Designing a Many-core Embedded Server

Shalman Jesse Ojukwu- MEng Electronic and computer Engineering
Dr Rishad Shafik- School of Engineering

Newcastle University

SUMMARY & MOTIVATION

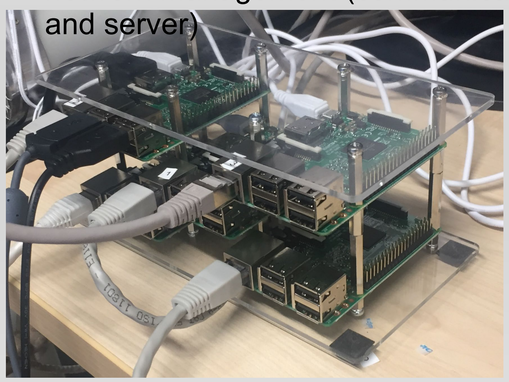
- Distributed computing is a computing concept that refers to multiple computer systems working on a single problem by dividing the problem into many parts which are solved by different computers.
- The overall goal of distributed computing is to maximize performance by connecting users and IT resources in a cost-effective, transparent and reliable manner.
- This project takes this concept and scales it down by using a low cost- many core embedded system (a small computerized devices designed for a specific function) or in our case a raspberry pi, enabling the many-core computing and inter-core

AIMS AND OBJECTIVE

- The overall aim of the project was to understand the hardware to software interfacing of an embedded system in a practical application

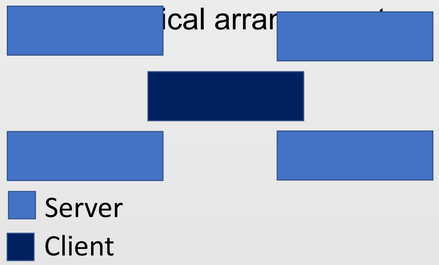
OBJECTIVE

- To design and build a many core embedded server with one client machine and four server machines as shown in the picture below
- Decide a suitable programming language for integration
- Decide and implement a suitable method for network management (communication of client and server)



ARCHITECTURAL DESIGN AND INTEGRATION

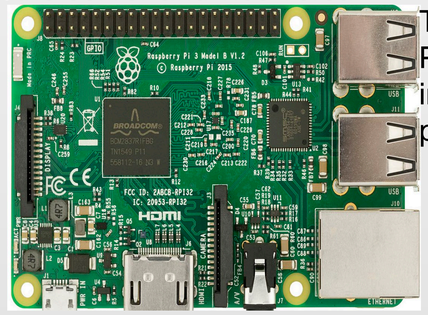
We used 5 raspberry Pis in our experiment. One acting as the **client**, in charge of receiving the problem and dividing the workload, and the other four acting as the **server**, each receiving a share of the workload, computing the workload, and sending the computed workload back to the client device. A huge problem



The diagram on the left is the chosen design as it both presentable, sturdy and easy to understand which device is the client and the server

INTEGRATION

Due to a lack of an available network, we decided to use a high-speed switch and ethernet cables as a link between the clients and the server; With the client being able to communicate with the server and vice versa



This is an image of a raspberry Pi 3 with all its various inputs including HDMI port , 4 USB port and an ethernet port

This Photo by Unknown Author is licensed under CC BY-SA

RESULTS AND DISCUSSIONS

SCREENSHOT OF TIME INFORMATION OF JOBS WITH ONE RASPBERRY PI AS THE SERVER

```
Node |  CPUs |    Jobs |  Sec/Job | Node Time Sec |    Sent |    Rcv
-------------------------------------------------------------------------

     raspberrypi |     4 |      16 |     13.2 |         211.5 |   2.

Total job time: 211.545 sec, wall time: 59.484 sec, speedup: 3.556
```

SCREENSHOT OF TIME INFORMATION OF JOBS WITH FOUR RASPBERRY PI AS THE SERVER

```
Node |  CPUs |    Jobs |    Sec/Job | Node Time Sec
-------------------------------------------------------------------------
 192.168.1.49 (raspberrypi)     |     4 |       4 |     16.040 |
 192.168.1.202 (raspberrypi)    |     4 |       2 |     12.031 |
 192.168.1.191 (raspberrypi)    |     4 |       2 |     13.029 |
 192.168.1.223 (raspberrypi)    |     4 |       0 |      0.000 |
 192.168.1.116 (raspberrypi)    |     4 |       2 |     10.025 |
 192.168.1.27 (raspberrypi)     |     4 |       2 |     15.535 |
 192.168.1.167 (raspberrypi)    |     4 |       4 |     14.537 |
 192.168.1.50 (raspberrypi)     |     4 |       0 |      0.000 |

Total job time: 223.548 sec, wall time: 20.245 sec, speedup: 11.042
```

- Both results suggest that the wall time of the same function is sped up by about a quarter when we use four raspberry pis instead pf the one as with one raspberry pi as the server (to simulate a single computer) we had a wall time of 59.484s while with four sever raspberry pi we have a wall time of 20.245s.
- Although still comparatively slower than a normal PC. One can only hypothesize that with more and more raspberry pis, it would eventually hit.
- Although a lot of tradeoffs are in check as network management becomes a problem

FURTHER WORK

For further work one could determine at what point the trade-off of scalability and cost against performance becomes impractical and weigh up if we are truly better off with raspberry pis or our basic PC especially with more complex application

ACKNOWLEDGEMENT

I would like to acknowledge my supervisor for this project, Dr Rishad Shafik, for allowing me to work with him and his colleagues in the micro systems group at Newcastle University. Also I like to give a special thanks to Paul Killan for helping to sort out any new parts needed. I would finally like to thank Newcastle University, for awarding me this scholarship to explore my interest in this research area and in postgraduate study.

References

1. Raspberry . Build an OctaPi – RaspbeeryPi. [accessed at Monday 10th June 2019]. https://projects.raspberrypi.org/en/projects/build-an-octapi