Interim Report

**A study of big data processing constraints on a low-power Hadoop cluster**

(Kaewkasi & Srisuruk, 2014) research focused on creating a low-cost Hadoop computer cluster aimed for processing Big Data at an acceptable rate and remain low-cost. The aim of their research was to see if a low-cost computer cluster can be be created for a small organization. Their research methodology achieves a **quantitative** (Creswell, 2009) result by performing several benchmarks on the created cluster, being: 1) Data processing to see the max output and throughput; 2) Power Consumption; 3) Performance and finally Memory Constraints. In their findings, they noticed that the cluster could process 34GB of data in 65 minutes with a throughput of 0.51GB/min. However, they noticed CPU limitations and that the software stack prevents the use of low-cost ARM boards for processing Big Data. On the other hand, they implied that newer versions of single-board computers may equip faster multi-core processors that could be able to handle such load. Therefore, with the presented findings they concluded that low-cost ARM boards cannot be used for processing Big Data.

**A Case for High Performance Computing with Virtual Machines**

(Huang et al., 2006) research presented a solution for High Performance Computing machines that address the performance and management overhead caused using Virtualization technologies. Their aim of this research is to create a framework that would bring the benefits of Virtual Machines to High performance computing application that causes minimal overhead in performance. The research methodology achieves a **quantitative** (Creswell, 2009) result as they have created a prototype system and performed benchmarking that tests the performance of their framework. In their findings they noticed that there a very little difference when the prototype was compared against the native environment as both achieved 4.9µs latency for 1-byte messages. Furthermore, the benchmarks demonstrate that the prototype environment performs comparably with the native environment for NAS applications. The implications noted were that other costs of virtualization such as memory consumption and image management can be reduced significantly by using optimizations. In their conclusion they stated that their approach did benefit HPC applications as their virtualized InfiniBand cluster was able to deliver almost the same performance for HPC applications running natively on the machines.

**Iridis-pi: a low-cost, compact demonstration cluster**

(Cox et al., 2013) research presented their “Iridis-Pi” cluster which consisted of 64 Raspberry Pi Model B nodes to address the current problems conventional data-centre based cluster offer. Their aim of this research is to see if they can create a low-cost, easy portable, affordable computer cluster that can operate in ambient cooling while running educational applications. Their research methodology is **quantitative** (Creswell, 2009) as they explore how a low-cost and low-power ARM processor can be able to support traditional technologies such as MPI. For their results, the cluster was benchmarked in terms of computational power and data handling by utilising the LINPACK and HPL benchmark suite on the cluster. In their conclusion, they stated that a low-cost, easy portable, affordable computer cluster that can handle a decent amount of processing power can be created for use in education applications.

**The Creation of a Low-cost Raspberry Pi Cluster for Teaching.**

(Doucet & Zhang, 2019) research presented a working Raspberry Pi cluster that utilise Open-MPI to test functionality for academic learning purposes. The aim of their research was to explore to see if a low-cost computer cluster be built that would allow students and staff to compile different programming languages. Their research methodology presented a **qualitative** (Creswell, 2009) result as several steps needed to be addressed such as: 1) Can a low cost computer cluster be implemented; 2) What message processing interface allows compiling multiple programs written in different programming languages and lastly how would students be able to learn about High performance computing. However, they did mention that they would be handing in surveys to students to discuss their learning experience, but they did not add their findings in the paper. Thus, if they did implement their findings about the survey, the paper would result would be mixed. In their conclusion, they stated that a low-cost computer cluster that can compile multiple programs written in different programming languages can be created for academic purposes.

**Malware Detection using Machine Learning and Deep Learning**

(Rathore et al., 2019) research presented the classification of malware using artificial intelligence. The aim of their research was to explore if the use of machine learning can be classified for malware analysis and detection.

Their research methodology achieves a **quantitative** (Creswell, 2009) result as various machine learning algorithms and deep learning models were created and compared by testing their accuracy, true positive rate, true negative rate and precision.

In their findings, they noticed that their dataset contained a significant difference between the amount of malware and benign executables and fixed this issue by opting for using adaptive synthetic sampling to balance out the dataset.

In their conclusion, they stated that machine learning for malware analysis and detection is possible as they have achieved 99.78% accuracy when classifying malware and benign classes.

**Orchestrating Docker Containers in the HPC Environment**

(Higgins et al., 2015) research presented a prototype that utilises docker in a high-performance computing cluster that executes an MPI job encapsulated in a container and compared its performance against virtual machines. The aim of this research to see if any performance benefits be achieved using docker containers on an HPC when executing parallel applications compared to the KVM hypervisor.

Their research methodology achieves a **quantitative** (Creswell, 2009) result as they utilise benchmarking software to see if any performance is gained when tested against the KVM and then compare both on a bare metal build (natively).

In their findings, they noted that without any optimisation for the processor architecture, the performance of virtual machines and containers are on-par. However, the overall peak performance is lower for the application they tested.

Thus, demonstrating that the docker container is a more appropriate execute method for high performance parallel applications. In their conclusion, they state that utilising docker containers demonstrates performance benefits compared to the KVM.

**A Low-Cost Computer Cluster for High-Performance Computing Education**

(Pfalzgraf & Driscoll, 2014) research presented prototype that built a High-performance computing that is small, low-cost and powerful enough that supported educational activities on HPC’s. The aim of this research is to see if a system could be designed and assembled utilising off-the-shelf components while remaining at a low-cost so that it can provide a programming environment to teach students about high-performance computing.

Their research methodology achieves a **quantitative** (Creswell, 2009) result as they benchmarked their system utilising the ‘numpy’ python library.

In their findings, they noted that serial (**code is executed on one machine**) outperformed several parallel implements. However, all parallel versions of the code they created outperformed the serial code when the problem size grew above a certain point. On the otherhand, when performing matrix vector multiplications, the serial code outperformed the parallel code.

In their research, they implicated that their system contains more nodes, costs less and is simpler to construct compared to a system called LittleFe.

In their conclusion, the tests demonstrated that the system does display the behavoir of parallel systems and can be used for educational activities.

**Paper discussions and comparisons**

**Cluster aimed for an academic setting.**

The research provided by (Cox et al., 2014; Doucet & Zhang, 2019; Pfalzgraf & Driscoll, 2014) shows that these researchers created a low-cost raspberry pi cluster that was aimed for academic research purposes. Their cluster had a similar software stack that enabled students to utilise multiple programming languages to learn about high performance computing. However, the way they approached the gather results is different from each other.

(Doucet & Zhang, 2019) research was qualitative, they demonstrated how the cluster was created but they did not present any diagrams or tables regarding benchmarking. However, in the conclusion they did state they handled surveys to students to see if their research aided in the student’s learning experience when they presented a workshop, but they did not add their findings in the paper.

(Pfalzgraf & Driscoll, 2014) results were quantitative as they created a custom python script to benchmark the cluster when testing different mathametic problems against node scaling and see which performed better.

(Cox et al., 2014) research was also quantitative however, instead of creating a custom python script like (Pfalzgraf & Driscoll, 2014). They opted to use a library created specifically to benchmark clusters and by utilising it, they were able to address more important factors about the cluster such as processing power, ram, network usage and storage.

**High Performance computing utilising isolation layers.**

(Higgins et al., 2015; Huang et al., 2006) both presented solutions to improve high performance computing that utilises any from of isolation layer to improve performance issues. (Huang et al., 2006) presented a framework that builds on top virutalisation to minimise performance overhead that is caused by using VMs. On the other hand, (Higgins et al.,2015) presented a prototype that utilised docker container instead of using VMs as docker containers sit atop the kernel as a process. Both of their methodolies achieved a quantitative result as they performed benchmarks that tested the performance of the cluster when utilising their stated improvements. Thus, their research contributed to avoid overhead issues whle utilising isolation technologies while also mentioning the limitations presented with their implimentations.

Bibliography

Cox, S. J., Cox, J. T., Boardman, R. P., Johnston, S. J., Scott, M., & Obrien, N. S. (2014). Iridis-pi: a low-cost, compact demonstration cluster. *Cluster Computing*, *17*(2), 349–358.

Creswell, J. W. (2009). *Research designs: Qualitative, quantitative, and mixed methods approaches*. Thousand Oaks, CA: Sage Publications.

Doucet, K., & Zhang, J. (2019). The Creation of a Low-cost Raspberry Pi Cluster for Teaching. *Proceedings of the Western Canadian Conference on Computing Education*, 7:1--7:5. https://doi.org/10.1145/3314994.3325088

Higgins, J., Holmes, V., & Venters, C. (2015). Orchestrating Docker Containers in the HPC Environment. In J. M. Kunkel & T. Ludwig (Eds.), *High Performance Computing* (pp. 506–513). Springer International Publishing.

Huang, W., Liu, J., Abali, B., & Panda, D. K. (2006). A Case for High Performance Computing with Virtual Machines. *Proceedings of the International Conference on Supercomputing*, 125–134. https://doi.org/10.1145/1183401.1183421

Kaewkasi, C., & Srisuruk, W. (2014). A study of big data processing constraints on a low-power Hadoop cluster. *2014 International Computer Science and Engineering Conference (ICSEC)*, 267–272.

Pfalzgraf, A. M., & Driscoll, J. A. (2014). A low-cost computer cluster for high-performance computing education. *IEEE International Conference on Electro/Information Technology*, 362–366.

Rathore, H., Agarwal, S., Sahay, S. K., & Sewak, M. (2019). Malware Detection using Machine Learning and Deep Learning. *CoRR*, *abs/1904.0*. http://arxiv.org/abs/1904.02441