Scale-Down Experiments on TPCx-HS

Maximilian Böther, Tilmann Rabl
Hasso Plattner Institute, University of Potsdam
maximilian.boether@student.hpi.de,tilmann.rabl@hpi.de

ABSTRACT

The Transaction Processing Performance Council's (TPC) benchmarks are the standard for evaluating data processing performance and are extensively used in academia and industry. Official TPC results are usually produced on high-end deployments, making transferability to commodity hardware difficult. Recent performance improvements on low-power ARM CPUs have made low-end computers, such as the Raspberry Pi, a candidate platform for distributed, low-scale data processing.

In this paper, we conduct a feasibility study of executing scaled-down big data workloads on low-power ARM clusters. To this end, we run the TPCx-HS benchmark on two Raspberry Pi clusters. TPCx-HS is the ideal candidate for hardware comparisons and understanding hardware characteristics for data processing workloads because TPCx-HS results do not depend on specific software implementations and the benchmark has limited options for workload-specific tuning. Our evaluation shows that Pis exhibit similar behavior to large-scale big data systems in terms of price performance and relative throughput to performance results. Current generation Pi clusters are becoming a reasonable choice for GB-scale data processing due to the increasing amount of available memory, while older versions struggle with stable execution of high-load scenarios.

CCS CONCEPTS

• **Computer systems organization → Distributed architectures**; **System on a chip**.

KEYWORDS

Benchmark, TPC, TPCx-HS, ARM, Raspberry Pi, Hadoop

ACM Reference Format:
Maximilian Böther, Tilmann Rabl. 2021. Scale-Down Experiments on TPCx-HS. In *Big Data in Emergent Distributed Environments (BiDEDE'21), June 20, 2021, Virtual Event, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3460866.3461774

1 INTRODUCTION

Over the last few years, edge devices and system on a chip (SoC) computers have gained popularity as well as performance. While in the beginning, they were mostly single-core low-memory chips, current generations of SoC computers often feature multi-core CPUs and several gigabytes of memory. One well-known SoC-device is the Raspberry Pi, used in various domains, such as primary school teaching [21], IoT scenarios [15], and academic research [5]. In this paper, we analyze whether modern Raspberry Pi clusters can support data analysis tasks and, to this end, conduct a feasibility study of running scaled-down big data workloads on SoC-clusters. We configure two five-node Raspberry Pi clusters and execute the TPCx-HS industry-standard big data benchmark on them. One cluster consists of Raspberry Pi 3 worker nodes, while the other consists of recent Raspberry Pi 4 nodes.

The TPCx-HS benchmark is used to evaluate big data deployments on performance, price-performance, and availability metrics [18, 24]. Unlike other TPC benchmarks, TPCx-HS does not measure a specific (database) implementation but comes with a complete software kit built on top of Hadoop or Spark, and profits mostly from correct system configuration instead of workload-specific tuning. It is an ideal benchmark to analyze the hardware characteristics of a big data system. Therefore, using TPCx-HS, we gain insights into whether SoC clusters behave differently to enterprise-scale deployments or if they have similar bottlenecks. In this paper, we make the following contributions.

- We run the TPCx-HS benchmark using the MapReduce framework on two generations of Raspberry Pi clusters, including the setup of Hadoop on ARM, for which we fixed the AArch32 compilation[1].
- We characterize the hardware and analyze the results to understand the behavior of the clusters.
- We compare the results to publicly available TPCx-HS results to evaluate the relative performance of our Raspberry Pi clusters.

Both analysis and comparison show that Pi clusters have similar characteristics to large-scale clusters on primary benchmark metrics. The rest of this paper is structured as follows. In Section 2, we briefly introduce the Raspberry Pi, Hadoop, and the TPCx-HS benchmark. Section 3 gives an overview of our cluster setup and configuration. In Section 4, we present our evaluation. The results are compared to large clusters in Section 5. We discuss related work in Section 6, before concluding the paper in Section 7.

2 BACKGROUND

In this section, we give a high-level overview of the hardware infrastructure and data processing frameworks and benchmarks.

Raspberry Pi. The Raspberry Pi (Pi) is an ARM-based system on a chip. It was launched in 2012 and has gone through several iterations since then. The Pi version 1 Model A started with a 700 MHz CPU and 256 MB of memory; the latest release in 2020, the Raspberry Pi 4, features a 1 500 MHz Cortex-A72 CPU and up to 8

BiDEDE'21, June 20, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8465-0/21/06…$15.00
https://doi.org/10.1145/3460866.3461774

[1] We provide an Ansible playbook and a setup guide for a Pi Hadoop cluster on our website: https://hpi.de/rabl/projects/raspberry-pi-cluster.html

GB of RAM. The focus of the system is to provide a fast, but low-power platform for development at a low price. Pis have been used in various application scenarios. They are often used as IoT-devices together with sensors, for home automation, or as a learning tool.

Hadoop. Hadoop [3] is an open-source software stack for big data processing. It comprises the Hadoop distributed file system HDFS [22], the Hadoop MapReduce engine, an implementation of Google's MapReduce data processing framework [7], and other components for job scheduling, data storage, and locking. MapReduce enables simple development of parallel data processing on large clusters of commodity server hardware by reducing the API at its core to two higher-order functions, map and reduce. While targeting large clusters of commodity hardware, Hadoop is not built for low-end platforms and requires special configuration to be run on Pis. A Hadoop cluster consists of multiple services. The NameNode process is responsible for managing the HDFS file system, so the node executing the NameNode process is considered the primary of the cluster. DataNode process nodes are considered the secondaries since this process is only responsible for managing the storage of the respective node and communicates with the NameNode. A primary/secondary architecture is also used for data processing, where the ResourceManager process manages the compute units, and on each compute unit the NodeManager is responsible for local resource management and communication with the ResourceManager. Resources are allocated in YARN containers [2].

TPCx-HS. The TPC Express Benchmark for Hadoop Sort is a big data processing benchmark based on the Hadoop TeraSort program [18]. It primarily measures hardware performance instead of software implementations. Besides verification steps, it consists of three benchmark phases, data generation, sorting, and data validation. In the first phase, 100 byte records are generated randomly and replicated three times on the cluster. In the smallest scale factor relevant for *official* results, one terabyte of data is generated. The data is then sorted and the result is validated.

Each run consists of two sub-runs, where the run with the lower performance metric defines the performance run. The performance metric $HSph@SF$ describes the throughput of the benchmark for the scale factor SF and is defined as $HSph@SF = \frac{SF}{T/3600}$, where T is the total elapsed time for the run in seconds. The higher the performance metric, the better the throughput. The benchmark also mandates the publication of a price-performance metric to evaluate price efficiency. It is defined as $\$/HSph@SF = \frac{P}{HSph@SF}$, where P is the total cost of ownership. An energy-performance metric is defined, which can optionally be reported with an official result.

3 CLUSTER SETUP

In this section, we describe the setup of the clusters, the total cost of ownership, and give details on the Hadoop configuration.

3.1 Overview

Our two clusters consist of five Raspberry Pis each. For each cluster, one Pi serves as the primary node. It is the gateway of the network and runs the NameNode as well as the ResourceManager and the HistoryServer of the Hadoop cluster. The four other Pis serve as worker nodes and run the DataNode as well as the NodeManager processes. All of the Pis are connected through their Ethernet network interface with a Gigabit network switch. We power the clusters using USB power. The primary node is connected over a USB-to-Ethernet adapter to a workstation or notebook which then redirects traffic to the internet. By dedicating the network to the Pis, we ensure that no external connections interfere with our benchmarks.

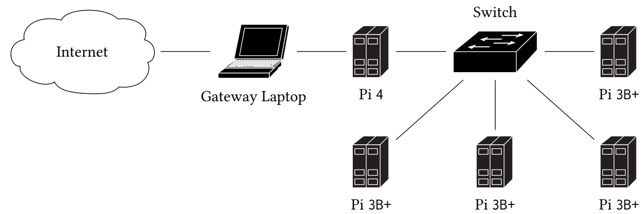


Figure 1: Schematic setup of the Pi 3 cluster.

Table 1: Total cost of ownership list of the Pi 3 cluster.

	Price	Amount	**Total Price**
Raspberry Pi 4	€61.99	x1	€61.99
Raspberry Pi 3B+	€44.10	x4	€176.40
Network Gear	€54.33	x1	€54.33
5 SanDisk & Samsung SD Cards	€42.34	x1	€42.34
Power Distribution	€56.27	x1	€56.27
Cluster Case	€20.00	x1	€20.00

Pi 3 Cluster. This cluster consists of four Raspberry Pi 3B+ with 1 GB of memory and one Raspberry Pi 4 with 4 GB of memory. Each Pi has access to 32 GB of SD card storage. We use Sandisk Ultra SD cards as well as Samsung Evo SD cards. The Raspberry Pi 4 serves as the primary node of the cluster. All of the Pis run Raspbian 10 Lite in 32-bit mode, based on Debian Buster and Linux Kernel `4.19.97-v7+ #1294`. Our Hadoop version is `3.1.3`. Figure 1 shows a schematic diagram of our setup.

Pi 4 Cluster. This cluster consists of five Raspberry Pi 4 with 8 GB of main memory. Each Pi has 128 GB of Samsung Evo+ SD card storage. All of the Pis run the Raspbian 10 64-bit beta using Kernel `5.4.51-v8+ #1327` and Hadoop `3.1.3`.

3.2 Cost of Ownership

One key metric for TPCx-HS results is the price-performance metric. To calculate it, we need to assess the total cost of ownership (TCO) of our cluster. We state the prices we paid for our systems in euros (after VAT). For the Pi 3 cluster, please find the details in Table 1; for the Pi 4 cluster, we refer to Table 2. For the Pi 3 cluster, the total amount sums up to €411.33, which, on March 17, 2021, was equivalent to $493. The Pi 4 cluster costs €737.76 in total, equivalent to $883. These calculations assume no additional maintenance fees.

3.3 Hadoop Specifics

For the Pi 3 cluster, one major issue in executing TPCx-HS is the 1 GB memory limit of a Pi 3B+. We thus need to fine-tune the Yarn and MapReduce settings of the cluster in order to balance stability and performance. For the Pi 4 cluster, there are various possible combinations of block size and container size that we investigate.

Table 2: Total cost of ownership of the Pi 4 cluster.

	Price	Amount	**Total Price**
Raspberry Pi 4	€76.86	x5	€398.40
Heatsinks	€2.00	x5	€10.00
Network Gear	€73.83	x1	€73.83
128 GB Samsung Evo+ SD Card	€22.62	x5	€113.10
Power Distribution	€109.43	x1	€109.43
Cluster Case	€11.00	x3	€33.00

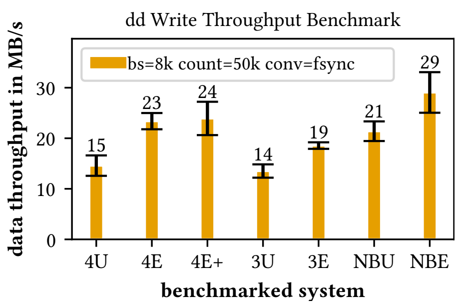


Figure 2: Results of the dd microbenchmarks. In the legend, the 3 and 4 indicate the respective Pi, NB indicates the Notebook, U, E, and E+ indicate the Ultra and Evo(+) MicroSDs. For example, 4E+ refers to the Evo+ card in a Pi 4.

In the following, we focus on the amount of memory assigned to the mappers and reducers, and the HDFS block size.

Pi 3 Cluster. In testing, when allocating 768 MB in total to YARN containers, we observed unpredictable operating system kernel crashes on several Pis. On the other hand, reducing a single mapper/reducer to below 256 MB of RAM impacted stability as well because container creation failed due to too little memory. Thus, we compromised on allocating two mappers/reducers, having 256 MB of memory each available, on each Pi 3B+. Regardless of the data volume in our experiments, no container or node crashes with these settings. These settings result in each Pi 3B+ either running one application primary or two mappers/reducers. We reduce the block size from 128 MB (default) to 32 MB as otherwise, reading two data blocks leads to an out-of-memory error for the containers.

Pi 4 Cluster. For this cluster, we try different combinations of HDFS block size and memory allocated to mappers/reducers. This investigation gives an intuition on what kind of performance we are losing due to the memory constraints of the Pi 3 cluster. We test the following combinations of memory size (in MB) for Mapper, Reducer, and Block Size: 512/512/32; 1024/1024/128; 1024/2048/128; 2048/2048/128; 2048/2048/256; 2048/2048/512. We number the configurations from one to six.

4 EXPERIMENTAL RESULTS

In this section, we present our experimental results. We perform both IO microbenchmarks to evaluate the clusters' throughputs as well as the full TPCx-HS benchmark.

4.1 IO Benchmarks

We measure the IO throughput of the cluster nodes. The Pi 3 cluster uses Samsung Evo and SanDisk Ultra SD cards, while the Pi 4 cluster only relies on Evo+ SD cards. We use *dd* and write 50k blocks of size 8k to the device, using the `fsync` flag to force stores onto the device, instead of OS caches.

The results are shown in Figure 2. We also benchmark the SD cards in the notebook to see whether the Pis have a bottleneck on their interface. The results show that, for the Pi 3 cluster, the Samsung Evo card is faster than the Sandisk Ultra card on every platform and that the bandwidth is limited on the Pis compared to the notebook. We observe that the interface of the Pi 4 is slightly faster than the interface of the Pi 3B+.

In the Pi 3 cluster, two Pi 3B+ use the Samsung Evo card and the other two Pi 3B+ use the Sandisk Ultra card. Thus, we obtain an aggregated throughput of **66 MB/s**. For the Pi 4 cluster, we only use Samsung Evo+ SD cards. We calculate a throughput of **96 MB/s**.

Table 3: TPCx-HS results for scale factor 1 GB, running on the Pi 3 cluster.

	Run 1	Run 2
HSGen Elapsed Time (in seconds)	285	290
HSSort Elapsed Time (in seconds)	642	563
HSValidate Elapsed Time (in seconds)	102	105
Total Time (in seconds)	1 044	971
HSph@SF	0.0034	0.0037
$/HSph@SF	145 000.00	133 243.24
€/HSph@SF	120 979.41	111 170.27

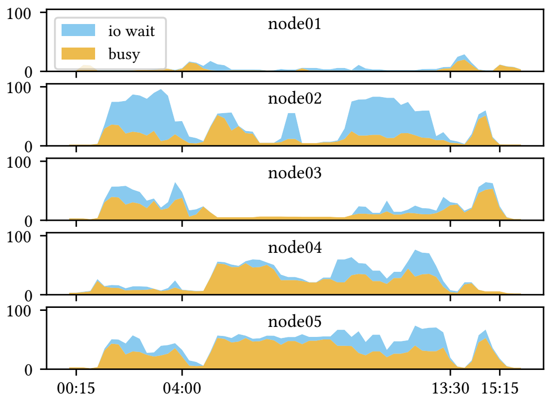
4.2 TPCx-HS

We perform two different experiments using TPCx-HS. In the first experiment, we use a scale factor of 0.001, which is equivalent to 1 GB of data being sorted. This is a good scale factor to test the general cluster and Hadoop setup as the tests finish quickly. In the second run, we increase the workload to a scale factor of 0.01, equivalent to 10 GB of data being sorted. This scale factor is harder to run, as the longer execution requires the cluster to sustain the load over a longer period of time.

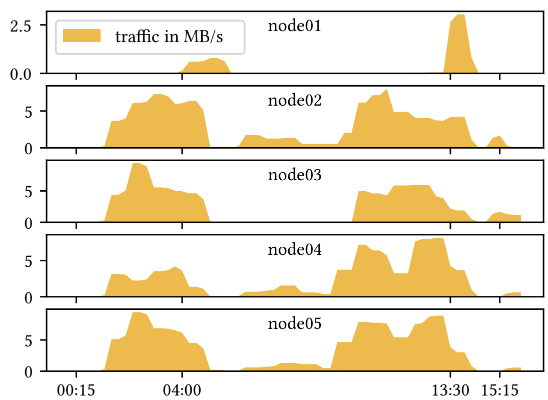
Scale Factor 1 GB @ Pi 3 Cluster. Please find the results in Table 3. To understand the behavior of the cluster during the benchmark, we log the CPU usage on all nodes during another run of TPCx-HS. The results are shown in Figure 3a. First, we can see that the NameNode is mostly active during data generation and data validation. The sort itself also utilizes the nodes differently. For example, node02 is often waiting for IO requests to finish. This is due to the distribution of the generated data. When checking where the blocks of the sorted files are saved, node02 was the node that was utilized the most. The high amount of IO waits suggest that IO is a bottleneck. Interestingly, some nodes are underutilized, e.g., node03, which only starts to get active towards the end of the run. On the other hand, node04 and node05 seem to perform a constant amount of processing.

In Figure 3b, you see the total network traffic on all nodes. It is clear that network is not a bottleneck of the benchmark. We are not near the network limits of the Raspberry Pi 3B+. HSSort only gets network-intensive during the last third of the execution. HSGen is also network intensive.

Scale Factor 10 GB @ Pi 3 Cluster. The results for a successful run can be seen in Table 4. While, as discussed in Section 3.3, there are no kernel crashes with the used configuration, the nodes still are



(a) CPU Utilization



(b) Network Traffic (in- and outbound)

Figure 3: Resource utilization of the Pi 3 cluster for HSGen, HSSort, and HSValidate (phases are marked).

Table 4: TPCx-HS results for scale factor 10 GB, running on the Pi 3 cluster.

	Run 1	Run 2
HSGen Elapsed Time (in seconds)	1 698	1 137
HSSort Elapsed Time (in seconds)	5 504	5 137
HSValidate Elapsed Time (in seconds)	304	304
Total Time (in seconds)	7 520	6 592
HSph@SF	0.0047	0.0054
$/HSph@SF	104 893.62	91 296.30
€/HSph@SF	83 261.70	72 468.52

under very high load and do not react to any other input using SSH or directly attached input devices during the runs.

Summary of Pi 3 Cluster Results. While we are able to get results for 1 GB and 10 GB runs, the 10 GB runs are less stable. Balancing the minimum container size with the block size for performance and stability is the most critical setting in configuration to prevent container crashes. The HSph metric for the 1 GB run is lower than for the 10 GB run, hence the cluster is better utilized for larger scale factors and overheads are amortized.

Table 5: TPCx-HS results for scale factor 1 GB, running on the Pi 4 cluster.

	Run 1		Run 2	
Config	HSph@SF	$/HSph@SF	HSph@SF	$/HSph@SF
Variant 1	0.0080	110 375.00	0.0086	102 674.42
Variant 2	0.0072	122 638.89	0.0069	127 971.01
Variant 3	0.0071	124 366.20	0.0072	122 638.89
Variant 4	0.0085	103 882.35	0.0109	81 009.17
Variant 5	0.0111	79 549.55	0.0104	84 903.85
Variant 6	0.0089	99 213.48	0.0099	89 191.92

Table 6: TPCx-HS results for scale factor 10 GB, running on the Pi 4 cluster.

	Run 1		Run 2	
Config	HSph@SF	$/HSph@SF	HSph@SF	$/HSph@SF
Variant 1	0.0115	76 782.61	0.0115	76 782.61
Variant 2	0.0088	100 340.91	0.0068	129 852.94
Variant 3	0.0087	101 494.25	0.0096	91 979.17
Variant 4	0.0122	72 377.05	0.0120	73 583.33
Variant 5	0.0114	77 456.14	0.0130	67 923.08
Variant 6	0.0100	87 300.00	0.0090	98 111.11

Scale Factor 1 GB @ Pi 4 Cluster. For both scale factors, we test each configuration variant discussed in Section 3.3 to derive a sweet spot of the Pi 4 cluster and validate whether larger containers lead to better performance. For each variant, we show the performance metric as well as the price-performance metric for both runs. The results are shown in Table 5.

We first note that the Pi 4 cluster outperforms the Pi 3 cluster in this scale factor for every possible configuration. The performance metric peaks in the first run of configuration variant 5. Regarding the performance runs, i.e., the runs for each configuration with the lower HSph value, configuration 5 also provides the best performance with an HSph value of 0.0104; it is three times better than the Pi 3 cluster for this scale factor.

Furthermore, note that the choice of the right configuration has a large impact on performance results. In comparison to variant 2, variant 5 is more than 50 % better with respect to the performance metric.

Scale Factor 10 GB @ Pi 4 Cluster. We again test each configuration variant and show the results in Table 6. Interestingly, for this scale factor, the variance between the different configuration variants is not as high as for 1 GB. Even variant 1 shows a large improvement over the Pi 3 cluster sorting 10 GB of data. Analogously to the Pi 3 cluster, this is explained by better utilization of the available resources when sorting 10 GB of data. The performance metric again peaks for variant 5, with an even higher HSph value of 0.0130. Compared to the 1 GB SF, this is an improvement of around 17 %, whereas for the Pi 3 cluster, the peak performance increased by 45 %. The best performance run is the second run of variant 4, though the differences between 4 and 5 are minor. We suspect that variants 4 and 5 perform best because they balance block and container sizes well. Larger block sizes would require even larger containers, in order to be used efficiently, and containers smaller than 2 GB can utilize smaller block sizes like 32 MB better, confirmed by the good results of variant 1 for both scale factors.

Energy Consumption. Using configuration 5 of the Pi 4 cluster, we measure the energy consumption of the 10 GB execution. This measurement includes the energy usage of the entire cluster, including the switch. In total, the cluster required 0.038 kWh of energy for the benchmark (two consecutive runs), where 0.02 kWh were required for the performance run (longer run). The Pi 3 cluster has a similar power, but runs much longer for the 10 GB benchmark. It requires 0.087 kWh for the entire benchmark (two runs). Unfortunately, there are no energy results published for TPCx-HS benchmarks on large clusters, to compare our measurements to.

Summary of Pi 4 Cluster Results. The Pi 4 cluster provides significant performance improvements over the Pi 3 cluster, offering performance values that are around three times better than the Pi 3 cluster values. For both scale factors, we see an improvement in performance when the container size is increased, if the block size is adjusted accordingly. For 2 GB-sized containers, the optimal block size is between 128 MB and 256 MB, while for the Pi 3 cluster, we had to reduce this to 32 MB. To use a block size of 512 MB effectively, we believe that even larger containers and/or more fine tuning of the memory settings (e.g., spillover settings [16]) are necessary to achieve benefits.

5 COMPARISON TO LARGE CLUSTERS

TPCx-HS benchmark results are officially not comparable across scale factors. However, we want to find out the relative performance of our Pi clusters to see if they provide a good insight into how a real big data cluster would perform (at much lower cost). Moreover, we want to compare the Pi clusters to bigger clusters from a price/performance standpoint. For comparison, we use the publicly available TPCx-HS results[2].

5.1 Throughput to Performance Ratio

We are interested in whether the ratio between the throughput of the system and the TPCx-HS result behaves similarly on large clusters and our Raspberry Pi clusters. To this end, we first analyze the *full disclosure report* of each 1 TB submission that uses MapReduce. For each submission, we estimate the throughput of the data nodes and then compare the ratio of the throughput to the HSph@SF metric. As there are no official throughput benchmarks for the TPCx-HS systems available, we assume a SAS HDD to have a throughput of 150 MB per second[3] and an NVME SSD to have a throughput of 2 GB per second[4]. As stated before, the Pi 3 cluster has a throughput of 66 MB per second, and the Pi 4 cluster has a throughput of 96 MB per second. We show the results in Figure 4. For the Pi 4 Pi cluster, we choose the configuration variants that provide the best performance run. Note that the throughput is normalized to GB per second. For example, the ratio for the Pi 4 cluster @ 1 GB is calculated as $\lfloor 0.096/0.0104 \rfloor$.

First of all, we notice that the ratio is in a reasonably small range between 6 and 19, which indicates that the systems' performance characteristics are indeed comparable. Especially the Pi 4 cluster shows a similar behavior to the large scale clusters, as it fits into the range from 6 to 10 given by the TPC systems. Note that this holds although the systems are very heterogeneous. Some systems only use NVME SSDs, like the Dell System 119091701, while others either use only HDDs (Cisco System 118013001) or hybrid-approaches (Cisco System 119121001). Our Pi clusters rely on MicroSD storage, while still maintaining a similar throughput-performance ratio. From these results, we can conclude that the Pi clusters can be used to test IO-bound big data approaches if the workload is scaled accordingly.

[2] http://www.tpc.org/tpcx-hs/results/tpcxhs_perf_results5.asp?version=2

[3] Modern enterprise SAS HDDs offer sustained write speeds up to 200 MB per second [12].

[4] The 8TB Intel P4510 NVMe used in Cisco System 119121001 even offers a sustained write speed of 3 GB per second [6].

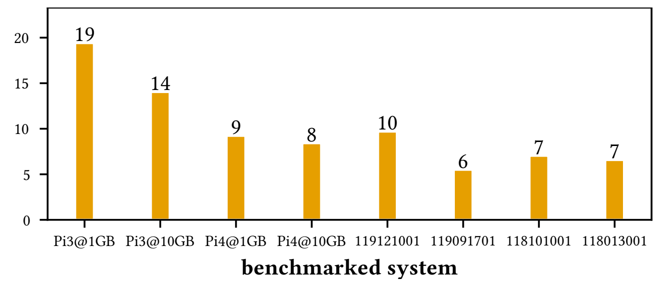


Figure 4: Throughput to performance ratio of the Pi clusters and official TPC results (result ID is shown).

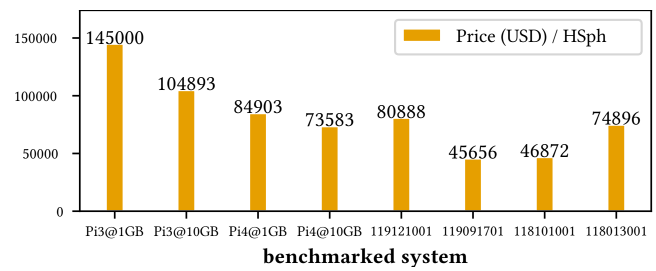


Figure 5: Price-Performance of the Pi clusters and official TPC results (result ID is shown).

5.2 Price-Performance Comparison

Next, we compare the price-performance value of the clusters. To this end, we use the official TPC Price/HSph metric, which is available on the TPCx-HS web page for each system. The results are shown in Figure 5. The Pi 3 cluster at 1 GB has the worst price-performance ratio of all systems. However, the HSph performance value of that Pi cluster is low compared to the other clusters; if we use the second run in Table 4, the Pi 3 Cluster @ 10GB already outperforms the Cisco system 119121001. Furthermore, at 1GB, the cluster is not evenly utilized. This is why the scale factor 10 GB shows a better price-performance behavior for the Pi 3 cluster. The Pi 4 cluster performs better in that regard, offering a reasonable price-performance characteristic for both scale factors and even beating the Cisco system for the 10 GB scale factor.

5.3 Summary of Comparison

The ratio between throughput and performance of the Pi clusters are within the same range as that of official TPC systems. The price-performance ratio of Pi clusters is comparable to TPC systems, but at the same time they have a significantly lower absolute cost. Our results show that Pis cannot only be used for scaled-down big data applications and cloud computing education; we expect that they will also be useful for analyzing data in fog and edge computing real-world use cases.

6 RELATED WORK

In this section, we give an overview of research on the design and evaluation of SoC clusters—mostly Raspberry Pi clusters—as well as on recent developments of the TPCx-HS benchmark.

SoC Clusters. Oracle recently presented a 1 060 node Raspberry Pi 3B+ cluster for demonstration purposes [4]. Academic research has been interested in Pi and SoC clusters since their release, with the Iridis-pi cluster being the first published paper to analyze one [5]. Several groups perform Message Passing Interface (*MPI*) related benchmarks on SoC-clusters [5, 9], stress the educational aspect of cluster computing via SoC-clusters [8, 9, 19], or employ SoC-clusters for big data workloads [1, 11, 14, 23]. In comparison to compute-bound benchmarks using MPI, big data workloads are often IO-bound, data-parallel problems. Kaewkasi et al. benchmark a Spark ARM SoC cluster and conclude that the *most-frequent-words* workload is IO-bound [14]. Anwar et al. evaluate different SoCs on different Hadoop-based workloads and motivate the future usage of ARM SoCs due to their high efficiency [1]. Both setups perform experiments on very low-power SoCs that exhibit large memory and CPU bottlenecks. Hajji and Tso conduct experiments on the impact of virtualization on big data and compute-bound workloads on a Pi cluster, with the result being that for big data workloads, the virtualization overhead is not as bad as for compute-bound workloads [11]. Srinivasan et al. evaluate the performance of an algorithm that employs OpenCV on a Raspberry Pi 3 cluster using Hadoop [23]. Mühlbauer et al. compare classical database benchmarks on a hybrid DBMS for ARM and x86 clusters [17].

Big Data Benchmarks. TPCx-HS was the first big data benchmark standardized by the TPC. It is based on TeraSort, which is part of the Hadoop framework. TeraSort is one of the default test workloads for Hadoop environments because it can fully utilize the hardware and makes it possible to find deficiencies in configuration and setup. Since TPCx-HS's development, the TPC has issued further benchmarks for big data workloads. Most notably, TPCx-BB, the standardized version of the BigBench benchmark [10]. In contrast to TPCx-HS, TPCx-BB is an end-to-end big data analysis benchmark with a rich set of queries that measure analysis performance on structured, semi-structured, and unstructured data. Recently, the TPC changed TPC-DS, a decision support benchmark, to be compatible with big data frameworks [20]. TPC-DS is a purely SQL-based benchmark, which challenges not only the hardware but also the query processor and optimizer. There are several other big data benchmarks, many of which include TeraSort as a workload. An overview can be found in Ivanov et al. [13]. To the best of our knowledge, we are the first to run and analyze industry-standard data processing benchmarks on low-end hardware.

7 CONCLUSIONS

In this paper, we conduct a feasibility study of running the TPCx-HS benchmark on two different Raspberry Pi clusters. From our results, we see that their behavior and hardware characteristics are comparable to a big data cluster on primary metrics, such as the throughput-to-performance ratio. The new Pi 4 generation resolves the memory bottleneck of the 3B+. While the CPU power increased, the Pi 4 offers an 8 GB memory model, and hence is well suited for data processing applications. This increase in available memory is a game-changer for SoC data processing and with the release of Hadoop 3.3.0 in July 2020, the Hadoop project has already started to adopt this movement as it starts to support 64-bit ARM systems officially. Another example of the rising importance of the ARM architecture for data processing is the release of Amazon AWS Gavitron, which is providing cloud instances based on ARM CPUs. Because of their low absolute cost–while exhibiting a reasonable price-performance ratio–we expect Pis to be used in various computing workloads, like sensor data aggregation as well as scaled big data workloads. Pis enable easy usage of ARM chips in an on-premise setup.

REFERENCES

[1] A. Anwar, K. R. Krish, and A. R. Butt. 2014. On the Use of Microservers in Supporting Hadoop Applications. In *CLUSTER*.
[2] Apache Software Foundation. 2020. Apache Hadoop YARN. https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html
[3] Apache Software Foundation. 2020. Hadoop. https://hadoop.apache.org/
[4] C. Coward. 2019. Oracle Builds a Supercomputer with 1,060 Raspberry Pi Computers. https://www.hackster.io/news/oracle-builds-a-supercomputer-with-1-060-raspberry-pi-computers-6da1c3e2f8d9
[5] S. Cox, J. Cox, R. Boardman, S. Johnston, M. Scott, and N. O'Brien. 2013. Iridis-pi: A Low-Cost, Compact Demonstration Cluster. *Cluster Computing* 17, 2 (2013).
[6] S. Crisp. 2019. Intel SSD DC P4510 8TB Review. https://www.kitguru.net/components/ssd-drives/simon-crisp/intel-ssd-dc-p4510-8tb-review/all/1/
[7] J. Dean and S. Ghemawat. 2008. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM* 51, 1 (2008), 107–113.
[8] G. Dorr, D. Hagen, B. Laskowski, E. Steinmetz, and D. Vo. 2017. Introduction to Parallel Processing with Eight Node Raspberry Pi Cluster. In *Midwest Instruction and Computing Symposium*.
[9] K. Doucet and J. Zhang. 2017. Learning Cluster Computing by Creating a Raspberry Pi Cluster. In *ACM SouthEast Regional Conference*.
[10] A. Ghazal, T. Rabl, M. Hu, F. Raab, M. Poess, A. Crolotte, and H. Jacobsen. 2013. BigBench: Towards an Industry Standard Benchmark for Big Data Analytics. In *SIGMOD*.
[11] W. Hajji and P. Tso. 2016. Understanding the Performance of Low Power Raspberry Pi Cloud for Big Data. *Electronics* 5, 2 (2016).
[12] Hardware-Mag. 2019. Seagate Exos X16 HDD Test. https://www.hardware-mag.de/artikel/festplatten_ssds/seagate_exos_x16_dd_mit_16_tb_im_test/7/
[13] T. Ivanov, T. Rabl, M. Poess, A. Queralt, J. Poelman, N. Poggi, and J. Buell. 2015. Big Data Benchmark Compendium. In *TPCTC*.
[14] C. Kaewkasi and W. Srisuruk. 2014. A Study of Big Data Processing Constraints on a low-power Hadoop Cluster. In *ICSEC*.
[15] S. Kumar and A. Jasuja. 2017. Air Quality Monitoring System based on IoT using Raspberry Pi. In *ICCCA*.
[16] T. Lipcon. 2012. Optimizing MapReduce Job Performance. https://de.slideshare.net/cloudera/mr-perf
[17] T. Mühlbauer, W. Rödiger, R. Seilbeck, A. Reiser, A. Kemper, and T. Neumann. 2014. One DBMS for all: the Brawny Few and the Wimpy Crowd. In *SIGMOD*.
[18] R. Nambiar, M. Poess, A. Dey, P. Cao, T. Magdon-Ismail, D. Qi Ren, and A. Bond. 2014. Introducing TPCx-HS: The First Industry Standard for Benchmarking Big Data Systems. In *TPCTC*.
[19] A. M. Pfalzgraf and J. A. Driscoll. 2014. A Low-Cost Computer Cluster for High-Performance Computing Education. In *EIT*. 362–366.
[20] M. Poess, T. Rabl, and H. Jacobsen. 2017. Analysis of TPC-DS: The First Standard Benchmark for SQL-based Big Data Systems. In *SOCC*.
[21] O. Quinlan and S. Baloro. 2018. Raspberry Pi Computers in Schools. https://www.raspberrypi.org/app/uploads/2018/08/Raspberry-Pi-Computers-in-Schools-2018.pdf
[22] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. 2010. The Hadoop Distributed File System. In *IEEE Symposium on Mass Storage Systems and Technologies*.
[23] K. Srinivasan, C. Chang, C. Huang, M. Chang, A. Sharma, and A. Ankur. 2018. An Efficient Implementation of Mobile Raspberry Pi Hadoop Clusters for Robust and Augmented Computing Performance. *J. Inf. Process. Syst.* 14, 4 (2018), 989–1009.
[24] Transaction Processing Performance Council (TPC). 2018. TPC Express Benchmark HS (TPCx-HS) - Standard Specification.