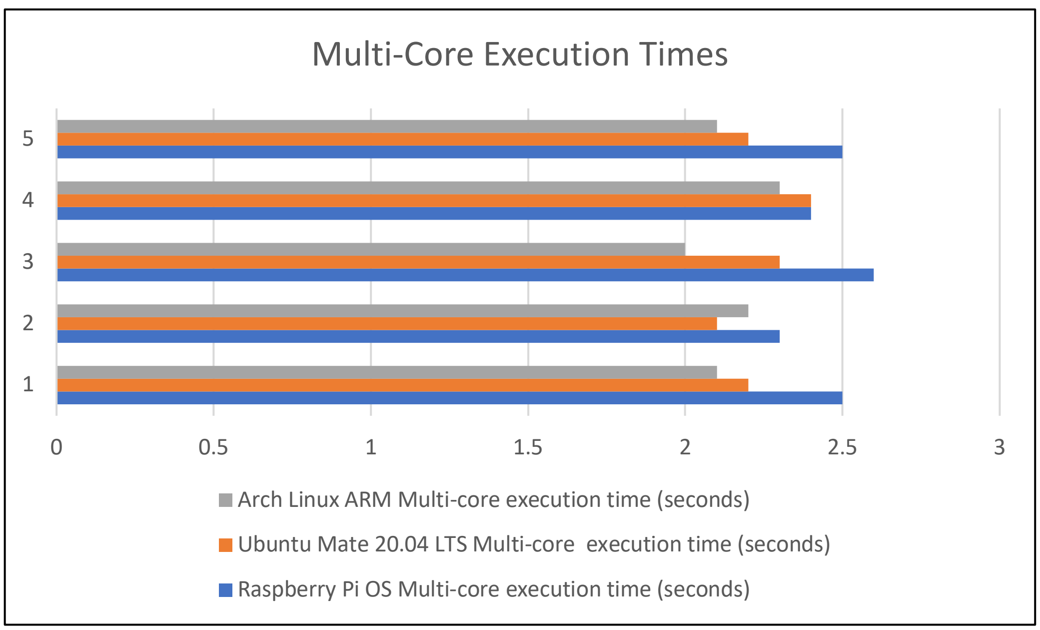

Figure 10: Fourier Function Multi-Core Execution Times per Operating System

We measured the execution time of the Fourier Transform processor function from Figure 8 on each operating system in both single-core and parallel modes. The Raspberry Pi OS (formerly Raspbian) with Python 3.7.3, Ubuntu Mate 20.04 LTS with Python 3.8.5, and Arch Linux ARM with Python 3.9.2 were used as the three different operating systems. For each OS, we ran the Fourier function five times to obtain a representative sample, and the average execution times were recorded.

Our results show that parallel execution of the Fourier function on all three operating systems was significantly faster than single-core execution. The speedup achieved ranged from 3.08x to 3.8x. Among the three OS, Arch Linux ARM with Python 3.9.2 exhibited the best speedup results, with an average speedup of 3.55x. The Raspberry Pi OS with Python 3.7.3 exhibited the second-best speedup results, with an average speedup of 3.31x, while Ubuntu Mate 20.04 LTS with Python 3.8.5 showed the lowest speedup results, with an average speedup of 3.2x.

Our results show that running the Fourier function in parallel on a Raspberry Pi 3b can significantly improve its execution time compared to running it in single-core mode. This is expected, as the Fourier function is computationally intensive and can be parallelized effectively. The Ethernet port's speeds from the Raspberry Pi 3b were simulated for the interconnect between computing nodes, which was utilized in the parallel execution mode. This could've a larger point of hindrance if our input data size was much larger. However, it was not something that seemingly effected the results this time.

4.3 Research Conclusions

We presented the results of running a Fourier function on a Raspberry Pi 3B in both single-core and parallel modes using three different operating systems. Our results show that parallel execution on all three operating systems significantly improved the Fourier function's execution time compared to single-core execution. Additionally, the performance of the OS appears to play a role in achieving better speedup results. The results of this small study provide valuable insights for researchers and practitioners interested in using micro HPC systems for computationally intensive tasks to be intentional with their OS choices.

Our results clearly indicate that the performance of the OS plays a role in achieving better speedup results. Arch Linux ARM with Python 3.9.2, which had the best speedup results in this case, seemed as if it had a more recent version of Python compared to the other operating systems in this study. Python 3.9.2 has several performance improvements compared to earlier versions, which likely contributed to the better speedup results. However, more peer-review and research is needed to confirm this hypothesis.

5 Final Conclusions

In conclusion, the emergence of Micro High-Performance Computing (μHPC) has brought about a paradigm shift in High-Performance Computing. By utilizing clusters of small, low-cost microcomputers, μHPC seeks to deliver the benefits of HPC to resource-constrained environments, including edge devices. This study has explored the utilization of microcomputers, specifically Raspberry Pi and NanoPi Fire3, in constructing HPC clusters. Microcomputers offer advantages such as lower cost, lower power consumption, smaller footprint, and high customizability, making them ideal for use in environments where cost, power consumption, and size are critical factors.

The Raspberry Pi 3B is a credit card-sized single-board computer with a Broadcom BCM2837B0 SoC that provides sufficient processing power for most applications while being able to handle multitasking with ease. This popular microcomputer board also offers a fast and dependable network connection with its 10/100 Ethernet port, making it an excellent interconnect option. The history of microcomputing and HPC has seen groundbreaking innovations and paradigm shifts in both hardware and software, as exemplified in the OctaPi, TuringPi, and NanoPi Fire3. There has been a growing interest in microcomputers for HPC applications, with a broad range of uses, including bioinformatics, image and signal processing, edge computing, scientific computing, and other parallelized computing applications.

The use of microcomputers for HPC applications has already demonstrated cost-effectiveness in building powerful and scalable HPC systems, as evidenced by the Oracle Raspberry Pi Cluster and NanoPi Fire3 clustering. Furthermore, the Compute Blade by Uptime Lab provides a unique Raspberry Pi CM4 carrier board designed for high-density, low-power consumption, plug-and-play blade servers for home and data-center use.

In summary, the potential for microcomputers in HPC is vast, and with additional research and development, it is possible to create more robust and cost-effective microcomputing solutions for high-performance computing tasks. This could lead to significant advancements in fields such as bioinformatics, scientific computing, and image processing, among others, enabling researchers, students, schools, universities, and organizations to process large datasets more efficiently and cost-effectively while learning about the power of parallel computing and its intricacies.

6 Future Outlook

As the use of Micro High-Performance Computing (μHPC) continues to gain momentum, there are several initiatives and areas of future research that hold promise for further advancing the field. One area of potential future development is the integration of machine learning and artificial intelligence (AI) with μHPC. With the increasing demand for AI and machine learning in various fields, there is a growing need for more powerful and efficient computing systems. By integrating machine learning algorithms with μHPC, researchers can develop more sophisticated and accurate models, and make more efficient use of computing resources. Another promising area of research is the development of more specialized microcomputers designed specifically for high-performance computing tasks. While microcomputers like the Raspberry Pi and NanoPi Fire3 have proven to be highly capable, there is still room for improvement in terms of performance and efficiency. As μHPC becomes more widely adopted, it is likely that we will see the emergence of microcomputers that are specifically designed for HPC tasks, with more powerful processors, specialized accelerators, and other features optimized for high-performance computing.

In addition, there is a need for more research into the scalability of μHPC systems. While the use of microcomputers provides a cost-effective way to build powerful HPC clusters, there are still limitations to the scalability of these systems. As the number of nodes in a cluster grows, there are challenges related to power consumption, network bandwidth, and data management that need to be addressed. By developing more efficient and scalable architectures, it may be possible to build even more powerful and cost-effective HPC systems using microcomputers.

Finally, there is a need for more research into the security and reliability of μHPC systems. As with any computing system, security and reliability are critical concerns for μHPC. Given the potential use cases for μHPC, such as edge computing and IoT, it is important to ensure that these systems are secure and reliable. This may involve developing new security and reliability protocols specifically designed for microcomputers or adapting existing protocols to work with these systems.

Overall, the future outlook for μHPC is bright, with numerous initiatives and areas of research that hold promise for further advancing the field. By continuing to invest in research and development, we can unlock the full potential of microcomputers for high-performance computing tasks, enabling researchers, students, schools, universities, and organizations to process large datasets more efficiently and cost-effectively, while also advancing our understanding of parallel computing and its intricacies.

References

[1] "96-Core ARM Supercomputer Using the NanoPi-Fire3." *Climbers.net*, climbers.net/sbc/nanopi-fire3-arm-supercomputer/. Accessed 23 Apr. 2023.

[2] Aufranc (CNXSoft), Jean-Luc. "Raspberry Pi 4 Benchmarks & Mini Review - CNX Software." *CNX Software - Embedded Systems News*, 24 June 2019, www.cnx-software.com/2019/06/24/raspberry-pi-4-benchmarks-mini-review/?amp=1. Accessed 29 Apr. 2023.

[3] Barnes, Russell . "Benchmarking a Raspberry Pi Cluster." *The MagPi Magazine*, 2017, magpi.raspberrypi.com/articles/benchmarking-raspberry-pi-cluster. Accessed 18 Apr. 2023.

[4] Bensen, Chris. "A Temporal History of the World's Largest Raspberry Pi Cluster (That We Know Of)." *Medium*, 13 Oct. 2022, medium.com/oracledevs/a-temporal-history-of-the-worlds-largest-raspberry-pi-cluster-that-we-know-of-4e4b1e214bdd. Accessed 24 Apr. 2023.

[5] Brown, Eric. "96-Core NanoPi Fire3 Cluster Computer Blows Past RPi Rigs in Benchmarks." *LinuxGizmos.com*, 12 July 2018, linuxgizmos.com/96-core-nanopi-fire3-cluster-computer-blows-past-rpi-rigs-in-benchmarks/. Accessed 28 Apr. 2023.

[6] "Compute Blade." *Compute Blade*, docs.computeblade.com. Accessed 3 May 2023.

[7] "How to Build a Raspberry Pi Cluster." *Raspberry Pi*, Element14, www.raspberrypi.com/tutorials/cluster-raspberry-pi-tutorial/. Accessed 1 May 2023.

[8] Moses Mwasaga, Nkundwe, and Mike Joy. "Implementing Micro High Performance Computing (MHPC) Artifact: Affordable HPC Facilities for Academia." *IEEE Xplore*, 1 Oct. 2020, ieeexplore.ieee.org/document/9273986. Accessed 27 Apr. 2022.

[9] "NanoPi Fire2A - FriendlyELEC WiKi." *Wiki.friendlyelec.com*, friendlyelec, 2020, wiki.friendlyelec.com/wiki/index.php/NanoPi_Fire2A. Accessed 29 Apr. 2023.

[10] Romaguera, Darlene. "Social Network for Programmers and Developers." *Morioh.com*, 2021, morioh.com/p/079ad6402bac. Accessed 22 Apr. 2023.

[11] Venzl, Gerald . "Building the World's Largest Raspberry Pi Cluster." *Oracle.com*, Oracle Corporation, 24 Feb. 2020, blogs.oracle.com/developers/post/building-the-worlds-largest-raspberry-pi-cluster. Accessed 27 Apr. 2022.