Raspberry Pi Cluster Performance: Scalability and Core Configuration Analysis

Vincent Cordova

Metropolitan State University of Denver

Dr. Ranjidha Rajan

29 March 2024

Abstract

The Raspberry Pi serves as a testament to the true advances in computing power and accessibility, however, its true potential lies in its ability to easily and affordably assemble a computer cluster. This ability transforms the Raspberry Pi from a single-board computer into a gateway to exploring distributed computing, parallel processing, and the scalability challenges faced by larger industry standard systems. A cluster of Raspberry Pis, also known as a "bramble", pushes the capabilities of what you can do with such small and accessible computers by utilization of multiple cores through distribution between each "node" or CPU. While such clusters offer a unique avenue for exploring the scalability of computational performance, they also aren't immune to the complexities of workload distribution–specifically, the diminishing returns introduced by the simple message passing model used for inter-node communication. This study investigates these tradeoffs, offering insights in the scalability within computer clusters, and finding the boundaries of what small, accessible computers can achieve in a distributed computing landscape. We used a Raspberry Pi cluster consisting of 2 Raspberry Pi 4Bs totaling our resources to 8 CPU cores and 8GB of RAM. We made two test programs to gather performance data from the cluster, using program completion time as the main metric recorded and analyzed. Our first program was tailored towards measuring the raw computing power of the cluster, while the second program was meant to test the cluster in a more practical light with the traveling salesman problem. Overall, we found that the cluster's increase in performance was highly dependent on the program being run, as the Message Passing Interface model has a high overhead tax in the total completion time. However, in all cases the cluster ended up greatly surpassing the performance of what the program would do on a single computer. This research goes to show that the limitations of distributed computing comes from the programming design, and the nature of message passing producing a large amount of overhead between computers. So, the question of whether or not a computer cluster would be worth using or not ends up highly dependent on what you'd be using it for. While they serve as exceptional tools for working through large amounts of calculations, adding more computers to a cluster doesn't necessarily equate to faster program completion times, and can even slow programs down.

Introduction

As software complexity continuously escalates as time goes on, the hunger for computing power grows greater and greater. Hardware technologies are always trying to catch up to the demands being asked of new software innovations. Consider OpenAI's GPT-3, with a deep learning model spanning 175 billion parameters to deliver near real-time responses. Running such models on one processor requires computing power far beyond whatever the average computer's hardware is capable of today, and requires GPUs or specialized hardware designed for machine learning tasks. Against the looming backdrop of computing power requirements, cluster computing and distributed computation serve as pivotal strategies, offering a solution by linking multiple processors to work on the same task concurrently. This research zeroes in on the utilization of Raspberry Pi clusters, a cost effective and scalable platform, to understand the efficiency, scalability, and trade-offs associated with distributed computer frameworks. By picking apart the performance dynamics of Raspberry Pi clusters, this study aims to uncover the point at which adding more nodes ceases to yield proportional benefits.

Model	CPU	RAM	USB	Network	Video Output
Raspberry Pi 4b	ARM Cortex-A72 1.5GHz	1GB, 2GB, 4GB, 8GB	3.0, 2.0	1Gb/s	Micro HDMI
Raspberry Pi 3b	ARM Cortex-A53 1.2GHz	1GB	2.0	300Mb/s	HDMI
Raspberry Pi 2b	ARM Cortex-A7 900MHz	1GB	2.0	100Mb/s	HDMI

Context and Rationale

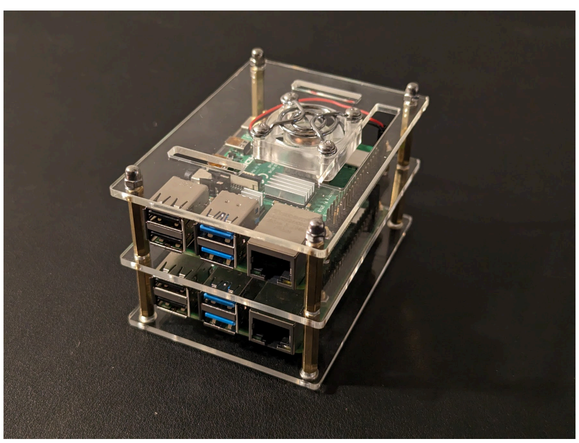
Vaguely speaking, a Raspberry Pi is a small computer that could fit in your pocket. There are many different kinds of Raspberry Pis, and they each tend to have different technical specifications based on their recency. As time goes on, technology advances, and these new technologies are applied to new Raspberry Pi models. All Raspberry Pis run on ARM architectures which are associated with RISC principles as opposed to the CISC architectures that you see in most modern day computers.We chose to do this research experiment because cluster computing is a very relevant form of boosting computing power and reducing program turnaround times. However, it is important to understand the usefulness alongside the drawbacks of cluster computing using message passing so that the practice of making multiple computers work together on the same task can be further optimized. Since computer clusters are now more accessible to make than ever, it was the perfect opportunity to analyze cluster computing, find out why it's a powerful computing solution, and to determine to what extent it's capable of completing tasks faster.

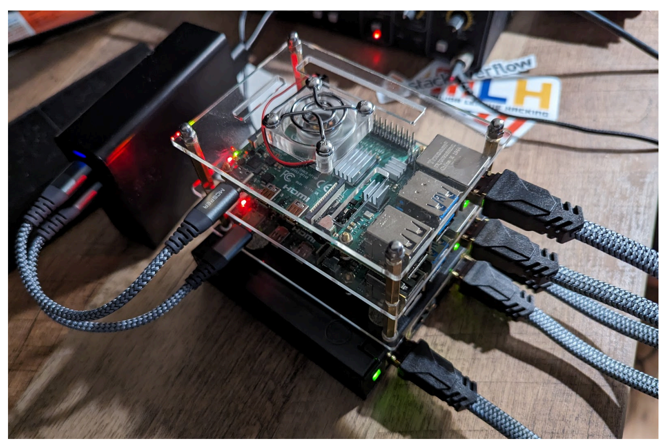
Performance and Benchmarking

The Raspberry Pi 4b's CPU is quite powerful compared to other Raspberry Pi models, and can have much more RAM than previous versions were capable of. It also uses more recent technologies than other versions, such as a USB-C port, a gigabit ethernet port, and micro-HDMI ports. This makes the Raspberry Pi 4b more capable of any task than the other models, and it serves as a perfect computer to explore cluster computing with. In general, Raspberry Pis are quite affordable computers and are very accessible to most people, which means cluster computing is more in reach than ever before due to the development of technology over time. We chose to test the cluster first with an arithmetic test program because arithmetic is the simplest and most straightforward thing that computers do. It is the best kind of test to find results about the raw computing power of the cluster with different amounts of resources. Afterwards, we chose to test the traveling salesman problem with brute force because it is a bit more methodical than the arithmetic test, and is closer to a real-world application of cluster usage. The traveling salesman problem is still widely studied today, mainly because no amount of computing power in reach is able to find solutions with too many nodes due to the time complexity of the program being (n!). This was a good test to see just how far adding computing power can get you when solving such problems.

Experimental Setup

"A cluster is a collection of parallel or distributed computers which are interconnected among themselves using high-speed networks, such as gigabit Ethernet, SCI, Myrinet and Infiniband. They work together in the execution of compute intensive and data intensive tasks that would be not feasible to execute on a single computer."(1) The Raspberry Pi cluster consists of 2 Raspberry Pi 4b's with 4GB of RAM and 4 cores in each of their CPUs. They are mounted on a cluster case, have heatsinks applied to the CPU and RAM, and each have one small fan blowing air into the heatsinks. They both get their power from a USB-C power hub which is plugged into an outlet. To communicate with each other, they are each connected to a gigabit ethernet switch with ethernet cables. Using this setup, both of the Raspberry Pis stayed cool through operation and ran into no networking or power draw issues.





Discussion

When performing tests using the Python programs on the Raspberry Pi cluster, there were multiple data points that could be recorded. However, since this research had the objective of quantifying the performance of the cluster under different core configurations, the main data point we aimed to record was the completion time of the programs. Specifically, we measured the time it takes for the program to load, run, and complete, otherwise known as the "real" time.

The reason for doing this was because the reason why people build computer clusters is for faster computation and sooner given outputs, so it would make less sense to measure only the CPU time of the cluster in case there are other factors that show themselves as drawbacks in computer clusters, such as an increase in overhead time taken.

Results

Maecenas id luctus ligula. Cras condimentum eleifend nibh sit amet iaculis. Suspendisse placerat sollicitudin mi, vel ornare augue hendrerit ac. Nulla sed suscipit sapien. Cras pellentesque orci lectus, eu consequat enim.

Arithmetic

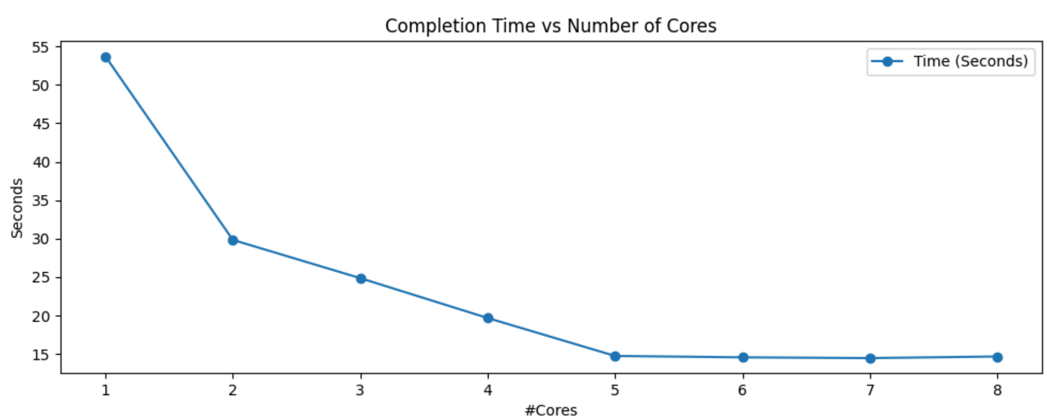
Starting with the program that performs arithmetic operations over an array, a single raspberry pi utilizing one core takes 58.94 seconds in real time to complete. When using two cores, the turnaround time has a substantial reduction and returns in 22.72 seconds. This is the largest reduction in performance exhibited when performing tests with this program, because after adding more and more cores, the reduction in turnaround time becomes less with every core added.

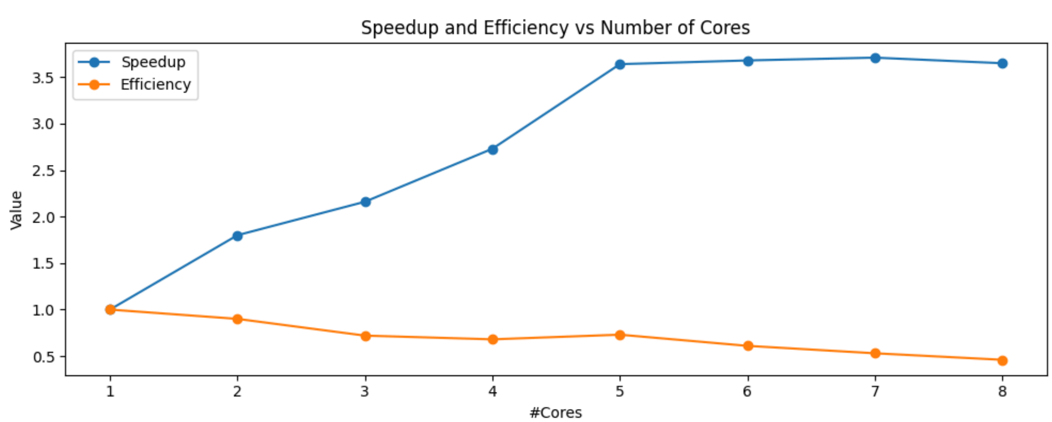
#Cores	Time (Seconds)	Speedup	Efficiency
1	58.94	1.0	1.0
2	22.72	2.59	1.29
3	12.65	4.66	1.55
4	9.44	6.24	1.56
5	7.38	7.99	1.6
6	6.29	9.37	1.56
7	5.34	11.04	1.58
8	4.71	12.51	1.56

Traveling Salesman

When testing the traveling salesman problem, the cluster did reduce the overall time taken to complete the program, but didn't show promise for efficiency per core. The greatest improvement in speedup happened using two cores, which completed at almost twice the speed of using one core. However, as more cores are added, the cluster is quick to give diminishing returns using this program starting when using five cores. When using five or more cores, the program had a negligible increase in speedup, and even showed to be slower when using eight cores than when using six or seven. The fastest completion time came from using seven cores, which hardly had a different result than when using five cores. So, although there was a large amount of speedup between using one core and using many, we saw a clear fall-off point where adding cores to the cluster would no longer be worth it.

#Cores	Time (Seconds)	Speedup	Efficiency
1	53.69	1.0	1.0
2	29.84	1.8	0.9
3	24.87	2.16	0.72
4	19.69	2.73	0.68
5	14.76	3.64	0.73
6	14.58	3.68	0.61
7	14.48	3.71	0.53
8	14.69	3.65	0.46





Discussion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan.

References

ject identifier). *Main Online Journal Name*, Vol#(Issue#), 159-192. https://doi.org/10.1000/182

Lastname, W. (2009). *Title of webpage*. Site Name. Retrieved July 3, 2019, from http://www.example.com

1.Cluster, grid and cloud computing: A detailed comparison | IEEE Conference Publication | IEEE Xplore