Association for Information Systems
AIS Electronic Library (AISeL)

CONF-IRM 2016 Proceedings

International Conference on Information Resources Management (CONF-IRM)

2016

# Feasibility of a low-cost computing cluster in comparison to a high-performance computing cluster: A developing country perspective

Eugene Botma
*University of the Free State*, eugene.botma@gmail.com

Eduan Kotzé
*University of the Free State*, kotzeje@ufs.ac.za

# 33. Feasibility of a low-cost computing cluster in comparison to a high-performance computing cluster: A developing country perspective

Eugene Botma
University of the Free State
eugene.botma@gmail.com

Eduan Kotzé
University of the Free State
kotzeje@ufs.ac.za

## *Abstract*

In recent years, many organisations use high-performance computing clusters to, within a few days, perform complex simulations and calculations that otherwise would have taken years, even lifetimes, with a single computer. However, these high-performance computing clusters can be very expensive to purchase and maintain. For developing countries, these factors are viewed as barriers that will slow them in their quest to develop the necessary computing platforms to solve complex, real-world problems. From previous studies, it was unclear if an off-the-shelf personal computer (single computer) and low-cost computing clusters are feasible alternatives to high-performance computing clusters for smaller scientific problems. The aim of this study was to investigate this gap in literature since according to our knowledge, this kind of study has not been conducted before. The study made use of High Performance Linpack benchmark applications to collect quantitative data comparing the time-to-complete, operational costs and computational efficiency of a single computer, a low-cost computing cluster and a high-performance cluster. The benchmark used the HPL main algorithm and matrix sizes for the $n$ x $n$ dense linear system ranged from 10 000 to 60 0000. The costs of the low-cost computing cluster were kept to the minimum (USD4000.00) and the cluster was constructed using locally available computer hardware components.

In this study for the cases we studied, we found that a low-cost computing cluster was a viable alternative to a high-performance cluster if the environment requires that costs be kept to a minimum. We concluded that for smaller scientific problems, both the single computer and low-cost computing cluster was better alternatives to a high-performance cluster. However, with large scientific problems and where performance and time are of more importance than costs, a high-performance cluster is still the best solution, offering the best efficiency for both theoretical energy consumption and computation.

## *Keywords*

Low-cost computing cluster, high-performance computing cluster, scientific computation, developing country.

## 1. Introduction

High performance computing (HPC) applications are often employed as powerful research instruments in academia and laboratories, and for business analytics in industry. A typical HPC

environment consists of either a supercomputer or a computer cluster to address complex computational requirements, support applications with significant processing time requirements or process large amounts of data (Middleton, 2011). HPC is used in a large variety of fields. NASA, for example, has a large range of advanced supercomputing projects including the High-End Computing Capability Project, Heliophysics Modelling and Simulation and Engineering Risk Assessment (NASA, n.d.). Other advanced supercomputing projects include weather forecasting simulation and meteorological models (Nyberg, 2013). In Africa, the Biblioteca Alexandrina (Library of Alexandria) in Egypt, which aims to host 8 million publications, uses a supercomputer for its Internet Archive (Bibliotheca Alexandrina, 2012). Also, South Africa in recent years, deployed HPC applications to conduct research in high energy physics and bioinformatics. Some of these supercomputing applications include modelling HIV-1 evolution, simulating tools for quantum computing, and Electromagnetic Computer Simulation for the MeerKAT and SKA projects (Centre for High Performance Computing, 2015).

HPCs utilise a high-degree of internal parallelism and typically use specialised multi-processors with custom memory architectures. These architectures have been highly-optimised for numerical calculations. A computing cluster on the other hand, is a group of individual computers linked by a high-speed communication bus. Computing clusters provide higher availability and reliability and are more cost-effective than a single supercomputer system with equivalent performance (Middleton, 2011). Computing clusters have also become a very attractive option for storing terabytes, or even petabytes of data inexpensively by using Apache Hadoop™ as opposed to using an HPC (Minelli et al., 2013). While expenses are falling rapidly due to an increase in low cost yet powerful hardware, there is still a large initial investment required for a dedicated cluster (Bergman et al., 2009). This can be a significant obstacle for a developing country.

HPCs consume a large amount of electrical power, for example, the HPC at the Lawrence Loivermore National Laboratory's Sequoia system uses approximately 8MW in power when achieving 16 Petaflop/s (Bates & Patterson, 2013). Electricity rates in South-Africa have rocketed by more than 170% in the last five years (Vecchiatto, 2013) and are set to rise even further. Together with these rate increases, computer hardware is also becoming less affordable, since South Africa has to rely on international manufacturers. With the local currency (Rand) depreciating on average by 14.7% per annum against the US dollar since January 2009 (United States Embassy in South Africa, 2014), organisations are pressured to maintain or replace the hardware infrastructure for a supercomputer.

This study will investigate the feasibility of using an off-the-shelf personal computer and a low-cost computing cluster for scientific computing as a means to counter these external pressure factors. For the purpose of this study, the off-the-shelf personal computer will be referred to as a single computer. Both the single computer and the low-cost computing cluster were constructed using locally available commodity computer hardware. The single computer and the low-cost computing cluster were benchmarked against each other and also against an industrial-size supercomputer in terms of performance and theoretical energy consumption. By benchmarking the three computer configurations using the High Performance Linpack (HPL) Benchmark, an initial cost performance analysis was compiled. The time and accuracy of the results were compared to determine if there were circumstances where a single computer or a low-cost computing cluster could be used as an alternative to a high-performance computing cluster.

This paper will proceed by first providing an overview of low-cost computing clusters, and specifically how it relates to computing performance and energy consumption. This will be followed with a discussion on the methodology used for the study. Finally, the results of the benchmarks test and the conclusions drawn from the results will be presented.

## 2. Overview of low-cost computing clusters

Low-cost commodity computer clusters are built using commodity-off-the-shelf (COTS) hardware components with free, or commonly used software and are linked by a high-speed bus (Apon et al., 2001; Jeun et al., 2003). The guiding principle for a low-cost commodity cluster is to have low performance, low-cost hardware working in parallel (Dorband et al., 2003). High-performance computing clusters (HPCC) on the other hand, are configured using commercial off-the-shelf components and will have fewer high-performance, high-cost hardware components (Middleton, 2011). A computer cluster will have hardware components that include computers (i.e. nodes) and networks (Apon et al., 2001; Middleton, 2011). Cluster nodes can be computers, workstations or even symmetric multiprocessors (SMP). Networks used for interconnecting cluster nodes can be local area networks (LAN) such as Ethernet and Fast Ethernet or InfiniBand communication fabric (Middleton, 2011). A computing cluster should not be confused with a parallel computer or supercomputer. A parallel computer is a single machine using a set of multi-core and multi-processors computers that work cooperatively to solve a computational problem. This can include supercomputers that have thousands of processors, networks of workstations and embedded systems (Foster, 1995). Since the focus of the study is on feasibility of a low-cost computing cluster, an overview of related work will now be discussed.

Sterling and Becker (1995) constructed the first low-cost computing (LCC) cluster using 16 off-the-shelf 486 processors in 1994. This computer became known as the Beowulf cluster computer. The Beowulf cluster computer consisted of three dedicated components: a collection of low-cost computers (general PCs) of a single type, a high-speed network for interconnecting them, and an operating system (Linux in their case) that allows the computers to operate in parallel (Bollinger, 1999). The cluster computer was tested with a series of experiments that measured the scaling characteristics in terms of communication bandwidth, fire transfer rates and processing performance (Sterling et al., 1995). The Beowulf cluster computer opened up the world of supercomputing and made it very attractive to groups that wanted to adopt clusters. The attraction lies in the (potentially) low cost of both hardware and software, and the control that builders and users have over their system (Dongarra & Van der Steen, 2012), thus avoiding the high costs of a fully customised parallel computer. Mai & De Rose (2000) presented a low-cost computing cluster architecture configuration as an alternative to the construction of parallel and distributed machines. However, their study did not evaluate the architectures in terms of cost-effectiveness, electricity usage or time-to-complete a job, but only proposed different configurations based on cost and application areas. Jeun et al. (2003) tested a cluster-based e-mail system architecture that satisfied both the high performance requirement of scalability and reliability, and the low cost requirement at the same time. Their research, however, focused on an e-mail system and did not address the factors that this study sought to evaluate, namely theoretical energy consumption, time-to-complete and cost-effectiveness. Misbabuddin et al. (2008) constructed a personal computer (PC) based cluster to execute a program in parallel for a computationally intensive complex problem. They did not compare the cluster to another computing platform. Bergman et al. (2009) created

and evaluated a low-cost computing cluster within a virtualised laboratory environment. Benchmarks tests were performed on several different workstation configurations to determine the maximum performance of the cluster, the performance drop observed due to usage of the operating system, and the number of workstations that would benefit clustering. They did not compare the low-cost computing cluster with any other cluster environment. Lang et al. (2010) conducted an experiment that compared a traditional cluster with a low power and low-cost computing cluster. The experimental results included energy measurements as well as scale-up experiments. Several SELECT queries on relational data were used for the experiment. Naik (2012) constructed an 8 node low-cost cluster using off-the-shelf personal computers and free open source software. Benchmark tests were conducted using the Message Passing Interface (MPI) programming model. They used MPICH and a parallel file system (PVFS2) in all the experiments. However, they did not compare their cluster with other computing platforms.

# 3. Methodology

A comparative study was conducted to collect quantitative data in order to compare the time-to-complete, operational costs and computational efficiency of three different computing environments. Previous studies have used this methodology successfully when conducting benchmark tests (Bergman et al., 2009; Lang et al., 2010; Naik, 2012). The three different computing environments for this study were a single computer, a low-cost computing (LCC) cluster and a high-performance computing (HPC) cluster.

## 3.1 Benchmark Applications

High Performance Linpack (HPL) benchmark applications were used to test the performance of the three platforms. HPL is software that solves a dense *n* by *n* linear system in double precision (64 bits) arithmetic on distributed-memory computers (Petitet et al., 2014). The HPL package also provides a testing and timing program to quantify the accuracy of, as well as the time it took to compute the obtained solution (Petitet et al., 2014). HPL is often used to benchmark scientific clusters in a production environment, is dependent on Random Access Memory (RAM) size and network speed, and is very CPU intensive (Davies et al., 2011). It was envisaged that the HPL benchmark will provide good insight into the efficiency, performance and effectiveness of a platform, from a theoretical point of view. The main HPL algorithm was used in this experiment (Netlib.org, 2015a).

Other software used in the experiment includes the high performance implementation of the message passing interface (MPI), MPICH. MPICH is used to run the benchmarks over multiple computers as a single program. MPICH includes a C compiler called MPICC that is needed to compile programs that implement MPI or a derivative thereof, as well as a linker called MPIF77. Automatically Tuned Linear Algebra Software (ATLAS), which is needed by HPL, was also used. The GNU Code Compiler (GCC) was used to compile MPICH and ATLAS. MPICC and MPIF77 from the compiled MPICH were used to compile HPL. Other compilers, such as the Intel Compilers that form part of the Intel Parallel Studio (which is optimised for Intel architectures), can significantly increase performance, but the increase in performance is not equal on Intel and AMD architectures. The Intel Parallel Studio is an additional software resource that includes programs often used on clusters, such as an optimised MPI implementation, linear algebra software, performance profilers and more. The Intel Parallel Studio is considered an expensive piece of software with costs that vary from USD1 199 up to USD29 499 (Intel, 2015). Since the

focus of this study was to evaluate a single computer and a low-cost computing cluster in terms of cost effectiveness, the Intel Parallel Studio was excluded from the study.

## 3.2 Computing Environment

As mentioned previously, the three computing environments consisted of a single computer, a low-cost computing (LCC) cluster and a high-performance computing (HPC) cluster. The prices of components for the single computer and LCC were obtained from Amazon.com (2014). The prices of components for the HPC were obtained from the University of the Free State's High Performance Computing Cluster division.

| | | |
|---|---|---|
| Motherboard | MSI Z77 GD65 | USD220.00 |
| CPU | Intel i7 3770k @ 4.2Ghz | USD330.00 |
| Memory | 32GB DDR3 1600 MHz (4x corsair vengeance 8Gb) | USD315.00 |
| Storage | 3 x 1TB Seagate barracuda 7200rpm – Raid 0 | USD162.00 |
| Network | 1GB On-board Ethernet | USD0.00 |
| | Total (US Dollar) | USD1 027.00 |

**Table 1**: Configuration and Investment Cost of the Single Computer

| | | |
|---|---|---|
| Motherboard | Gigabyte G41MT-S2PT | USD74.00 |
| CPU | Intel Core2Duo E7400 @2.8GHz | USD70.00 |
| Memory | 4GB DDR2 800 MHz (2x 2Gb) | USD82.00 |
| Storage | 250GB Seagate barracuda 7200rpm | USD55.00 |
| Network | 1GB On-board Ethernet | USD0.00 |
| | Total per node | USD281.00 |
| Nodes | 11+Head (12 in total) | USD3 372.00 |
| Switch | | USD600.00 |
| | Total (US Dollar) | USD3 972.00 |

**Table 2**: Configuration and Investment Cost of the LCC Cluster

| | | |
|---|---|---|
| Dell Model | Box configuration | Only an aggregate amount available per node<br><br>USD10 504.00 |
| CPU | 4x AMD Opteron 6172 @ 2.1GHz | |
| Memory | 64 GB DDR3 1333MHz Registered Ram Modules | |
| Storage | Intel x25-E SATA SSD | |
| Network | Infiniband QDR Network Adaptor.<br>Dual-Port Gigabit Ethernet controller | |
| Nodes | 4 | USD42 016.00 |
| Switch | | USD3 153.00 |
| | Total (US Dollar) | USD45 169.00 |

**Table 3**: Configuration and Investment Cost of the HPC Cluster

## 3.3 Square Two-Dimensional Grid

To maximise HPL performance for this study, a square two-dimensional grid of processors for HPL should be constructed (Netlib.org, 2015a). A two-dimensional grid indicates that certain cores will be unused in order to maximize HPL performance (Netlib.org, 2015b). For the single

computer the configuration already was a two-dimensional grid of 2x2 cores. For the LCC cluster, only 20 of the available 22 cores were used, resulting in a theoretical maximum of 216 Giga Floating-Point Operations per Second (GFLOPS) and a grid of 4x5 cores. Finally, for the HPC cluster, restrictions imposed by the system administrator of the high-performance cluster meant that only 46 of the 48 cores were usable per node. Thus, 184 cores were available but a two-dimensional grid cannot be constructed with 184 cores. To construct a 13x14 two-dimensional grid, only 182 cores were used resulting in a theoretical maximum of 1528.8 GFLOPS. These values were used in calculating the computing efficiency of each of the configurations. In order to calculate and compare computing efficiency, the theoretical peak performance (TPP), as well as the power consumption per hour (PCPH), is of particular interest to this study. Both these values will be discussed and calculated for all three computing environments.

## 3.4 Theoretical Peak Performance (TPP)

Theoretical Peak Performance (TPP) is the maximum amount of Floating-Point Operations per Second (FLOPS) that a computer or cluster can compute without any bottlenecks or overheads. It is used to determine the efficiency of a cluster and can help in discovering bottlenecks. For both the single computer and the HPC cluster, the processors execute 8 CPU instructions per cycle (FLOPS). The processors for the LCC cluster execute only 4 CPU instructions per cycle (FLOPS). See Table 4 for the TPP of each computing environment. The following formula was used to calculate TPP (Microsoft, 2013; Novatte, 2014):

$$TPP_{(GFLOPS)} = \text{node} * ( \text{sockets} / \text{node} ) * ( \text{cores} / \text{socket} ) * \text{GHz} * \text{FLOPS}$$

| | Node Performance (GFLOPS) | Computing Environment Performance (GFLOPS) | Theoretical Peak Performance (TPP) (GFLOPS) |
|---|---|---|---|
| Single Computer | 1 x 4 x 4.2Ghz x 8 | 1 x 134.4 | 134.4 |
| LCC Cluster | 1 x 2 x 2.8Ghz x 4 | 11 x 22.4 | 246.4 |
| HPC Cluster | 4 x 12 x 2.1Ghz x 4 | 4 x 403.2 | 1612.8 |

**Table 4:** Theoretical Peak Performance (TPP)

## 3.5 Power Consumption Per Hour (PCPH)

Power Consumption Per Hour (PCPH) is calculated by adding the thermal design power (TDP), 30W overhead for the motherboard and hard drive and an additional 3W for every memory module. See Table 5 for the PCPH of each computing environment.

| | Units | Calculation | Power Consumption per Hour (W) |
|---|---|---|---|
| Single Computer | Intel i7 CPU + 4 x 8GB Memory Modules | 77+30+(4 x 3) | 119 |
| LCC Cluster | Intel Core2duo CPU + 2 x 2GB Memory Modules per machine | (65+30+(2 x 3)) x 12 | 1212 |
| HPC Cluster | 4x AMD Opteron CPU's + 16 x 4GB Memory Modules | ((115 x 4)+30+(16 x 3)) x 4 | 2152 |

**Table 5:** Power Consumption Per Hour (PCPH)

## 3.6 Benchmark Test and Problem Size

The aim of the study was to investigate the feasibility of using a single computer and an LCC cluster for a scientific experiment. In order to accomplish this, the same benchmark test was executed with respect to all three computing environments and the results compared. The

benchmark test used the HPL main algorithm with problem sizes ranging from 10000 to 60000. The problem size is the size of the matrix that is used as the *n* by *n* dense linear system to be solved by the HPL algorithms. Problem sizes lower than 10000 usually do not reflect realistic results as a larger portion of the time is spent on overheads, such as data transfer, than actual computation. Although both the LCC cluster and HPC cluster have enough memory to allow larger problem sizes, the single computer cannot support problem sizes of significant increase. Thus, 60000 was chosen as the maximum problem size for the experiment.

### 3.7 Metrics

It was previously established that the increase in problem size increases the time-to-complete exponentially, but the application of additional cores to the problem reduces the time-to-complete (Sun & Chen, 2010). Furthermore, energy consumption of a computer is directly related to the time spent on the problem (Halliday et al., 2007). Therefore, this study used problem size, time-to-complete, theoretical energy consumption and cost effectiveness to evaluate the efficiency of the computing platforms.

## 4. Results

For the purpose of this study, data on the measurements for each of the computing platforms were collected over a period of two weeks. Selected results will now be presented and discussed.

### 4.1 Theoretical Energy Consumption

As previously mentioned, the energy consumption is directly related to the time spent on the problem. The theoretical energy consumption (TEC) was calculated using the power consumption per hour (W) and the time-to-complete of each experiment (Hitachi, 2014). The results of the experiment for each computing platform and problem size are shown in Table 6.

$$E_{(\mathrm{Wh})} = P_{(\mathrm{W})} \times t_{(\mathrm{h})}$$

| | Single Computer | | LCC Cluster | | HPC Cluster | |
|---|---|---|---|---|---|---|
| Problem Size | Time (s) | Theoretical Energy Consumption (Wh) | Time (s) | Theoretical Energy Consumption (Wh) | Time (s) | Theoretical Energy Consumption (Wh) |
| 10,000 | 24.07 | 0.796 | 108.25 | 36.444 | 3.84 | 2.295 |
| 20,000 | 188.40 | 6.227 | 235.99 | 79.450 | 13.00 | 7.771 |
| 30,000 | 635.24 | 20.996 | 377.63 | 127.135 | 30.76 | 18.388 |
| 40,000 | 1438.05 | 47.535 | 552.80 | 186.109 | 62.22 | 37.194 |
| 50,000 | 2901.62 | 95.915 | 1487.70 | 500.859 | 111.82 | 66.844 |
| 60,000 | 5012.21 | 165.681 | 2513.44 | 846.191 | 181.26 | 108.353 |
| **Average** | **1699.93** | **56.192** | **879.30** | **296.031** | **67.15** | **40.141** |

**Table 6:** Theoretical Energy Consumption (TEC)

To determine the Theoretical Energy Consumption Cost (TECC), the study analysed the costs associated with the Power Consumption Per Hour of each of the three computing environments. The electricity cost per hour in dollars were calculated using the following formula:

$$\text{Cost}_{(\$)} = E_{(\text{kWh})} \times \text{Cost}_{(\text{cent/kWh})} / 100_{(\text{cent/\$})}$$

The peak cost of one kilowatt hour (1kWh) was ZAR0.80 in South Africa at the time the study was conducted (NERSA, 2014). Using a South African Rand-to-US Dollar exchange rate of ZAR11.5379 per US Dollar, as on 10 December 2014 (Exchange-Rates.org, 2015), this means the peak cost of one kWh is approximately 6.93 US cents at the time this study was conducted. The results of the analyses for each computing platform and problem size are shown in Table 7.

| | Single Computer | | LCC Cluster | | HPC Cluster | |
|---|---|---|---|---|---|---|
| Problem Size | Time (s) | Cost (USD) | Time (s) | Cost (USD) | Time (s) | Cost (USD) |
| 10,000 | 24.07 | 0.00006 | 108.25 | 0.00253 | 3.84 | 0.00016 |
| 20,000 | 188.4 | 0.00043 | 235.99 | 0.00551 | 13.00 | 0.00054 |
| 30,000 | 635.24 | 0.00146 | 377.63 | 0.00881 | 30.76 | 0.00127 |
| 40,000 | 1438.05 | 0.00329 | 552.8 | 0.01290 | 62.22 | 0.00258 |
| 50,000 | 2901.62 | 0.00665 | 1487.7 | 0.03471 | 111.82 | 0.00463 |
| 60,000 | 5012.21 | 0.01148 | 2513.44 | 0.05864 | 181.26 | 0.00751 |
| **Average** | **1699.93** | **0.003895** | **879.30** | **0.02051** | **67.15** | **0.00278** |

**Table 7:** Theoretical Energy Consumption Cost (TECC)

From Tables 6 and 7 it appears that the single computer had the lowest theoretical energy consumption (6.227Wh) and cost (0.00043 USD) with regard to the small problem sizes (n<=20000). On the other hand, the HPC cluster showed the lowest theoretical energy consumption (37.19Wh) at larger problem sizes (n>=40000). This is because the HPC cluster used a fraction of the time-to-complete the same problem as the single computer and LCC cluster. The LCC cluster had the highest theoretical energy consumption (846.191Wh) and cost (0.05864 USD), even though the time-to-complete the problem (2513.44s) was half of the single computer time-to-complete. The LCC cluster had a much higher energy rating than the single computer, but the difference in time-to-complete was not significant enough to counter act the energy consumption, unlike in the case of the HPC cluster.

## 4.2 Cost Effectiveness

Using the results of the previous findings (see section 4.1), a theoretical timeframe was calculated when the operational cost and initial investment of a LCC cluster (USD 3972) equals the initial investment of a HPC cluster (USD 45 169). The Power Conception Per Hour of 1.212kW of the LCC and the peak cost for 1kWh of 0.00693 USD were used. The theoretical timeframe was calculated as follows:

$$Time = \frac{HPC\ Initial\ Investment - LCC\ Initial\ Investment}{24hours \times Power\ Consumption\ Per\ Hour\ (kW) \times Peak\ Cost\ of\ 1kWh}$$

$$Time = \frac{45169 - 3972}{24 \times 1.212 \times 0.0693}$$

$$Time = \frac{41197}{2.0157984}$$

$$Time = 20437\ Days = 55.95\ Years$$

The calculation assumed that the LCC cluster will operate on a 24x7 basis without downtime and at maximum capacity for the duration of the 56 years. The operational costs of the LCC cluster also did not take into account costs associated with cooling, part replacement and maintenance costs but solely focused on energy consumption costs. This result showed that there is a practical application for using a LCC cluster in an environment where the cost factor is more important than the time-to-complete factor, but are subject to a number of limitations and factors, including the available time as well as the size of the problems to be completed. In other words, the LCC cluster could hypothetically provide an organisation with low-cost cluster computing capabilities for 56 years before it exceeds the investment capital investment of USD45 169 for a HPC cluster.

## 4.3 Computational Efficiency

Computational efficiency is used to determine how efficiently the processors of a supercomputer can be utilised. In other words, to determine if a component other than the processor is causing a bottleneck or lower performance of the supercomputer as a whole. Computational efficiency is calculated by using the best performance and theoretical peak performance. The following formula was used (Microsoft, 2013):

Efficiency = Best Performance GFLOPS / Theoretical Peak Performance GFLOPS

The results for the best performance of each computing platform and problem size are shown in Table 8 and Figure 1.

| | Single Computer | | LCC Cluster | | HPC Cluster | |
|---|---|---|---|---|---|---|
| TPP (GLOPS) | 119 | | 1212 | | 2152 | |
| Problem Size | Actual (GFLOPS) | Efficiency (%) | Actual (GFLOPS) | Efficiency (%) | Actual (GFLOPS) | Efficiency (%) |
| 10,000 | 27.700 | 20.61% | 6.160 | 2.85% | 173.600 | 11.36% |
| 20,000 | 28.310 | 21.06% | 22.600 | 10.46% | 410.200 | 26.83% |
| 30,000 | 28.340 | 21.09% | 47.670 | 22.07% | 585.300 | 38.28% |
| 40,000 | 29.670 | 22.08% | 77.190 | 35.74% | 685.800 | 44.86% |
| 50,000 | 28.720 | 21.37% | 56.020 | 25.94% | 745.300 | 48.75% |
| 60,000 | 28.730 | 21.38% | 57.290 | 26.52% | 794.500 | 51.97% |

**Table 8:** Computational Efficiency

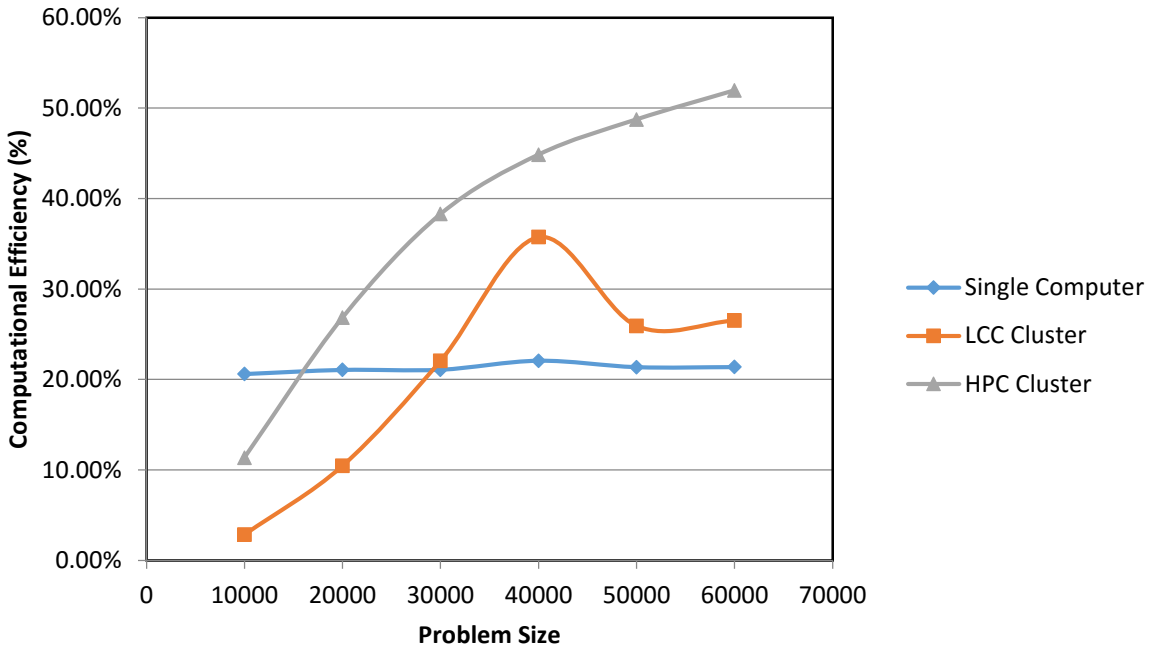


**Figure 1:** Computational efficiency

From Table 8 and Figure 1, it appears on average as if the single computer had the highest computational efficiency (20.61%) for a small problem size (n<10000). This is because there was no network fabric involved, thus removing certain overheads associated with the other two computing platforms. For the higher problem sizes (n>=30000), where the initial overheads are a smaller portion of the total time-to-complete, efficiency for both the LCC cluster (22.07%) and the HPC cluster (38.28%) surpasses the single computer (21.09%). When comparing the findings on computational efficiency, the single computer's computational efficiency did not increase with problem size as with the other two platforms. This was because the single computer had a small number of cores (n=4), which allowed the other platform to surpass the single computer in computational efficiency. It also appears, on average, that the HPC cluster at 51.97%, while still under-utilised by even the largest test (n=60000), outperformed the LCC cluster and single computer in terms of computational efficiency. With the larger problem sizes (n>=50000), the LCC cluster and single computer reached their maximum performance, with 26% and 21% respectively, while the HPC cluster could still increase in performance. The decline of computational efficiency for the LCC cluster from n=40000 to n=50000 was because of a network bottleneck. In the LCC cluster, there was a wide distribution of cores on a slower 1GB network, whereas the HPC cluster's cores were spread between fewer nodes with a much lower network latency. Core distribution and network speed played no role in the results for the single computer.

# 5. Discussion and conclusion

The aim of this study was to investigate the feasibility of using a single computer and an LCC cluster as alternatives to an HPC cluster for scientific computing in a developing country. This was done as a means to counter external pressure factors such as high energy prices and the initial investment required for an HPC. HPL benchmark applications were used to test the performance

of the three different computing platforms namely a single computer, an LCC cluster and an HPC cluster. The results of the benchmark tests were compared with respect to the time-to-complete, the energy consumption and operational costs of the three computing platforms.

The HPC cluster on average took less time-to-complete the HPL benchmark test (67.15s) than both the LCC cluster (879.3s) and the single computer (1699.93s). This makes the HPC cluster an attractive and valuable tool for research as it can solve more problems within a certain time span. However, the LCC cluster completed the HPL benchmark faster than the single computer, making it a viable alternative to a single computer. The more tasks a computer can complete within a timeframe, the more valuable it can be for research or commercial use.

While the HPC's power consumption per hour of 2152W (see Table 5), was much higher than that of the LCC cluster (1212W) or the single computer (119W), the HPC cluster on average used much less energy. Thus, with regard to a specific task, the HPC cluster on average consumed less energy (40.141W) than either the LCC cluster (296.031W) or the single computer (56.192W). On average, the HPC's theoretical energy consumption cost was 0.002USD, followed by the single computer (0.0038USD) and finally, the LCC with 0.02USD. The HPC was therefore the most cost effective computing platform and the LCC the least cost effective computing platform. Finally, for the largest problem size (n=60000), the HPC showed the highest computational efficiency (51.97%), followed by the LCC cluster (26.52%) and then the single computer (21.38%). This finding would suggest that a single computer's computational efficiently did not scale as the problem size increased as oppose to the HPC cluster and the LCC cluster.

It was also theoretically calculated that the LCC cluster would provide an organisation with 56 years of computing capacity before the costs would equalise the initial investment required for the HPC cluster. The operational cost of acquiring and retaining technical personal was not taken into the equation since both the LCC and HPC cluster would require these resources. This finding would indicate that an LCC cluster may theoretically have practical applications in environments, where the cost and computational efficiency factors for a particular problem size are more important than the time-to-complete factor. There are limitations to the situations where an LCC cluster could be useful. It would not be possible for a computing cluster to be up-and-running for nearly 56 years while working on a single scientific problem, without downtime or interruptions. Furthermore, the importance of solving an important scientific problem within the shortest possible timeframe outweighs the high acquisition cost of an HPC cluster.

This study found that there is a practical application for a single computer to conduct scientific computation as an alternative to an LCC cluster or an HPC cluster when the problem size was small (n=10000). A small scientific problem could be a stress test of a vehicle part by using a computer simulation. However, both the LCC cluster and HPC cluster demonstrated more efficiency for larger problems (n>=30000). For the largest problem size (n=60000), the HPC cluster had the highest efficiency (51.97%). As mentioned before, the application of the LCC is directly linked to the size of the problem. For this reason, a large scientific experiment would not be feasible on an LCC cluster as it could take years of uninterruptable computation to complete. A single fault or failure occurring in the cluster during these years could result in failure of the simulations and the data could possibly be lost. However, applications of a low-cost computing cluster could include small scale simulations of fluid dynamics, such as commonly used by

engineers all over the world. With regard to this example, a small engineering firm might not be able to put forward the large initial investment required for an HPC cluster or afford the high electricity costs associated with it. In this situation, it would be more feasible to use an LCC cluster to perform a simulation within a week than to acquire and deploy an expensive HPC cluster to complete the same simulation within a day. A single computer might take a month or more to complete the same simulation, assuming that the single computer has enough resources to perform the simulation at all. We also argue that an LCC could be an attractive low-cost solution for inexpensively storing and processing terabytes of data using Apache Hadoop ™ when taking into account its computational efficiency and time-to-complete. Hadoop cluster computing platforms are known for their ability to process batch data and an LCC would be a viable alternative as opposed to a single computer and HPC. However, since the focus of the study was to investigate scientific computing using an $n$ x $n$ dense linear system, further research is required in terms of processing large amounts of data on an LCC.

This study therefore suggests that LCC clusters could be a viable solution in an environment where the problems or programs that need to be solved or executed are small and costs should be kept to the minimum. The initial investment of USD4000.00 for an LCC cluster makes it an attractive solution for any developing country. The electricity costs of an LCC cluster are negligible compared to the large investment in acquiring an HPC cluster. However, in an environment where performance and time are of more importance than costs, and funding is available, an HPC cluster is still the preferred solution as it offers the best efficiency for both theoretical energy consumption and computation.

## 6. Acknowledgements

The authors would like to thank Fanie Riekerk and Albert van Eck for their assistance during the collection of data from the University of the Free State's High Performance Computing Cluster. The authors would also like to thank the anonymous reviewers for their valuable comments and suggestions on the paper.

## *References*

Amazon.com. (2014). Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more. Retrieved October 30, 2014, from http://www.amazon.com/

Apon, A., Buyya, R., Jin, H., & Mache, J. (2001). Cluster computing in the classroom: Topics, guidelines, and experiences. In *Proceedings - 1st IEEE/ACM International Symposium on Cluster Computing and the Grid*, CCGrid 2001, pp. 476–483. IEEE.

Bates, N. J., & Patterson, M. K. (2013). Achieving the 20MW target: Mobilizing the HPC community to accelerate energy efficient computing. *Advances in Parallel Computing*, 24, 37–45.

Bergman, M., Funston, J., & Gilfeather-Crowley, P. (2009). Low cost compute clusters in virtualized lab environments. *Journal of Computing Sciences in Colleges*, 25(1), 159–166.

Bibliotheca Alexandrina. (2012). Supercomputer - Bibliotheca Alexandrina. Retrieved August 3, 2015, from http://www.bibalex.org/en/Project/Details?DocumentID=281

Bollinger, T. (1999). Linux in practice: An overview of applications. *Software, IEEE*, 16(1), 72–79.

Centre for High Performance Computing. (n.d.). Previous flagship projects. Retrieved August 3, 2015, from http://www.chpc.ac.za/index.php/research-and-collaboration/flagship-projects/2-uncategorised/110-previous-flagship-projects

Davies, T., Karlsson, C., Liu, H., Ding, C., & Chen, Z. (2011). High performance linpack benchmark: A fault tolerant implementation without checkpointing. In *Proceedings of the International Conference on Supercomputing*, 162–171.

Dongarra, J. J., & Van der Steen, A. J. (2012). High-performance computing systems: Status and outlook. *Acta Numerica*, 21, 379–474.

Dorband, J., Palencia, J., & Ranawake, U. (2003). Commodity computing clusters at goddard space flight center. *Journal of Space Communication*, *1*(3), 113–123.

Exchange-Rates.org. (2015). Echange Rates. Retrieved December 10, 2014, from http://www.exchange-rates.org/Rate/USD/ZAR/12-10-2014

Foster, I. (1995). Parallelism and Computing. Retrieved August 20, 2015, from http://www.mcs.anl.gov/~itf/dbpp/text/node7.html

Halliday, D., Resnick, R., & Walker, J. (2007). *Fundamentals of Physics*, 8th edition. John Wiley & Sons, Inc.

Hitachi. (2014). Calculation of ideal energy consumption (theoretical value) and energy consumption (theoretical value). Retrieved August 20, 2015, from http://itdoc.hitachi.co.jp/manuals/3021/3021333710e/DMDS0136.HTM

Intel. (2015). Intel Parallel Studio XE 2015. Retrieved February 10, 2015, from https://software.intel.com/en-us/intel-parallel-studio-xe/try-buy#buynow

Jeun, W.-C., Kee, Y.-S., Kim, J.-S., & Ha, S. (2003). A High Performance and Low Cost Cluster-Based E-mail System. In *Parallel Computing Technologies*. Springer Berlin Heidelberg.

Lang, W., Patel, J. M., & Shankar, S. (2010). Wimpy node clusters: What About Non-Wimpy Workloads? In *Proceedings of the Sixth International Workshop on Data Management on New Hardware - DaMoN '10* (Vol. 1, pp. 47–55). ACM.

Mai, G. C., & De Rose, C. A. (2000). Low cost cluster architectures for parallel and distributed processing. *CLEI Electronic Journal*, 3(1).

Microsoft. (2013). Lizard Results. Retrieved August 20, 2015, from https://msdn.microsoft.com/en-us/library/ee146531%28v=ws.10%29.aspx

Middleton, A. (2011). HPCC Systems: Introduction to HPCC (High-Performance Computing Cluster). *LexisNexis Risk Solutions*, White Paper.

Minelli, M., Chambers, M., & Dhiraj, A. (2013). *Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses*. Hoboken, NJ, US: John Wiley & Sons, Inc.

Misbahuddin, S., Noor, F., & Zugail, S. (2008). Parallel Processing on Pentium Machines : a Low cost Solution for High Performance Computing. *Journal of Information & Communication Technology*, 2(2), 116–125.

Naik, L. S. (2012). Designing a Low Cost and Scalable PC Cluster System for HPC Environment. In *Proceedings on International Conference on Advances in Communication and Computing Technologies 2012,* 23–25.

NASA. (n.d.). NASA Project Portfolio. Retrieved February 19, 2015, from http://www.nas.nasa.gov/projects/projects.html

Nersa. (2014). Eskom Future Pricepath. Retrieved January 15, 2015, from http://www.nersa.org.za/Admin/Document/Editor/file/Electricity/PricingandTariffs/Eskom Future Pricepath.pdf

Netlib.org. (2015a). HPL Algorithm. Retrieved January 21, 2015, from http://www.netlib.org/benchmark/hpl/algorithm.html

Netlib.org. (2015b). HPL Tuning. Retrieved February 3, 2015, from http://www.netlib.org/benchmark/hpl/tuning.html

Novatte. (2014). How to Calculate Peak Theoretical Performance of a CPU-based HPC System. Retrieved August 25, 2015, from http://www.novatte.com/our-blog/197-how-to-calculate-peak-theoretical-performance-of-a-cpu-based-hpc-system

Nyberg, P. (2013). The Critical Role of Supercomputers in Weather Forecasting. Retrieved February 19, 2015, from http://www.cray.com/blog/the-critical-role-of-supercomputers-in-weather-forecasting/

Petitet, A., Whaley, R., Dongarra, J., & Cleary, A. (2014). HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers. Retrieved March 2, 2016, from http://www.netlib.org/benchmark/hpl/

Sterling, T., Becker, D. J., Savarese, D., Dorband, J. E., Ranawake, U. a., & Packer, C. V. (1995). Beowulf: A Parallel Workstation For Scientific Computation. In *Proceedings of the 24th International Conference on Parallel Processing.*

Sun, X. H., & Chen, Y. (2010). Reevaluating Amdahl's law in the multicore era. *Journal of Parallel and Distributed Computing*, 70(2), 183–188.

United States Embassy in South Africa. (2014). South Africa Economic Fact Sheet. Retrieved October 3, 2014, from http://photos.state.gov/libraries/southafrica/231771/PDFs/economic-fact-sheet_June-2014.pdf

Vecchiatto, P. (2013). Electricity tariff rises have hurt economy, says energy minister. Retrieved October 3, 2014, from http://www.bdlive.co.za/business/energy/2013/10/28/electricity-tariff-rises-have-hurt-economy-says-energy-minister