Performance Analysis of Single Board Computer Clusters

Philip J. Basford[a], Steven J. Johnston[a,*], Colin S. Perkins[b], Tony Garnock-Jones[b], Fung Po Tso[c], Dimitrios Pezaros[b], Robert D. Mullins[d], Eiko Yoneki[d], Jeremy Singer[b], Simon J. Cox[a]

[a] *Faculty of Engineering & Physical Sciences, University of Southampton, Southampton, SO16 7QF, UK.*
[b] *School of Computing Science, University of Glasgow, Glasgow, G12 8QQ, UK.*
[c] *Department of Computer Science, Loughborough University, Loughborough, LE11 3TU, UK.*
[d] *Computer Laboratory, University of Cambridge, Cambridge, CB3 0FD, UK.*

Abstract

The past few years have seen significant developments in Single Board Computer (SBC) hardware capabilities. These advances in SBCs translate directly into improvements in SBC clusters. In 2018 an individual SBC has more than four times the performance of a 64-node SBC cluster from 2013. This increase in performance has been accompanied by increases in energy efficiency (GFLOPS/W) and value for money (GFLOPS/$). We present systematic analysis of these metrics for three different SBC clusters composed of Raspberry Pi 3 Model B, Raspberry Pi 3 Model B+ and Odroid C2 nodes respectively. A 16-node SBC cluster can achieve up to 60 GFLOPS, running at 80 W. We believe that these improvements open new computational opportunities, whether this derives from a decrease in the physical volume required to provide a fixed amount of computation power for a portable cluster; or the amount of compute power that can be installed given a fixed budget in expendable compute scenarios. We also present a new SBC cluster construction form factor named Pi Stack; this has been designed to support edge compute applications rather than the educational use-cases favoured by previous methods. The improvements in SBC cluster performance and construction techniques mean that these SBC clusters are realising their potential as valuable developmental edge compute devices rather than just educational curiosities.

Keywords: Raspberry Pi, Edge Computing, Multicore Architectures, Performance Analysis

1. Introduction

Interest in Single Board Computer (SBC) clusters has been growing since the initial release of the Raspberry Pi in 2012 [1]. Early SBC clusters, such as Iridis-Pi [2], were aimed at educational scenarios, where the experience of working with, and managing, a compute cluster was more important than its performance. Education remains an important use case for SBC clusters, but as the community has gained experience, a number of additional use cases have been identified, including edge computation for low-latency, cyber-physical systems and the Internet of things, and next generation data centres [3].

The primary focus of these use cases is in providing location-specific computation. That is, computation that is located to meet some latency bound, that is co-located with a device under control, or that is located within some environment being monitored. Raw compute performance matters, since the cluster must be able to support the needs of the application, but power efficiency (GFLOPS/W), value for money (GFLOPS/$), and scalability can be of equal importance.

In this paper, we analyse the performance, efficiency, value-for-money, and scalability of modern SBC clusters implemented using the Raspberry Pi 3 Model B [4] or Raspberry Pi 3 Model B+ [5], the latest technology available from the Raspberry Pi Foundation, and compare their performance to a competing platform, the Odroid C2 [6]. Compared to early work by Gibb [7] and Papakyriakou et al. [8], which showed that early SBC clusters were not a practical option because of the low compute performance offered, we show that performance improvements mean that for the first time SBC clusters have moved from being a curiosity to a potentially useful technology.

To implement the SBC clusters analysed in this paper, we developed a new SBC cluster construction technique: the Pi Stack. This is a novel power distribution and control board, allowing for increased cluster density and improved power proportionality and control. It has been developed taking the requirements of edge compute deployments into account, to enable these SBC clusters to move from edu-

[*] Corresponding author

Email addresses: p.j.basford@soton.ac.uk (Philip J. Basford), sjj698@zepler.org (Steven J. Johnston), csp@csperkins.org (Colin S. Perkins), tonyg@leastfixedpoint.com (Tony Garnock-Jones), p.tso@lboro.ac.uk (Fung Po Tso), dimitrios.pezaros@glasgow.ac.uk (Dimitrios Pezaros), robert.mullins@cl.cam.ac.uk (Robert D. Mullins), eiko.yoneki@cl.cam.ac.uk (Eiko Yoneki), jeremy.singer@glasgow.ac.uk (Jeremy Singer), sjc@soton.ac.uk (Simon J. Cox)

cational projects to useful compute infrastructure.

We structure the remainder of this paper as follows. The motivations for creating SBC clusters and the need for analysing their performance are described in Section 2. A new SBC cluster creation technique is presented and evaluated in Section 3. The process used for the benchmarking and the results obtained from the performance benchmarks are described in Section 4. As well as looking at raw performance, power usage analysis of the clusters is presented in Section 5. The results are discussed in Section 6, Section 7 describes how this research could be extended. Finally Section 8 concludes the paper.

2. Context and rationale

The launch of the Raspberry Pi in 2012 [1] popularised the use of SBCs. Until this point SBCs were available but were not as popular. Since 2012 over 19 million Raspberry Pis have been sold [9]. In tandem with this growth in the SBC market there has been developing interest in using these SBC to create clusters [2, 10, 11]. There are a variety of different use-cases for these SBC clusters which can be split into the following categories: education, edge compute, expendable compute, resource constrained compute, next-generation data centres and portable clusters [3].

The ability to create a compute cluster for approximately the cost of workstation [2] has meant that using and evaluating these micro-data centres is now within the reach of student projects. This enables students to gain experience constructing and administering complex systems without the financial outlay of creating a data centre. These education clusters are also used in commercial research and development situations to enable algorithms to be tested without taking up valuable time on the full scale cluster [12].

The practice of using a SBC to provide processing power near the data-source is well established in the sensor network community [13, 14]. This practice of having compute resources near to the data-sources known as Edge Compute [15] can be used to reduce the amount of bandwidth needed for data transfer. By processing data at the point of collection privacy concerns can also be addressed by ensuring that only anonymized data is transferred.

These edge computing applications have further sub categories; expendable and resource constrained compute. The low cost of SBC clusters means that the financial penalty for loss or damage to a device is low enough that the entire cluster can be considered expendable. When deploying edge compute facilities in remote locations the only available power supplies are batteries and renewable sources such as solar energy. In this case the power efficiency in terms of GFLOPS/W is an important consideration. While previous generations of Raspberry Pi SBC have been measured to determine their power efficiency [11], the two most recent Raspberry Pi releases have not been evaluated previously. The location of an edge compute infrastructure may mean that maintenance or repairs are not practical. In such cases the ability to over provision the compute resources means spare SBCs can be installed and only powered when needed. This requires the ability to individually control the power for each SBC. Once this power control is available it can also be used to dynamically scale the cluster size depending on current conditions.

The energy efficiency of a SBC cluster is also important when investigating the use of SBCs in next-generation data centres. This is because better efficiency allows a higher density of processing power and reduces the cooling capacity required within the data centre. When dealing with the quantity of SBC that would be used in a next-generation data centre the value for money in terms of GFLOPS/$ is important.

The small size of SBCs which is beneficial in data centres to enable maximum density to be achieved also enables the creation of portable clusters. These portable clusters could vary in size from those requiring a vehicle to transport to a cluster than can be carried by a single person in a backpack. A key consideration of these portable clusters is their ruggedness. This rules out the construction techniques of Lego and laser-cut acrylic used in previous clusters [2, 16].

Having identified the potential use cases for SBC clusters the different construction techniques used to create clusters can be evaluated. Iridis-Pi [2], Pi Cloud [10], the Mythic Beast cluster [16], the Los Alamos cluster [12] and the Beast 2 [17] are all designed for the education use case. This means that they are lacking features that would be beneficial when considering the alternative use cases available. The feature of each cluster construction technique are summarised in Table 1. The use of Lego as a construction technique for Iridis-Pi was a solution to the lack of mounting holes provided by the Raspberry Pi 1 Model B [2]. Subsequent versions of the Raspberry Pi have adhered to an updated standardised mechanical layout which has four M2.5 mounting holes [18]. The redesign of the hardware interfaces available on the Raspberry Pi also led to a new standard for Raspberry Pi peripherals called the Hardware Attached on Top (HAT) [19]. The mounting holes have enabled many new options for mounting, such as using 3D-printed parts like the Pi Cloud. The Mythic Beasts cluster uses laser-cut acrylic which is ideal for their particular use case in 19-inch racks, but is not particularly robust. The most robust cluster is that produced by BitScope for Los Alamos which uses custom Printed Circuit Boards (PCBs) to mount the Raspberry Pi enclosed in a metal rack.

Following the release of the Raspberry Pi 1 Model B in 2012 [1], there have been multiple updates to the platform released which as well as adding additional features have increased the processing power from a 700 MHz single core Central Processing Unit (CPU) to a 1.4 GHz quad-core CPU [20]. These processor upgrades have increased the available processing power, and they have also increased the power demands of the system. This increase in power

Table 1: Comparison of different Raspberry Pi cluster construction techniques.

Feature	Iridis Pi	Pi Cloud	Mythic Beasts	Los Alamos	The Beast 2	Pi Stack
Material	Lego	3D printed plastic	Laser cut acrylic	Custom PCB	Laser cut acrylic	Custom PCB
Requires Power over Ethernet (PoE) switch?	No	No	Yes	No	No	No
Individual Power control	No	No	Yes	Unknown	No	Yes
Individual Power monitoring	No	No	Yes	No	No	Yes
Individual Heartbeat monitoring	No	No	No	No	No	Yes
Raspberry Pi Version	1B	3B	3B	B+/2B/ 3B/0	3B	A+/B+/2B/ 3B/3B+/3A/ 0/0W
Cooling	Passive	Passive	Active	Active	Passive	Active
DC input Voltage	5V	5V	48V (PoE)	9V - 48V	12V	12V - 30V
Ruggedness	Poor	Good	Medium	Good	Medium	Good
Density	High	Medium	Medium	High	Low	High

draw leads to an increase in the heat produced. Iridis-Pi used entirely passive cooling and did not observe any heat-related issues [2]. All the Raspberry Pi 3 Model B clusters listed in Table 1 that achieve medium- or high-density use active cooling.

The evolution of construction techniques from Lego to custom PCB and enclosures shows that these SBC have moved from curiosities to commercially valuable prototypes of the next generation of Internet of Things (IoT) and cluster technology. As part of this development a new SBC cluster construction technique is required to enable the use of these clusters in use cases other than education.

3. Pi Stack

The Pi Stack is a new SBC construction technique that has been developed to build clusters supporting the use cases identified in Section 2. It features high-density, individual power control, heartbeat monitoring, and reduced cabling compared to previous solutions. The feature set of the Pi Stack is summarised in Table 1.

3.1. Key Features

These features have been implemented using a PCB which measures 65.0 mm× 69.5 mm and is designed to fit between two Raspberry Pi boards facing opposite directions. The reason for having the Raspberry Pis in opposite directions is it enables two Raspberry Pis to be connected to each Pi Stack PCB and it enables efficient tessellation of the Raspberry Pis, maximising the number of boards that can be fitted in a given volume to give **high density**. Technical drawings showing the construction technique for a 16 node cluster and key components of the Pi Stack, are shown in Figure 1. The PCB layout files are published under the CC-BY-SA-4.0 license [21].

Individual power control for each SBC means that any excess processing power can be turned off, therefore reducing the energy demand. The instantaneous power demand of each SBC can be measured meaning that when used on batteries the expected remaining uptime can be calculated.

The provision of **heartbeat monitoring** on Pi Stack boards enables it to detect when an attached SBC has failed, this means that when operating unattended the management system can decide to how to proceed given the hardware failure, which might otherwise go un-noticed leading to wasted energy. To provide flexibility for power sources the Pi Stack has a wide range input, this means it is compatible with a variety of battery and energy harvesting techniques. This is achieved by having on-board voltage regulation. The input voltage can be measured enabling the health of the power supply to be monitored and the cluster to be safely powered down if the voltage drops below a set threshold.

The Pi Stack offers **reduced cabling** by injecting power into the SBC cluster from a single location. The metal stand-offs that support the Raspberry Pi boards and the Pi Stack PCBs are then used to distribute the power and management communications system throughout the cluster. To reduce the current flow through these stand-offs, the Pi Stack accepts a range of voltage inputs, and has on-board regulators to convert its input to the required 3.3 V and 5 V output voltages. The power supply is connected directly to the stand-offs running through the stack by using ring-crimps between the stand-offs and the Pi Stack PCB. In comparison to a PoE solution, such as the Mythic

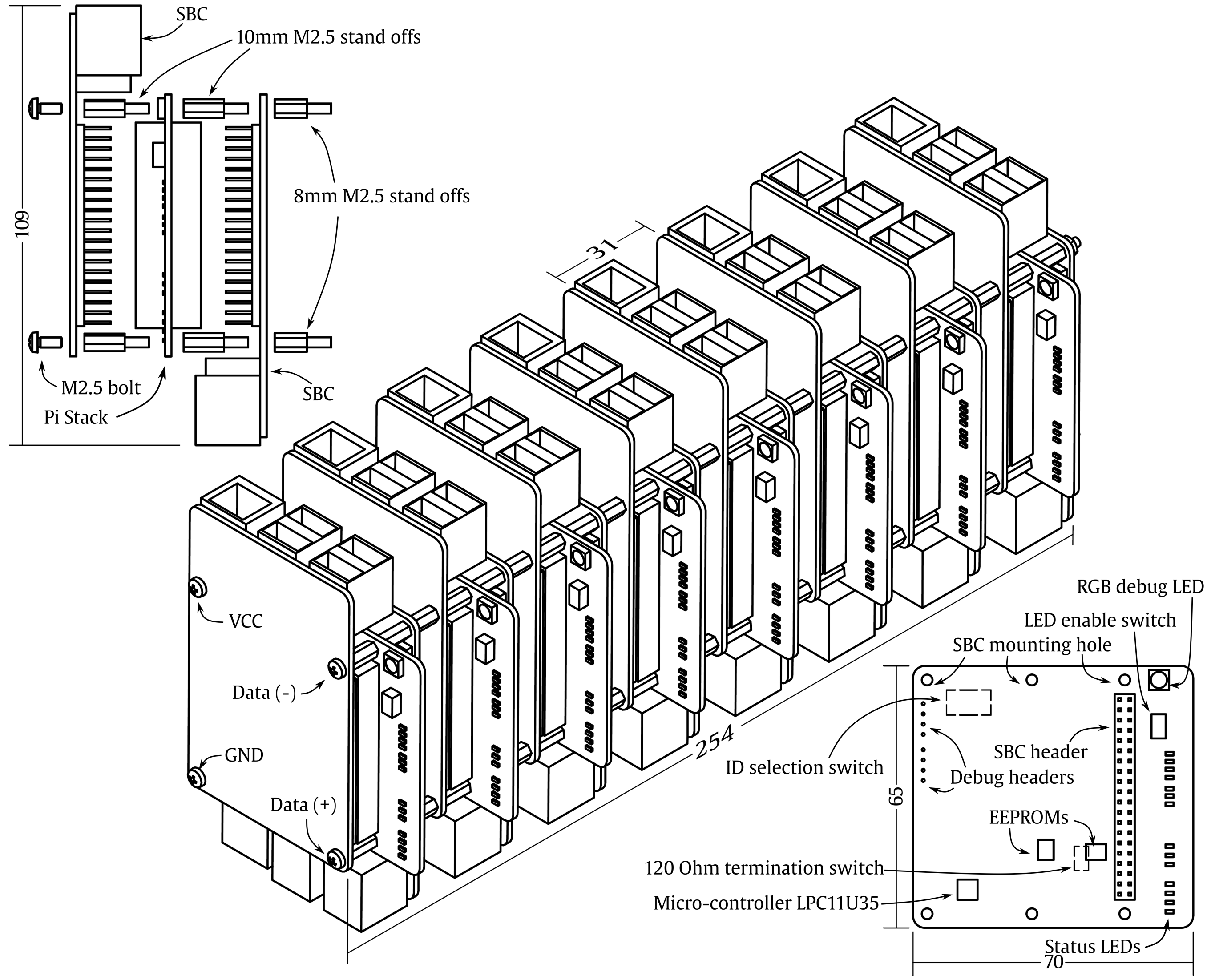


Figure 1: Details of the Pi Stack Cluster, illustrated with Raspberry Pi 3 Model B Single Board Computers (SBCs). a) Exploded side view of a pair of SBCs facing opposite directions, with Pi Stack board. b) A full stack of 16 nodes, with Pi Stack boards. c) Pi Stack board with key components.

Beasts cluster, the Pi Stack maintains cabling efficiency and reduces cost since it does not need a PoE HAT for each Raspberry Pi, and because it avoids the extra cost of a PoE network switches compared to standard Ethernet.

The communication between the nodes of the cluster is performed using the standard Ethernet interfaces provided by the SBC. An independent communication channel is needed to manage the Pi Stack boards and does not need to be a high speed link.

This management communication bus is run up through the mounting posts of the stack requiring a multi-drop protocol. The communications protocol used is based on RS485 [22], a multi-drop communication protocol that supports up to 32 devices, which sets a maximum number of Pi Stack boards that can be connected together. RS485 uses differential signalling for better noise rejection, and supports two modes of operation: full-duplex which requires four wires, or half-duplex which requires two wires. The availability of two conductors for the communication system necessitated using the half-duplex communication mode. The RS485 specification also mandates an impedance between the wires of 120 Ω, a requirement of the RS485 standard which the Pi Stack does not meet. The mandated impedance is needed to enable RS485 communications at up to 10 Mbit/s or distances of up to ≈1200 m; however, because the Pi Stack requires neither long-distance nor high-speed communication, this requirement can be relaxed. The communication bus on the Pi Stack is configured to run at a baud rate of 9600 bit/s, which means the longest message (6 B) takes 6.25 ms to be transmitted.

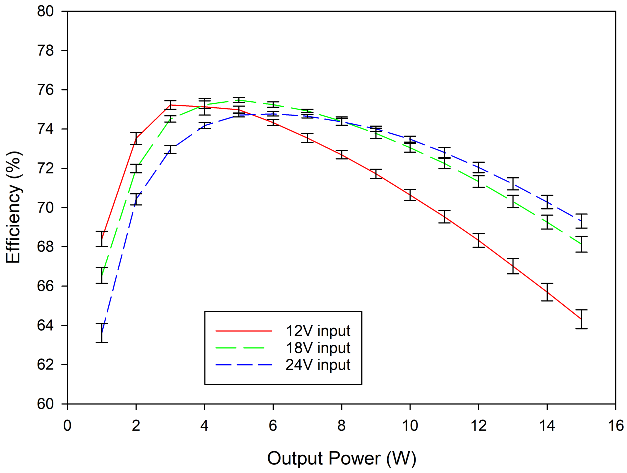


Figure 2: Pi Stack power efficiency curves for different input voltages and output loads. Measured using a BK Precision 8601 dummy load. Note Y-axis starts at 60 %. Error bars show 1 standard deviation.

The RS485 standard details the physical layer and does not provide details of communication protocol running on the physical connection. This means that the data transferred over the link has to be specified separately. As RS485 is multi-drop, any protocol implemented using it needs to include addressing to identify the destination device. As different message types require data to be transferred, the message is variable length. to ensure that the message is correctly received it also includes a CRC8 field [23]. Each field in the message is 8-bits wide. Other data types are split into multiple 8-bit fields for transmission and are reassembled by receivers. As there are multiple nodes on the communication bus, there is the potential for communication collisions. To eliminate the risk of collisions, all communications are initiated by the master, while Pi Stack boards only respond to a command addressed to them. The master is defined as the main controller for the Pi Stack. This is currently a computer (or a Raspberry Pi) connected via a USB-to-RS485 converter.

When in a power constrained environment such as edge computing the use of a dedicated SBC for control would be energy inefficient. In such a case a separate PCB containing the required power supply circuitry and a micro controller could be used to coordinate the use of the Pi Stack.

3.2. Power Supply Efficiency

The Pi Stack accepts a wide range of input voltages means that be on-board regulation is required to provide power both for Pi Stack control logic, running at 3.3 V, and for each SBC, running at 5 V. Voltage regulation is never 100 % efficient as some energy will be dissipated as heat. This wasted energy needs to be minimised for two specific reasons: i) to provide maximum lifetime when used on stored energy systems for example solar panels, ii) to reduce the amount of heat to minimise cooling requirements. As the 3.3 V power supply is used to power the Analog Digial Converter (ADC) circuitry used for voltage and current measurements, it needs to be as smooth as possible. It was implemented as a switch mode Power Supply Unit (PSU) to drop the voltage to 3.5 V before a low-dropout regulator to provide the smooth 3.3 V supply. This is more efficient than using an low-dropout regulator direct from the input voltage and so therefore produces less heat. The most efficient approach for this power supply would be to use a switch-mode PSU to directly provide the 3.5 V supply, which was ruled out because the micro controller used uses the main power supply as the analogue reference voltage.

Table 2: Idle power consumption of the Pi Stack board. Measured with the LEDs and Single Board Computer (SBC) Power Supply Unit (PSU) turned off.

Input Voltage (V)	Power Consumption (mW)	Standard Deviation
12	82.9	4.70
18	91.2	6.74
24	100.7	8.84

This multi-stage approach was deemed unnecessary for the SBC PSU because the 5 V rail is further regulated before it is used. As the efficiency of the power supply is dependent on the components used this has been measured on three Pi Stack boards which has shown an idle power draw of 100.7 mW at 24 V (other input voltages are shown in Table 2), and a maximum efficiency of 75.5 % which is achieved at a power draw of 5 W at 18 V as shown in Figure 2. Optimal power efficiency of the Pi Stack PSU is achieved when powering a single SBC drawing ≈5 W. When powering two SBCs, the PSU starts to move out of the optimal efficiency range. The drop in efficiency is most significant for a 12 V input, with both 18 V and 24 V input voltages achieving better performance. This means that when not using the full capacity of the cluster it is most efficient to distribute the load amongst the nodes so that it is distributed between as many Pi Stack PCBs, and therefore PSUs as possible. This is particularly important for the more power hungry SBC that the Pi Stack supports.

3.3. Thermal Design Considerations

The Pi Stack uses the inversion of half the SBCs in the stack to increase the density. This in turn increases the density of heat production within the cluster. Iridis-Pi and the Beast 2 (see Table 1) have been able to operate at ambient temperatures with no active cooling the Pi Stack clusters require active cooling. The heat from the PSU attached to each Pi Stack PCB also contributes to the heating of the cluster. The alternative of running a 5 V feed to each SBC would reduce the heat from the PSU, but more

Table 3: Comparison between Single Board Computer (SBC) boards. The embedded Multi-Media Card (eMMC) module for the Odroid C2 costs $15 for 16 GB. Prices correct April 2019.

	Raspberry Pi 3 Model B	Raspberry Pi 3 Model B+	Odroid C2
Processor cores	4	4	4
Processor speed (GHz)	1.2	1.4	1.5
RAM (GB)	1	1	2
Network Speed (Mbit/s)	100	1000	1000
Network Connection	USB2	USB2	Direct
Storage	micro-SD	micro-SD	micro-SD / eMMC
Operating System	Raspbian Stretch Lite	Raspbian Stretch Lite	Ubuntu 18.04.1
Price (USD)	$35	$35	$46

heat would be generated by the power distribution infrastructure due to running at higher currents. This challenge is not unique to Pi Stack based SBC clusters as other clusters such as Mythic Beasts and Los Alamos (see Table 1) also require active cooling. This cooling has been left independent of the Pi Stack boards as the cooling solution required will be determined by the application environment. The cooling arrangement used for the benchmark tests presented in this paper is further discussed in Section 6.2.

3.4. Pi Stack Summary

Three separate Pi Stack clusters of 16 nodes have been created to perform the tests presented in Section 4 and Section 5. The process of experimental design and execution of these tests over 1,500 successful High-Performance Linpack (HPL) benchmarks. During initial testing it was found that using a single power connection did not provide sufficient power for the successful completion of large problem size HPL runs. This was remedied by connecting power leads at four different points in the stack, reducing the impedance of the power supply. The arrangement means 16 nodes can be powered with four pairs of power leads, half the number that would be required without using the Pi Stack system. Despite this limitation of the Pi Stack it has proven to be a valuable cluster construction technique, and has facilitated new performance benchmarks.

4. Performance Benchmarking

The HPL benchmark suite [24] has been used to both measure the performance of the clusters created and to fully test the stability of the Pi Stack SBC clusters. HPL was chosen to enable comparison between the results gathered in this paper and results from previous studies into clusters of SBCs. HPL has been used since 1993 to benchmark the TOP500 supercomputers in the world [25]. The TOP500 list is updated twice a year in June and November.

HPL is a portable and freely available implementation of the High Performance Computing Linpack Benchmark [26]. HPL solves a random dense linear system to measure the Floating Point Operations per Second (FLOPS) of the system used for the calculation. HPL requires Basic Linear Algebra Substem (BLAS), and initially the Raspberry Pi software library version of Automatically Tuned Linear Algebra Software (ATLAS) was used. This version of ATLAS unfortunately gives very poor results, as previously identified by Gibb [7], who used OpenBLAS instead. For this paper the ATLAS optimisation was run on the SBC to be tested to see how the performance changed with optimisation. There is a noticeable performance increase, with the optimised version being over 2.5 times faster for 10 % memory usage and over three times faster for problem sizes ≥20 %, results from a average over three runs. The process used by ATLAS for this optimisation is described by Whaley and Dongarra [27]. For every run of a HPL benchmark a second stage of calculation is performed. This second stage is used to calculate residuals which are used to verify that the calculation succeeded, if the residuals are over a set threshold then the calculation is reported as having failed.

$$\sqrt{\frac{Node_{count} * Node_{RAM} * 1024^3}{8}} * RAM_{usage} \qquad (1)$$

HPL also requires a Message Passing Interface (MPI) implementation to co-ordinate the multiple nodes. These tests used MPICH [28]. When running HPL benchmarks, small changes in configuration can give big performance changes. To minimise external factors in the experiments, all tests were performed using a version of ATLAS compiled for that platform, HPL and MPICH were also compiled, ensuring consistency of compilation flags. The HPL configuration and results are available from `doi:10.5281/zenodo.2002730`. The only changes between runs of HPL were to the parameters N, P and Q, which were changed to reflect the number of nodes in the cluster. HPL is highly configurable, allowing multiple interlinked parameters to be adjusted to achieve maximum performance from the system under test. The guidelines from the HPL FAQ [29] have been followed regarding P and Q being "approximately equal, with Q slightly larger than P". Further, the problem sizes have been chosen to be a multiple of the block size. The problem size is the size of the square matrix that the program attempts to solve. The optimal problem size for a given cluster is calculated according to Equation 1, and then rounded down to a multiple of the block size [30]. The equation uses the number of nodes ($\text{Node}_{\text{count}}$) and the amount of Random Access Memory (RAM) in GB (Node_{RAM}) to calculate the total amount of RAM available in the cluster in B. The number of double precision values (8 B) that can be stored in this space is calculated. This gives the number of elements in the matrix, the square root gives the length of the matrix size. Finally this matrix size is scaled to occupy the requested amount of total RAM. To measure the performance as the problem size increases, measurements have been taken at 10% steps of memory usage, starting at 10% and finishing at 80%. This upper limit was chosen to allow some RAM to be used by the Operating System (OS), and is consistent with the recommended parameters [31].

4.1. Experimental set-up used

The use of the Pi Stack requires all the boards to be the same form factor as the Raspberry Pi. During the preparations for this experiment the Raspberry Pi 3 Model B+ was released [9]. The Raspberry Pi 3 Model B+ has better processing performance than the Raspberry Pi 3 Model B, and has also upgraded the 100 Mbit/s network interface to 1 Gbit/s. The connection between the network adapter and the CPU remains USB2 with a bandwidth limit of 480 Mbit/s [32]. These improvements have led to an increase in power usage the Raspberry Pi 3 Model B+ when compared to the Raspberry Pi 3 Model B. As the Pi Stack had been designed for the Raspberry Pi 3 Model B, this required a slight modification to the PCB. This was performed on all the boards used in these cluster experiments. An iterative design process was used to develop the PCB, with versions 1 and 2 being in produced in limited numbers and hand soldered. Once the design was tested with version 3 a few minor improvements were made before a larger production run using pick and place robots was ordered. To construct the three 16-node clusters, a combination of Pi Stack version 2 and Pi Stack version 3 PCBs was used. The changes between version 2 and 3 mean that version 3 uses 3 mW more power than the version 2 board when the SBC PSU is turned on, given the power demands of the SBC this is insignificant.

Having chosen two Raspberry Pi variants to compare, a board from a different manufacturer was required to see if there were advantages to choosing an SBC from outside the Raspberry Pi family. As stated previously, to be compatible with the Pi Stack, the SBC has to have the same form factor as the Raspberry Pi. There are several boards that have the same mechanical dimensions but have the area around the mounting holes connected to the ground plane, for example the Up board [33]. The use of such boards would connect the stand-offs together breaking the power and communication buses. The Odroid C2 was identified as a suitable board as it meets both the two-dimensional mechanical constraints and the electrical requirements [6]. The Odroid C2 has a large metal heat sink mounted on top of the CPU, taking up space needed for Pi Stack components. Additional stand-offs and pin-headers were used to permit assembly of the stack. The heartbeat monitoring system of the Pi Stack is not used because early tests showed that this introduced a performance penalty.

As shown in Table 3, as well as having different CPU speeds, there are other differences between the boards investigated. The Odroid C2 has more RAM than either of the Raspberry Pi boards. The Odroid C2 and the Raspberry Pi 3 Model B+ both have gigabit network interfaces, but they are connected to the CPU differently. The network card for the Raspberry Pi boards is connected using USB2 with a maximum bandwidth of 480 Mbit/s, while the network card in the Odroid board has a direct connection. The USB2 connection used by the Raspberry Pi 3 Model B+ means that it is unable to utilise the full performance of the interface. It has been shown using iperf that the Raspberry Pi 3 Model B+ can source 234 Mbit/s and receive 328 Mbit/s, compared to 932 Mbit/s and 940 Mbit/s for the Odroid C2, and 95.3 Mbit/s and 94.2 Mbit/s for the Raspberry Pi 3 Model B [34]. The other major difference is the storage medium used. The Raspberry Pi boards only support micro SD cards, while the Odroid also supports embedded Multi-Media Card (eMMC) storage. eMMC storage supports higher transfer speeds than micro SD cards. Both the Odroid and the Raspberry Pi boards are using the manufacturer's recommended operating system, (Raspbian Lite and Ubuntu, respectively) updated to the latest version before running the tests. The decision to use the standard OS was made to ensure that the boards were as close to their default state as possible.

The GUI was disabled, and the HDMI port was not connected, as initial testing showed that use of the HDMI port led to a reduction in performance. All devices are using the `performance` governor to disable scaling. The graphics card memory split on the Raspberry Pi devices is set to 16 MB. The Odroid does not offer such fine-grained control over the graphics memory split. If the graphics are enabled, 300 MB of RAM are reserved; however, setting `nographics=1` in `/media/boot/boot.ini.default` disables this and releases the memory.

For the Raspberry Pi 3 Model B, initial experiments showed that large problem sizes failed to complete successfully. This behaviour has also been commented on in the Raspberry Pi forums [35]. The identified solution is to alter the boot configuration to include `over_voltage=2`, this increases the CPU/Graphical Processing Unit (GPU) core voltage by 50 mV to stop the memory corruption causing these failures occurring. The firmware running on the Raspberry Pi 3 Model B was also updated using `rpi-update`, to ensure that it was fully up to date in case of any stability improvements. These changes enabled the Raspberry Pi 3 Model B to successfully complete large HPL problem sizes. This change was not needed for the Raspberry Pi 3 Model B+. OS images

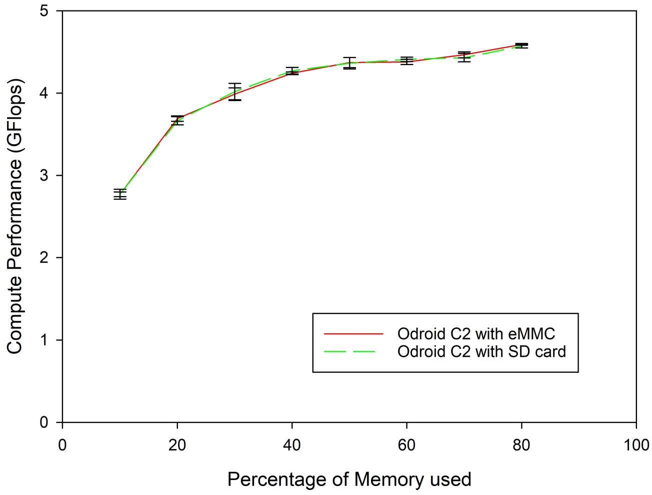


Figure 3: Comparison between an Odroid C2 running with a micro SD card and an embedded Multi-Media Card (eMMC) for storage. The lack of performance difference between storage mediums despite their differing speeds shows that the benchmark is not limited by Input / Output (I/O) bandwidth. Higher values show better performance. Error bars show one standard deviation.

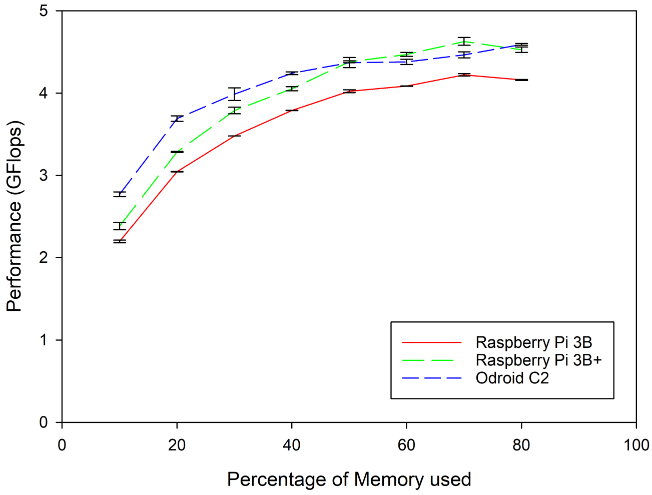


Figure 4: Comparison between the performance of a single Raspberry Pi 3 Model B, Raspberry Pi 3 Model B+, and Odroid C2. Higher values show better performance. Error bars show one standard deviation.

for all three platforms are included in the dataset.

All tests were performed on an isolated network connected via a Netgear GS724TP 24 port 1 Gbit/s switch which was observed to consume 26 W during a HPL run. A Raspberry Pi 2 Model B was also connected to the switch which ran Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), and Network Time Protocol (NTP) servers. This Raspberry Pi 2 Model B had a display connected and was used to control the Pi Stack PCBs and to connect to the cluster using Secure Shell (SSH). This Raspberry Pi 2 Model B was powered separately and is excluded from all power calculations, as it is part of the network infrastructure.

The eMMC option for the Odroid offers faster Input / Output (I/O) performance than using a micro-SD card. The primary aim of HPL is to measure the CPU performance of a cluster, and not to measure the I/O performance. This was tested by performing a series of benchmarks on an Odroid C2 first with a UHS1 micro-SD card and then repeating with the same OS installed on an eMMC module. The results from this test are shown in Figure 3, which shows minimal difference between the performance of the different storage mediums. This is despite the eMMC being approximately four times faster in terms of both read and write speed [36]. The speed difference between the eMMC and micro-SD for I/O intensive operations has been verified on the hardware used for the HPL tests. These results confirm that the HPL benchmark not I/O bound on the tested Odroid. As shown in Table 3, the eMMC adds to the price of the Odroid cluster, and in these tests it does not show a significant performance increase. The cluster used for the performance benchmarks is equipped with eMMC as further benchmarks in which storage speed will be a more significant factor are planned.

4.2. Results

The benchmarks presented in the following sections are designed to allow direct comparisons between the different SBCs under test. The only parameter changed between platforms is the problem size. The problem size is determined by the amount of RAM available on each node. The P and Q values are changed as the cluster size increases to provide the required distribution of the problem between nodes. Using a standard set of HPL parameters for all platforms ensures repeatability and that any future SBC clusters can be accurately compared against this dataset.

4.2.1. Single board

The performance of standalone SBC nodes is shown in Figure 4. As expected, given its lower CPU speed, as shown in Table 3, the Raspberry Pi 3 Model B has the lowest performance. This is consistent across all problem sizes. The comparison between the Raspberry Pi 3 Model B+ and the Odroid C2 is closer. This is also expected, as they are much closer in terms of specified CPU speed. The comparison is complicated by the fact that the Odroid has twice the RAM and therefore requires larger problem sizes to use the same percentage of memory. The raw processing power the Odroid C2 and the Raspberry Pi 3 Model B+ are comparable. This similarity is not reflected once multiple nodes are combined into a cluster. Both the Raspberry Pi versions tested show a slight performance drop when moving from 70 % to 80 % memory

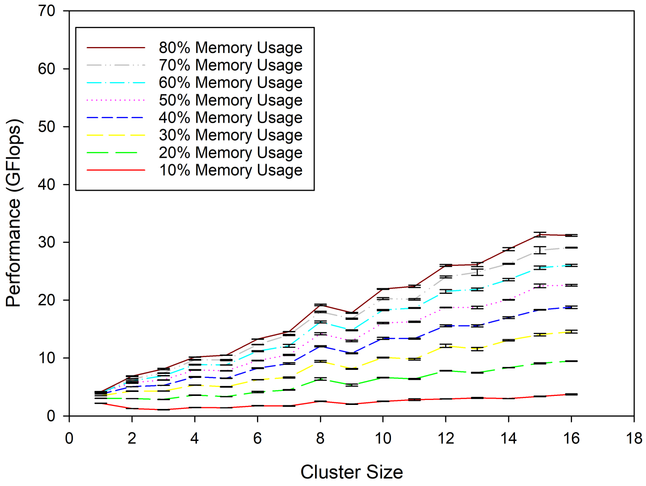


Figure 5: Summary of all cluster performance tests performed on Raspberry Pi 3 Model B. Shows the performance achieved by the cluster for different cluster sizes and memory utilisations. Higher values are better. Error bars are one standard deviation.

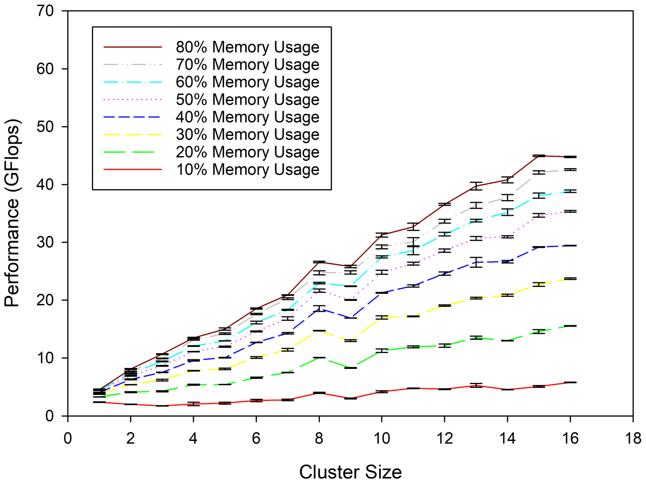


Figure 6: Summary of all cluster performance tests performed on Raspberry Pi 3 Model B+. Shows the performance achieved by the cluster for different cluster sizes and memory utilisations. Higher values are better. Error bars are one standard deviation.

usage. The Odroid does not exhibit this behaviour. Virtual memory (Swap space) was explicitly disabled on all machines in these tests to make sure that it could not be activated at high memory usage, so this can be ruled out as a potential cause. Possible causes of this slow down include: fragmentation of the RAM due to high usage decrease read/write speeds, and background OS tasks being triggered due to the high RAM usage using CPU cycles that would otherwise be available for the HPL processes. The Odroid C2 and the Raspberry Pi SBCs have different amounts of RAM, this means that at 80 % memory usage the Raspberry Pis have 204 MB available for the OS and the Odroid C2 has 409 MB available meaning that the low memory action threshold may not have been reached for the Odroid.

4.2.2. Cluster

Performance benchmark results for the three board types are shown in Figures 5, 6, and 7, respectively. All figures use the same vertical scaling. Each board achieved best performance at 80 % memory usage; Figure 8 compares the clusters at this memory usage point. The fact that maximum performance was achieved at 80 % memory usage is consistent with Equation 1. The Odroid C2 and Raspberry Pi 3 Model B+ have very similar performance in clusters of two nodes; beyond that point the Odroid C2 scales significantly better. Figure 9 illustrates relative cluster performance as cluster size increases.

5. Power Usage

Figure 10 shows the power usage of a single Raspberry Pi 3 Model B+ performing a HPL compute task with problem size 80% memory. The highest power consumption was observed during the main computation period which is shown between vertical red lines. The graph starts with the system idling at approximately 2.5 W. The rise in power to 4.6 W is when the problem is being prepared. The power draw rises to 7 W during the main calculation. After the main calculation, power usage drops to 4.6 W for the verification stage before finally returning to the idle power draw. It is only the power usage during the main compute task that is taken into account when calculating average power usage. The interval between the red lines matches the time given in the HPL output.

Power usage data was recorded at 1 kHz during each of three consecutive runs of the HPL benchmark at 80 % memory usage. The data was then re-sampled to 2 Hz to reduce high frequency noise in the measurements which would interfere with automated threshold detection used to identify start and end times of the test. The start and end of the main HPL run were then identified by using a threshold to detect when the power demand increases to the highest step. To verify that the detection is correct, the calculated runtime was compared to the actual run time. In all cases the calculated runtime was within a second of the runtime reported by HPL. The power demand for this period was then averaged. The readings from the three separate HPL runs were then averaged to get the final value. In all of these measurements, the power requirements of the switch have been excluded. Also discounted is the energy required by the fans which kept the cluster cool, as this is partially determined by ambient environmental conditions. For example, cooling requirements will be lower in a climate controlled data-centre compared to a normal office.

To measure single board power usage and performance,

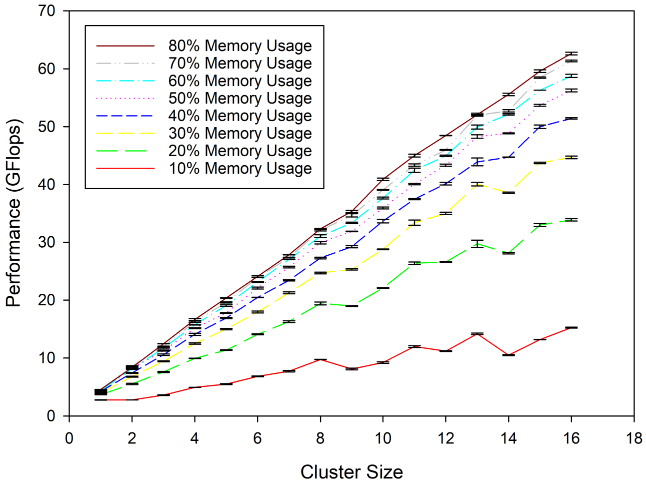


Figure 7: Summary of all cluster performance tests performed on Odroid C2. Shows the performance achieved by the cluster for different cluster sizes and memory utilisations.Higher values are better. Error bars are one standard deviation.

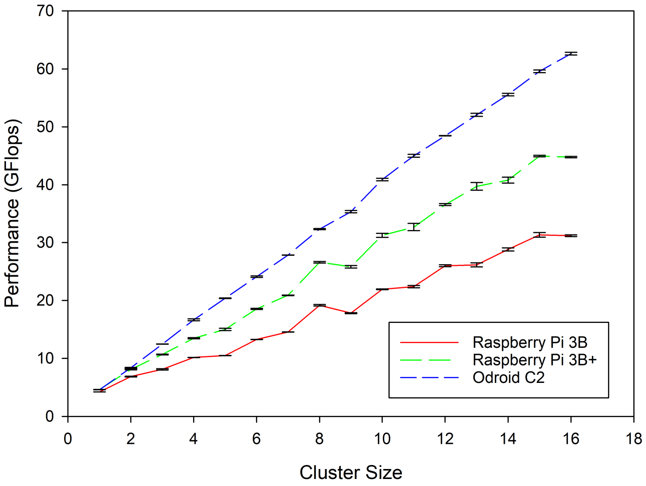


Figure 8: Comparison between the maximum performance of 16-node Raspberry Pi 3 Model B, Raspberry Pi 3 Model B+, and Odroid C2 clusters. All measurements at 80 % memory usage. Higher values are better. Error bars are one standard deviation.

a Power-Z KM001 [37] USB power logger sampled the current and voltage provided by a PowerPax 5 V 3 A USB PSU at a 1 kHz sample rate. A standalone USB PSU was used for this test because this is representative of the way a single SBC might be used by an end user. This technique was not suitable for measuring the power usage of a complete cluster. Instead, cluster power usage was measured using an Instrustar ISDS205C [38]. Channel 1 was directly connected to the voltage supply into the cluster. The current consumed by the cluster was non-intrusively monitored using a Fluke i30s [39] current clamp. The average power used and 0.5 s peaks are shown in Table 4, and the GFLOPS/W power efficiency comparisons are presented in Figure 11. These show that the Raspberry Pi 3 Model B+ is the most power hungry of the clusters tested, with the Odroid C2 using the least power. The Raspberry Pi 3 Model B had the highest peaks, at 23 % above the average power usage. In terms of GFLOPS/W, the Odroid C2 gives the best performance, both individually and when clustered. The Raspberry Pi 3 Model B is more power efficient when running on its own. When clustered, the difference between the Raspberry Pi 3 Model B and 3 Model B+ is minimal, at 0.0117 GFLOPS/W.

6. Discussion

To achieve three results for each memory usage point a total of 384 successful runs was needed from each cluster. An additional three runs of 80 % memory usage of 16 nodes and one node were run for each cluster to obtain measurements for power usage. When running the HPL benchmark, a check is performed automatically to check that the residuals after the computation are acceptable. Both the Raspberry Pi 3 Model B+ and Odroid C2 achieved 0 % failure rates, the Raspberry Pi 3 Model B, produced invalid results 2.5 % of the time. This is after having taken steps to minimise memory corruption on the Raspberry Pi 3 Model B as discussed in Section 4.1.

Table 4: Power usage of the Single Board Computer (SBC) clusters.

SBC	Average Power (W)		0.5 s Peak Power (W)	
	Single	Cluster	Single	Cluster
Raspberry Pi 3 Model B	5.66	103.4	6.08	127.2
Raspberry Pi 3 Model B+	6.95	142.7	7.61	168
Odroid C2	5.07	82.3	5.15	90

When comparing the maximum performance achieved by the different cluster sizes shown in Figure 8, the Odroid C2 scales linearly. However, both Raspberry Pi clusters have significant drops in performance at the same points, for example at nine and 11 nodes. A less significant drop can also be observed at 16 nodes. The cause of these drops in performance which are only observed on the Raspberry Pi SBCs clusters is not known.

6.1. Scaling

Figure 9 shows that as the size of the cluster increases, the achieved percentage of maximum theoretical performance decreases. Maximum theoretical performance is defined as the processing performance of a single node multiplied by the number of nodes in the cluster. By this definition, a single node reaches 100% of the maximum theoretical performance. As is expected the percentage

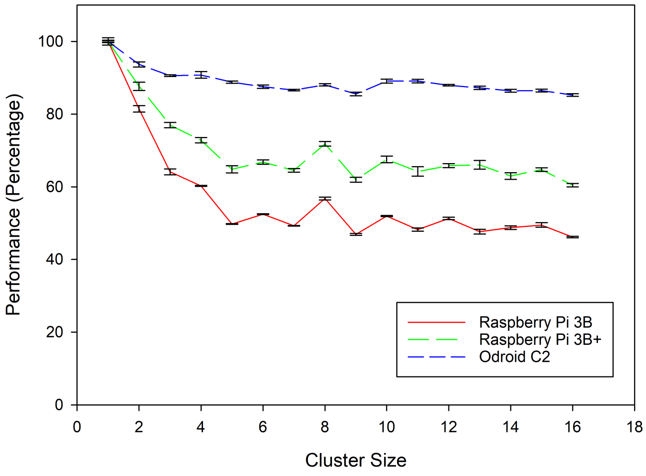


Figure 9: Percentage of maximum performance achieved by each of the Raspberry Pi 3 Model B, Raspberry Pi 3 Model B+ and Odroid C2 clusters. Maximum performance is defined as the performance of a single node multiplied by the number of nodes. All measurements at 80 % memory usage. Higher values are better. Error bars are one standard deviation.

of theoretical performance is inversely proportional to the cluster size. The differences between the scaling of the different SBCs used can be attributed to the differences in the network interface architecture shown in Table 3. The SBC boards with 1 Gbit/s network cards scale significantly better than the Raspberry Pi 3 Model B with the 100 Mbit/s, implying that the limiting factor is network performance. While this work focuses on clusters of 16 nodes (64 cores), it is expected that this network limitation will become more apparent if the node count is increased. The scaling performance of the clusters has significant influence on the other metrics used to compare the clusters because it affects the overall efficiency of the clusters.

6.2. Power efficiency

As shown in Figure 11, when used individually, the Odroid C2 is the most power efficient of the three SBCs considered in this research. This is due to it having not only the highest performance of the three nodes, but also the lowest power usage, both in terms of average power usage during a HPL run and at 0.5 s peaks. Its power efficiency becomes even more apparent in a cluster, where it is 2.4 times more efficient than the Raspberry Pi 3 Model B+, compared to 1.38 times for single nodes. This can be explained by the more efficient scaling exhibited by the Odroid C2. The values for power efficiency could be improved for all SBC considered by increasing the efficiency of the PSU on the Pi Stack PCB.

When performing all the benchmark tests in an office without air-conditioning, a series of 60, 80 and 120 mm fans were arranged to provide cooling for the SBC nodes.

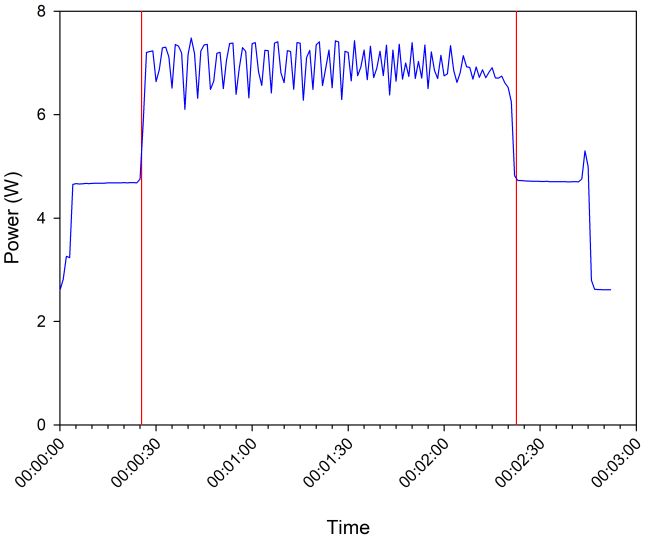


Figure 10: Power usage of a single run of problem size 80% memory on a Raspberry Pi 3 Model B+. Vertical red lines show the period during which performance was measured.

The power consumption for these is not included because of the wide variety of different cooling solutions that could be used. Of the three different nodes, the Raspberry Pi 3 Model B was the most sensitive to cooling, with individual nodes in the cluster reaching their thermal throttle threshold if not receiving adequate airflow. The increased sensitivity to fan position is despite the Raspberry Pi 3 Model B cluster using less power than the Raspberry Pi 3 Model B+ cluster. The redesign of the PCB layout and the addition of the metal heat spreader as part of the upgrade to the Raspberry Pi 3 Model B+ has made a substantial difference [9]. Thermal images of the Pi SBC under high load in [40] show that the heat produced by the Raspberry Pi 3 Model B+ is better distributed over the entire PCB. In the Raspberry Pi 3 Model B the heat is concentrated around the CPU with the actual microchip reaching a higher spot temperature.

6.3. Value for money

When purchasing a cluster a key consideration is how much compute performance is available for a given financial outlay. When comparing cost performance, it is important to include the same components for each item. As it has been shown in Figure 3 that the compute performance of the Odroid C2 is unaffected by whether an SD card or eMMC module is used for the storage, the cost of storage will be ignored in the following calculations. Other costs that have been ignored are the costs of the network switch and associated cabling, and the cost of the Pi Stack PCB. Whilst a 100 Mbit/s switch can be used for the Raspberry Pi 3 Model B as opposed to the 1 Gbit/s switch used for the Raspberry Pi 3 Model B+ and Odroid C2, the dif-

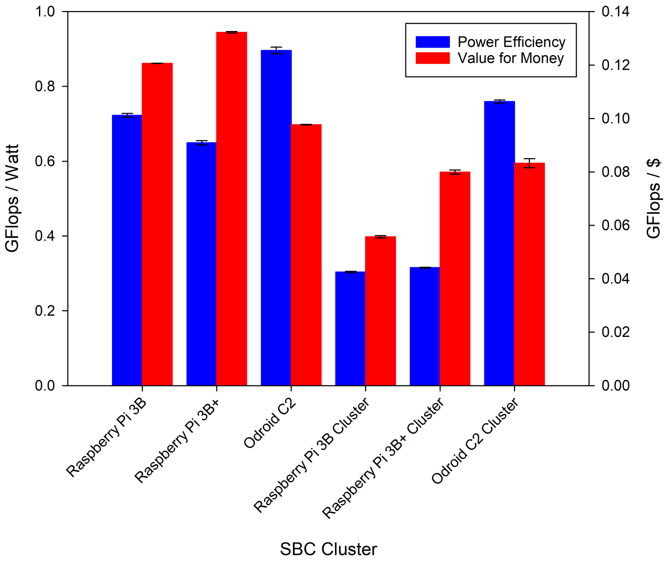


Figure 11: Power efficiency and value for money for each SBC cluster. Value for money calculated using Single Board Computer (SBC) price only. In all cases a higher value is better.

ference in price of such switches is insignificant when compared to the cost of the SBC modules. As such, the figures presented for GFLOPS/$ only include the purchase cost of the SBC modules, running costs are also explicitly excluded. The effect of the scaling performance of the nodes has the biggest influence on this metric. The Raspberry Pi 3 Model B+ performs best at 0.132 GFLOPS/$ when used individually, but it is the Odroid C2 that performs best when clustered, with a value of 0.0833 GFLOPS/$. The better scaling offsets the higher cost per unit. The lower initial purchase price of the Raspberry Pi 3 Model B+ means that a value of 0.0800 GFLOPS/$ is achieved.

6.4. Comparison to previous clusters

Previous papers publishing benchmarks of SBC clusters include an investigation of the performance of the Raspberry Pi 1 Model B [2]. The boards used were the very early Raspberry Pi Boards which only had 256 MB of RAM. Using 64 Raspberry Pi nodes, they achieved a performance of 1.14 GFLOPS. This is significantly less than even a single Raspberry Pi 3 Model B. When performing this test, a problem size of 10 240 was specified which used 22 % of the available RAM of the cluster. This left 199 MB available for the OS to use. The increase in the amount of RAM included in all Raspberry Pi SBCs since the Raspberry Pi 2 means that 205 MB of RAM are available when using 80 % of the available RAM on Raspberry Pi 3 Models B or B+. It may have been possible to increase the problem size tested on Iridis-Pi further, the RAM required in absolute terms for OS tasks was identified as a limiting factor. The clusters tested as part of this paper are formed of 16 nodes, each with four cores. The core counts of the Iridis-Pi cluster and the clusters benchmarked here are the same, enabling comparisons between the two.

When comparing how the performance has increased with node count compared to an optimal linear increase, the Raspberry Pi Model B achieved 84 % performance (64 cores compared to linear extrapolation of four cores). The Raspberry Pi 3 Model B achieved 46 % and the Raspberry Pi 3 Model B+ achieved 60 %. The only SBC platform in this study to achieve comparable scaling is the Odroid C2 at 85 %. This shows that while the Raspberry Pi 3 Model B and 3 Model B+ are limited by the network, with the Raspberry Pi 1 Model B this was not the case. This can be due to the fact that the original Model B has such limited RAM resources that the problem size was too small to highlight the network limitations. An alternative explanation is that the relative comparison between CPU and network performance of the Model B is such that computation power was limiting the benchmark before the lack of network resources became apparent.

In 2016, Mappuji et al. performed a study of a cluster of eight Raspberry Pi 2 nodes [41]. Their chosen problem size of 5040 only made use of 15 % of available RAM, and this limitation is the reason for the poor results and conclusions. They state that there is “no significant difference between using single–core and quad–cores”. This matches the authors experience that when the problem size is too small, increasing the number of nodes in the cluster leads to a decrease in overall performance. This is because the overhead introduced by distributing the problem to the required nodes becomes significant when dealing with large clusters and small problem sizes.

Cloutier et al. [42], used HPL to measure the performance of various SBC nodes. They showed that it was possible to achieve 6.4 GFLOPS by using overclocking options and extra cooling in the form of a large heatsink and forced air. For a non-overclocked node they report a performance of 3.7 GFLOPS for a problem size of 10,000, compared to the problem size 9216 used to achieve the highest performance reading of 4.16 GFLOPS in these tests. This shows that the tuning parameters used for the HPL run are very important. It is for this reason that, apart from the problem size, all other variables have been kept consistent between clusters. Clouteir at al. used the Raspberry Pi 2 Model B because of stability problems with the Raspberry Pi 3 Model B. A slightly different architecture was also used: 24 compute nodes and a single head node, compared to using a compute node as the head node. They achieved a peak performance of 15.5 GFLOPS with the 24 nodes (96 cores), which is 44 % of the maximum linear scaling from four cores. This shows that the network has started to become a limiting factor as the scaling efficiency has dropped. The Raspberry Pi 2 Model B is the first of the quad-core devices released by the Raspberry Pi foundation, and so each node places greater demands on the networking subsystem. This quarters the amount of network bandwidth available to each core when compared to the Raspberry Pi Model B.

The cluster used by Cloutier et al. [42] was instrumented for power consumption. They report a figure of 0.166 GFLOPS/W. This low value is due to the poor initial performance of the Raspberry Pi 2 Model B at 0.432 GFLOPS/W, and the poor scaling achieved. This is not directly comparable to the values in Figure 11 because of the higher node count. The performance of 15.5 GFLOPS achieved by their 24 node Raspberry Pi 2 Model B cluster can be equalled by either eight Raspberry Pi 3 Model B, six Raspberry Pi 3 Model B+, or four Odroid C2. This highlights the large increase in performance that has been achieved in a short time frame. In 2014, a comparison of the available SBC and desktop grade CPU was performed [11]. A 12-core Intel Sandybridge-EP CPU was able to achieve 0.346 GFLOPS/W and 0.021 GFLOPS/$. The Raspberry Pi 3 Model B+ betters both of these metrics, and the Odroid C2 cluster provides more than double the GFLOPS/W. A different SBC, the Odroid XU, which was discontinued in 2014, achieves 2.739 GFLOPS/W for a single board and 1.35 GFLOPS/W for a 10 board cluster [43] therefore performing significantly better than the Sandybridge CPU. These figures need to be treated with caution because the network switching and cooling are explicitly excluded from the calculations for the SBC cluster, and it is not known how the figure for the Sandybridge-EP was obtained, in particular whether peripheral hardware was or was not included in the calculation.

The Odroid XU board, was much more expensive at $169.00 in 2014 and was also equipped with a 64 GB eMMC and a 500 GB Solid-State Drive (SSD). It is not known if the eMMC or SSD affected the benchmark result, but they would have added considerably to the cost. Given the comparison between micro SD and eMMC presented in Figure 3, it is likely that this high performance storage did not affect the results, and is therefore ignored from the following cost calculations. The Odroid XU cluster achieves 0.0281 GFLOPS/$, which is significantly worse than even the Raspberry Pi 3 Model B, the worst value cluster in this paper. This poor value-for-money figure is despite the Odroid XU cluster reaching 83.3 % of linear scaling, performance comparable to the Odroid C2 cluster. The comparison between the Odroid XU and the clusters evaluated in this paper highlights the importance of choosing the SBC carefully when creating a cluster.

Priyambodo et al. created a 33-node Raspberry Pi 2 Model B cluster called “Wisanggeni01” [44], with a single head node and 32 worker nodes. The peak performance of 6.02GFLOPS was achieved at 88.7 % memory usage, far higher than the recommended values. The results presented for a single Raspberry Pi 2 Model B are less than a fifth of the performance achieved by Cloutier et al. [42] for the same model. This performance deficit may be due to the P and Q values chosen for the HPL runs, as the values of P and Q when multiplied give the number of nodes, and not the number of cores. The numbers chosen do not adhere to the ratios set out in the HPL FAQ [29]. The extremely low performance figures that were achieved also partially explain the poor energy efficiency reported, 0.054 GFLOPS/W.

By comparing the performance of the clusters created and studied in this paper with previous SBC clusters, it has been shown that the latest generation of Raspberry Pi SBC are a significant increase in performance when compared to previous generations. These results have also shown that, while popular, with over 19 million devices sold as of March 2018 [9], Raspberry Pi SBCs are not the best choice for creating clusters mainly because of the limitations of the network interface.

6.5. TOP500 Comparison

By using the same benchmarking suite as the TOP500 list of supercomputers, it is possible to compare the performance of these SBC clusters to high performance machines. At the time of writing, the top computer is Summit, with 2,397,824 cores producing 143.5 PFLOPS and consuming 9.783 MW [45]. The SBC clusters do not and cannot compare to the latest super computers; comparison to historical super computers puts the developments into perspective. The first Top500 list was published in June 1993 [46], and the top computer was located at the Los Alamos National Laboratory and achieved a performance of 59.7 GFLOPS. This means that the 16 node Odroid C2 cluster outperforms the winner of the first TOP500 list in June 1993. The Raspberry Pi 3 Model B+ cluster would have ranked 3rd, and the Model B cluster would have been 4th. The Odroid C2 cluster would have remained in the top 10 of this list until June 1996 [47], and would have remained in the TOP500 list until November 2000 [48], at which point it would have ranked 411th. As well as the main TOP500 list there is also the Green500 list which is the TOP500 supercomputers in terms of power efficiency. Shoubu system B [49] currently tops this list, with 953280 cores producing 1063 TFLOPS, consuming 60 kW of power to give a power efficiency of 17.6 GFLOPS/W. This power efficiency is an order of magnitude better than is currently achieved by the SBC clusters studied here. The Green500 list also specifies that the power used by the interconnect is included in the measurements [50], something that has been deliberately excluded from the SBC benchmarks performed.

6.6. Uses of SBC clusters

The different use cases for SBC clusters are presented in Section 2. The increases in processing power observed throughout the benchmarks presented in this paper can be discussed in the context of these example uses.

When used in an education setting to give students the experience of working with a cluster the processing power of the cluster is not an important consideration as it is the exposure to the cluster management tools and techniques that is the key learning outcome. The situation is very different when the cluster is used for research and developed such as the 750 node Los Alamos Raspberry Pi cluster [12].

The performance of this computer is unknown, and it is predicted that the network performance is a major limiting factor affecting how the cluster is used. The performance of these research clusters determines how long a test takes to run, and therefore the turnaround time for new developments. In this context the performance of the SBC clusters is revolutionary. A cluster of 16 Odroid C2 gives performance that, less than 20 years ago, would have been world leading. Greater performance can now be achieved by a single powerful computer such as a mac pro (3.2 GHz dual quad-core CPU) which can achieve 91 GFLOPS [51], but this is not an appropriate substitute because developing algorithms to run efficiently in a highly distributed manner is a complex process. The value of the SBC cluster is that it uses multiple distributed processors, emulating the architecture of a traditional supercomputer, exposing issues that may be hidden by the single OS on a single desktop PC.

All performance increases observed in SBC boards leads to a direct increase in the available compute power available at the edge. These performance increases enable new more complex algorithms to be push out to the edge compute infrastructure. When considering expendable compute use cases the value for money GFLOPS/$, is an important consideration. The Raspberry Pi has not increased in price since the initial launch in 2012 but the Raspberry Pi 3 Model B+ has more than 100 times the processing power of the Raspberry Pi 1 Model B. The increase in power efficiency GFLOPS/W means that more computation can be performed for a given amount of energy therefore enabling more complex calculations to be performed in resource constrained environment.

The increase in value for money and energy efficiency also has a direct impact for the next generation data centres. Any small increase in these metrics for a single board is amplified many times when scaled up to data centre volumes.

Portable clusters are a type of edge compute cluster and are affected in the same way as these clusters. In portable clusters size is a key consideration. In 2013 64 Raspberry Pi 1 Model B and their associated infrastructure were needed to achieve 1.14 GFLOPS of processing power, less than quarter available on a single Raspberry Pi 3 Model B+.

As shown the increase in performance achieved by SBCs in general and specifically SBC clusters have applications across all the use cases discussed in Section 2.

7. Future Work

Potential areas of future work from this research can be split into three categories: further development of the Pi Stack PCB clustering technique, further benchmarking of SBC clusters, and applications of SBC clusters, for example in edge compute applications.

The Pi Stack has enabled the creation of multiple SBC clusters, the production of these clusters and evaluation of the Pi Stack board have identified two areas for improvement. Testing of 16 node clusters has shown that the M2.5 stand-offs required by the physical design of the Pi do not have sufficient current capability even when distributing power at 24 V. This is an area which would benefit from further investigation to see if the number of power insertion points can be reduced. The Pi Stack achieved efficiency in the region of 70 %, which is suitable for use in non-energy constrained situations better efficiency translates directly into more power available for computation when in energy limited environments, for the same reason investigation into reducing the idle current of the Pi Stack would be beneficial.

This paper has presented a comparison between 3 different SBC platforms in terms of HPL benchmarking. This benchmark technique primarily focuses on CPU performance however, the results show that the network architecture of SBCs can have a significant influence on the results. Further investigation is needed to benchmark SBC clusters with other workloads, for example Hadoop as tested on earlier clusters [2, 52, 53].

Having developed a new cluster building technique using the Pi Stack these clusters should move beyond bench testing of the equipment, and be deployed into real-world edge computing scenarios. This will enable the design of the Pi Stack to be further evaluated and improved and realise the potential that these SBC clusters have been shown to have.

8. Conclusion

Previous papers have created different ways of building SBC clusters. The power control and status monitoring of these clusters has not been addressed in previous works. This paper presents the Pi Stack, a new product for creating clusters of SBCs which use the Raspberry Pi HAT pin out and physical layout. The Pi Stack minimises the amount of cabling required to create a cluster by reusing the metal stand-offs used for physical construction for both power and management communications. The Pi Stack has been shown to be a reliable cluster construction technique, which has been designed to create clusters suitable for use in either edge or portable compute clusters, use cases for which none of the existing cluster creation techniques are suitable.

Three separate clusters each of 16 nodes have been created using the Pi Stack. These clusters are created from Raspberry Pi 3 Model B, Raspberry Pi 3 Model B+ and Odroid C2 SBCs. The three node types were chosen to compare the latest versions of the Raspberry Pi to the original as benchmarked by Cox et al. [2]. The Odroid C2 was chosen as an alternative to the Raspberry Pi boards to see if the architecture decisions made by the Raspberry Pi foundation limited performance when creating clusters. The physical constraints of the Pi Stack PCB restricted the boards that were suitable for this test. The Odroid C2 can use either a micro-SD or eMMC for storage, and tests have shown that this choice does not affect the results of a comparison using the HPL benchmark as does not test the I/O bandwidth. When comparing single node performance, the Odroid C2 and the Raspberry Pi 3 Model B+ are comparable at large problem sizes of 80 % memory usage. Both node types outperform the Raspberry Pi 3 Model B. The Odroid C2 performs the best as cluster size increases: a 16 node Odroid C2 cluster provides 40 % more performance in GFLOPS than an equivalently-sized cluster of Raspberry Pi 3 Model B+ nodes. This is due to the gigabit network performance of the Odroid C2; Raspberry Pi 3 Model B+ network performance is limited by the 480 Mbit/s USB2 connection to the CPU. This better scaling performance from the Odroid C2 also means that the Odroid cluster achieves better power efficiency and value for money.

When comparing these results to benchmarks of previous generations of Raspberry Pi SBCs, it becomes clear how much the performance of these SBCs has improved. A 16-node (64-core) cluster of Raspberry Pi Model B achieved 1.14 GFLOPS, about a quarter of the performance of a single Raspberry Pi 3 Model B+. The results presented in this paper show that the performance and construction of SBC clusters has moved from interesting idea to being able to provide meaningful amounts of processing power for use in the infrastructure of future IoT deployments. The results presented in this paper are available from: `doi.org/10.5281/zenodo.2002730`.

References

[1] R. Cellan-Jones, The raspberry pi computer goes on general sale, `https://www.bbc.co.uk/news/technology-17190918`, Accessed: 2019-05-03 (2012).

[2] S. J. Cox, J. T. Cox, R. P. Boardman, S. J. Johnston, M. Scott, N. S. O'Brien, Iridis-pi: a low-cost, compact demonstration cluster , Cluster Computing (2013). `doi:10.1007/s10586-013-0282-7`.

[3] S. J. Johnston, P. J. Basford, C. S. Perkins, H. Herry, F. P. Tso, D. Pezaros, R. D. Mullins, E. Yoneki, S. J. Cox, J. Singer, Commodity single board computer clusters and their applications, Future Generation Computer Systems (2018). `doi:10.1016/j.future.2018.06.048`.

[4] Raspberry Pi Foundation, Raspberry Pi 3 Model B, `https://www.raspberrypi.org/products/raspberry-pi-3-model-b/`, Accessed: 2019-05-03 (2016).

[5] Raspberry Pi Foundation, Raspberry Pi 3 Model B+, `https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/`, Accessed: 2019-05-03 (2018).

[6] Hard Kernel, Odroid C2, `https://www.hardkernel.com/shop/odroid-c2/`, Accessed: 2019-05-03 (2018).

[7] G. Gibb, Linpack and BLAS on Wee Archie, `https://www.epcc.ed.ac.uk/blog/2017/06/07/linpack-and-blas-wee-archie`, Accessed: 2019-05-03 (2017).

[8] D. Papakyriakou, D. Kottou, I. Kostouros, Benchmarking Raspberry Pi 2 Beowulf Cluster, International Journal of Computer Applications 179 (32) (2018) 21–27 (April 2018). `doi:10.5120/ijca2018916728`.

[9] E. Upton, Raspberry Pi 3 Model B+ on sale now at $35, `https://www.raspberrypi.org/blog/raspberry-pi-3-model-bplus-sale-now-35/`, Accessed: 2019-05-03 (2018).

[10] F. P. Tso, D. R. White, S. Jouet, J. Singer, D. P. Pezaros, The Glasgow Raspberry Pi Cloud: a scale model for cloud computing infrastructures, in: 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops, IEEE, 2013, pp. 108–112 (July 2013). `doi:10.1109/ICDCSW.2013.25`.

[11] M. F. Cloutier, C. Paradis, V. M. Weaver, Design and Analysis of a 32-bit Embedded High-Performance Cluster Optimized for Energy and Performance, in: 2014 Hardware-Software Co-Design for High Performance Computing, IEEE, 2014, pp. 1–8 (November 2014). `doi:10.1109/Co-HPC.2014.7`.

[12] L. Tung, Raspberry Pi supercomputer: Los alamos to use 10,000 tiny boards to test software, `https://www.zdnet.com/article/raspberry-pi-supercomputer-los-alamos-to-use-10000-tiny-boards-to-test-software/`, Accessed: 2019-05-03 (2017).

[13] M. Keller, J. Beutel, L. Thiele, Demo Abstract: Mountainview – Precision image sensing on high-alpine locations, in: D. Pesch, S. Das (Eds.), Adjunct Proceedings of the 6th European Workshop on Sensor Networks (EWSN), Cork, 2009, pp. 15–16 (February 2009).

[14] K. Martinez, P. J. Basford, J. Ellul, R. Spanton, Gumsense - a high power low power sensor node, in: D. Pesch, S. Das (Eds.), Adjunct Proceedings of the 6th European Workshop on Sensor Networks (EWSN), Cork, 2009, pp. 27–28 (February 2009).

[15] C. Pahl, S. Helmer, L. Miori, J. Sanin, B. Lee, A container-based edge cloud PaaS architecture based on Raspberry Pi clusters, in: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), IEEE, Vienna, 2016, pp. 117–124 (August 2016). `doi:10.1109/W-FiCloud.2016.36`.

[16] P. Stevens, Raspberry Pi cloud, `https://blog.mythic-beasts.com/wp-content/uploads/2017/03/raspberry-pi-cloud-final.pdf`, Accessed: 2019-05-03 (2017).

[17] A. Davis, The Evolution of the Beast Continues, `https://www.balena.io/blog/the-evolution-of-the-beast-continues/`, Accessed: 2019-05-03 (2017).

[18] J. Adams, Raspberry Pi Model B+ mechanical drawing, `https://www.raspberrypi.org/documentation/hardware/raspberrypi/mechanical/rpi_MECH_bplus_1p2.pdf`, Accessed: 2019-05-03 (2014).

[19] Add-on boards and HATs, `https://github.com/raspberrypi/hats/releases/tag/1.0`, Accessed: 2019-05-03 (2014).

[20] E. Upton, G. Halfacree, Raspberry Pi User Guide, 4th Edition, Wiley, 2016 (2016).

[21] P. Basford, S. Johnston, Pi Stack PCB (2017). `doi:10.5258/SOTON/D0379`.

[22] H. Marias, RS-485/RS-422 Circuit Implementation Guide Application Note, Tech. rep., Analog Devices, Inc. (2008).

[23] W. Peterson, D. Brown, Cyclic Codes for Error Detection, Proceedings of the IRE 49 (1) (1961) 228–235 (January 1961). `doi:10.1109/JRPROC.1961.287814`.

[24] A. Petitet, R. Whaley, J. Dongarra, A. Cleary, HPL – a portable implementation of the high–performance Linpack benchmark for distributed–memory computers, `www.netlib.org/benchmark/hpl`, Accessed: 2019-05-03 (2004).

[25] H. W. Meuer, The TOP500 project: Looking back over 15 years of supercomputing experience, Informatik-Spektrum 31 (3) (2008) 203–222 (2008). `doi:10.1007/s00287-008-0240-6`.

[26] J. J. Dongarra, Performance of various computers using standard linear equations software, `http://icl.cs.utk.edu/projectsfiles/rib/pubs/performance.pdf`, Accessed: 2019-05-03 (2007).

[27] R. Whaley, J. Dongarra, Automatically Tuned Linear Algebra Software, in: Proceedings of the IEEE/ACM SC98 Conference, 1998, pp. 1–27 (1998). `doi:10.1109/SC.1998.10004`.

[28] P. Bridges, N. Doss, W. Gropp, E. Karrels, E. Lusk, A. Skjellum, User's Guide to MPICH, a Portable Implementation of MPI, Argonne National Laboratory (1995).

[29] HPL frequently asked questions, `http://www.netlib.org/benchmark/hpl/faqs.html`, accessed: 2019-05-03.

[30] M. Sindi, How to - High Performance Linpack HPL, `http://www.crc.nd.edu/~rich/CRC_Summer_Scholars_2014/HPL-HowTo.pdf`, Accessed: 2019-05-03 (2009).

[31] T. Leng, R. Ali, J. Hsieh, V. Mashayekhi, R. Rooholamini, Performance impact of process mapping on small-scale SMP clusters - a case study using high performance linpack, in: Parallel and Distributed Processing Symposium, International, Vol. 2, Fort Lauderdale, Florida, US, 2002, p. 8 (2002). doi:10.1109/IPDPS.2002.1016657.

[32] Compaq, Hewlet-Packard, Intel, Lucent, Microsoft, NEC, Philips, Universal serial bus specification: Revision 2.0, http://sdphca.ucsd.edu/lab_equip_manuals/usb_20.pdf, Accessed: 2019-05-03 (2000).

[33] UP specification, https://up-shop.org/index.php?controller=attachment&id_attachment=146, Accessed: 2019-05-03 (2016).

[34] P. Vouzis, iPerf Comparison: Raspberry Pi3 B+, NanoPi, Up-Board & Odroid C2, https://netbeez.net/blog/iperf-comparison-between-raspberry-pi3-b-nanopi-up-board-odroid-c2/, Accessed: 2019-05-03 (2018).

[35] Dom, Pi3 incorrect results under load (possibly heat related), https://www.raspberrypi.org/forums/viewtopic.php?f=63&t=139712&start=25#p929783, Accessed: 2019-05-03 (2016).

[36] Hard Kernel, 16GB eMMC module c2 linux, https://www.hardkernel.com/shop/16gb-emmc-module-c2-linux/, Accessed: 2019-05-03 (2018).

[37] Power-Z king meter manual, https://sourceforge.net/projects/power-z-usb-software-download/files/Manual%2BPOWER-Z%20KM001.pdf/download, Accessed: 2019-05-03 (2017).

[38] Instrustar, ISDS205 user guide, http://instrustar.com/upload/user%20guide/ISDS205%20User%20Guide.pdf, Accessed: 2019-05-03 (2016).

[39] Fluke, i30s/i30 AC/DC current clamps instruction sheet, http://assets.fluke.com/manuals/i30s_i30iseng0000.pdf, Accessed: 2019-05-03 (2006).

[40] G. Halfacree, Benchmarking the Raspberry Pi 3 B+, https://medium.com/@ghalfacree/benchmarking-the-raspberry-pi-3-b-plus-44122cf3d806, Accessed: 2019-05-03 (2018).

[41] A. Mappuji, N. Effendy, M. Mustaghfirin, F. Sondok, R. P. Yuniar, S. P. Pangesti, Study of Raspberry Pi 2 quad-core Cortex-A7 CPU cluster as a mini supercomputer, in: 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), IEEE, 2016, pp. 1–4 (October 2016). doi:10.1109/ICITEED.2016.7863250.

[42] M. Cloutier, C. Paradis, V. Weaver, A Raspberry Pi Cluster Instrumented for Fine-Grained Power Measurement, Electronics 5 (4) (2016) 61 (September 2016). doi:10.3390/electronics5040061.

[43] D. Nam, J.-s. Kim, H. Ryu, G. Gu, C. Y. Park, SLAP : Making a Case for the Low-Powered Cluster by leveraging Mobile Processors, in: Super Computing: Student Posters, Austin, Texas, 2015, pp. 4–5 (2015).

[44] T. Priyambodo, A. Lisan, M. Riasetiawan, Inexpensive green mini supercomputer based on single board computer cluster, Journal of Telecommunication, Electronic and Computer Engineering 10 (1-6) (2018) 141–145 (2018).

[45] TOP500 list - November 2018, https://www.top500.org/lists/2018/11/, Accessed: 2019-05-03 (2018).

[46] TOP500 list - June 1993, https://www.top500.org/list/1993/06/, Accessed: 2019-05-03 (1993).

[47] TOP500 list - June 1996, https://www.top500.org/list/1996/06/, Accessed: 2019-05-03 (1996).

[48] TOP500 list - November 2000, https://www.top500.org/list/2000/11/, Accessed: 2019-05-03 (2000).

[49] GREEN500 list - November 2018, https://www.top500.org/green500/list/2018/11/, Accessed: 2019-05-03 (2018).

[50] Energy Efficient High Performance Computing Power Measurement Methodology, Tech. rep., EE HPC Group (2015).

[51] J. Martellaro, The fastest Mac compared to today's supercomputers, https://www.macobserver.com/tmo/article/The_Fastest_Mac_Compared_to_Todays_Supercomputers, Accessed: 2019-05-03 (2008).

[52] C. Kaewkasi, W. Srisuruk, A study of big data processing constraints on a low-power Hadoop cluster, in: 2014 International Computer Science and Engineering Conference (ICSEC), IEEE, 2014, pp. 267–272 (July 2014). doi:10.1109/ICSEC.2014.6978206.

[53] W. Hajji, F. Tso, Understanding the Performance of Low Power Raspberry Pi Cloud for Big Data, Electronics 5 (4) (2016) 29 (June 2016). doi:10.3390/electronics5020029.

Acknowledgement

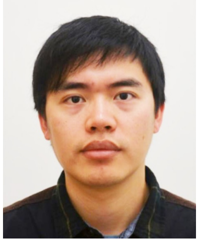
This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) Reference EP/P004024/1. Alexander Tinuic contributed to the design and development of the Pi Stack board.

Philip J Basford is a Senior Research Fellow in Distributed Computing in the Faculty of Engineering & Physical Sciences at the University of Southampton. Prior to this he was an Enterprise Fellow and Research Fellow in Electronics and Computer Science at the same institution. He has previously worked in the areas of commercialising research and environmental sensor networks. Dr Basford received his MEng (2008) and PhD (2015) from the University of Southampton, and is a member of the IET.

Steven J Johnston is a Senior Research Fellow in the Faculty of Engineering & Physical Sciences at the University of Southampton. Steven completed a PhD with the Computational Engineering and Design Group and he also received an MEng degree in Software Engineering from the School of Electronics and Computer Science. Steven has participated in 40+ outreach and public engagement events as an outreach program manager for Microsoft Research. He currently operates the LoRaWAN wireless network for Southampton and his current research includes the large scale deployment of environmental sensors. He is a member of the IET.

Colin Perkins is a Senior Lecturer (Associate Professor) in the School of Computing Science at the University of Glasgow. He works on networked multimedia transport protocols, network measurement, routing, and edge computing, and has published more than 60 papers in these areas. He is also a long time participant in the IETF, where he co-chairs the RTP Media Congestion Control working group. Dr Perkins holds a DPhil in Electronics from the University of York, and is a senior member of the IEEE, and a member of the IET and ACM.

Tony Garnock-Jones is a Research Associate in the School of Computing Science at the University of Glasgow. His interests include personal, distributed computing and the design of programming language features for expressing concurrent, interactive and distributed programs. Dr Garnock-Jones received his BSc from the University of Auckland, and his PhD from Northeastern University.



Fung Po Tso is a Lecturer (Assistant Professor) in the Department of Computer Science at Loughborough University. He was Lecturer in the Department of Computer Science at Liverpool John Moores University during 2014-2017, and was SICSA Next Generation Internet Fellow based at the School of Computing Science, University of Glasgow from 2011-2014. He received BEng, MPhil, and PhD degrees from City University of Hong Kong in 2005, 2007, and 2011 respectively. He is currently researching in the areas of cloud computing, data centre networking, network policy/functions chaining and large scale distributed systems with a focus on big data systems.

Dimitrios Pezaros is a Senior Lecturer (Associate Professor) and director of the Networked Systems Research Laboratory (netlab) at the School of Computing Science, University of Glasgow. He has received funding in excess of £3m for his research, and has published widely in the areas of computer communications, network and service management, and resilience of future networked infrastructures. Dr Pezaros holds BSc (2000) and PhD (2005) degrees in Computer Science from Lancaster University, UK, and has been a doctoral fellow of Agilent Technologies between 2000 and 2004. He is a Chartered Engineer, and a senior member of the IEEE and the ACM.

Robert Mullins is a Senior Lecturer in the Computer Laboratory at the University of Cambridge. He was a founder of the Raspberry Pi Foundation. His current research interests include computer architecture, open-source hardware and accelerators for machine learning. He is a founder and director of the lowRISC project. Dr Mullins received his BEng, MSc and PhD degrees from the University of Edinburgh.

Eiko Yoneki is a Research Fellow in the Systems Research Group of the University of Cambridge Computer Laboratory and a Turing Fellow at the Alan Turing Institute. She received her PhD in Computer Science from the University of Cambridge. Prior to academia, she worked for IBM US, where she received the highest Technical Award. Her research interests span distributed systems, networking and databases, including complex network analysis and parallel computing for large-scale data processing. Her current research focus is auto-tuning to deal with complex parameter spaces using machine-learning approaches.

Jeremy Singer is a Senior Lecturer (Associate Professor) in the School of Computing Science at the University of Glasgow. He works on programming languages and runtime systems, with a particular focus on manycore platforms. He has published more than 30 papers in these areas. Dr Singer has BA, MA and PhD degrees from the University of Cambridge. He is a member of the ACM.

Simon J Cox is Professor of Computational Methods at the University of Southampton. He has a doctorate in Electronics and Computer Science, first class degrees in Maths and Physics and has won over £30m in research & enterprise funding, and industrial sponsorship. He has published over 250 papers. He has co-founded two spin-out companies and, as Associate Dean for Enterprise, has most recently been responsible for a team of 100 staff with a £11m annual turnover providing industrial engineering consultancy, large-scale experimental facilities and healthcare services.