DOI: 10.15514/ISPRAS-2020-32(3)-5

Hardware and software data processing system for research and scientific purposes based on Raspberry Pi 3 microcomputer

P.A. Pankov, ORCID: 0000-0002-4007-451X <pankov.pavel.a@gmail.com>
I.V. Nikiforov, ORCID: 0000-0003-0198-1886 <igor.nikiforovv@gmail.com>
D.F. Drobintsev, ORCID: 0000-0001-7876-3313 <drobintsev@mail.ru>

Peter the Great St.Petersburg Polytechnic University,
29, Polytechnicheskaya, St.Petersburg, 195251, Russia

Abstract. In the past ten years, rapid progress has been observed in science and technology through the development of smart mobile devices, workstations, supercomputers, smart gadgets and network servers. Increase in the number of Internet users and a multiple increase in the speed of the Internet led to the generation of a huge amount of data, which is now commonly called «big data». Given this scenario, storing and processing data on local servers or personal computers can cause a number of problems that can be solved using distributed computing, distributed data storage and distributed data transfer. There are currently several cloud service providers to solve these problems, like Amazon Web Services, Microsoft Azure, Cloudera and etc. Approaches for distributed computing are supported using powerful data processing centers (DPCs). However, traditional DPCs require expensive equipment, a large amount of energy to run and operate the system, a powerful cooling system and occupy a large area. In addition, to maintain such a system, its constant use is necessary, because its stand-by is economically disadvantageous. The article is aimed at the possibility of using a Raspberry Pi and Hadoop cluster for distributed storage and processing of «big data». Such a trip provides low power consumption, the use of limited physical space, high-speed solution to the problems of processing data. Hadoop provides the necessary modules for distributed processing of big data by deploying Map-Reduce software approaches. Data is stored using the Hadoop Distributed File System (HDFS), which provides more flexibility and greater scalability than a single computer. The proposed hardware and software data processing system based on Raspberry Pi 3 microcomputer can be used for research and scientific purposes at universities and scientific centers. Considered distributed system shows economically efficiency in comparison to traditional DPCs. The results of pilot project of Raspberry Pi cluster application are presented. A distinctive feature of this work is the use of distributed computing systems on single-board microcomputers for academic purposes for research and educational tasks of students with minimal cost and ease of creating and using the system.

Keywords: data processing; data storage; big data; cluster; supercomputer; Raspberry Pi; Hadoop

For citation: Pankov P.A., Nikiforov I.V., Drobintsev D.F. Hardware and software data processing system for research and scientific purpose based on Raspberry Pi 3 microcomputer. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 3, 2020, pp. 57-70. DOI: 10.15514/ISPRAS–2020–32(3)–5

Программно-аппаратный комплекс обработки данных для исследовательских и научных целей с использованием микрокомпьютера Raspberry Pi 3

П.А. Панков, ORCID: 0000-0002-4007-451X <pankov.pavel.a@gmail.com>
И.В. Никифоров, ORCID: 0000-0003-0198-1886 <igor.nikiforovv@gmail.com>
Д.Ф. Дробинцев, ORCID: 0000-0001-7876-3313 <drobintsev@mail.ru>

Санкт-Петербургский политехнический университет Петра Великого,
195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29

Abstract. В последние десять лет наблюдается быстрый прогресс в науке и технологиях благодаря разработке интеллектуальных мобильных устройств, рабочих станций, суперкомпьютеров, интеллектуальных гаджетов и сетевых серверов. Увеличение числа пользователей интернета и многократное увеличение скорости интернета привело к генерации огромного количества данных, которые сейчас обычно называют «большими данными». При таком сценарии хранение и обработка данных на локальных серверах или персональных компьютерах может вызвать ряд проблем, которые могут быть решены с помощью распределенных вычислений, распределенного хранения данных и распределенной передачи данных. В настоящее время существует несколько провайдеров облачных услуг для решения этих проблем, таких как Amazon Web Services, Microsoft Azure, Cloudera и т. Д. Подходы к распределенным вычислениям поддерживаются с помощью мощных центров обработки данных (ЦОД). Однако традиционные ЦОДы требуют дорогого оборудования, большого количества энергии для работы и эксплуатации системы, мощной системы охлаждения и занимают большую площадь. Кроме того, для поддержания такой системы необходимо ее постоянное использование, поскольку ее резервирование экономически невыгодно. Целью статьи является возможность использования кластера Raspberry Pi и Hadoop для распределенного хранения и обработки «больших данных». Такое отключение обеспечивает низкое энергопотребление, использование ограниченного физического пространства, быстрое решение проблем обработки данных. Hadoop предоставляет необходимые модули для распределенной обработки больших данных путем развертывания программных подходов MapReduce. Данные хранятся с использованием распределенной файловой системы Hadoop (HDFS), которая обеспечивает большую гибкость и большую масштабируемость, чем один компьютер. Предлагаемая аппаратно-программная система обработки данных на базе микрокомпьютера Raspberry Pi 3 может быть использована для исследовательских и научных целей в университетах и научных центрах. Рассмотренная распределенная система демонстрирует экономическую эффективность по сравнению с традиционными ЦОД. Представлены результаты пилотного проекта применения кластера Raspberry Pi. Отличительной особенностью данной работы является использование распределенных вычислительных систем на одноплатных микрокомпьютерах для академических целей для исследовательских и учебных задач учащихся с минимальными затратами и простотой создания и использования системы.

Ключевые слова: обработка данных; хранение данных; большие данные; кластер; суперкомпьютер; Raspberry Pi; Hadoop

Для цитирования: Панков П.А., Никифоров И.В., Дробинцев Д.В. Программно-аппаратный комплекс обработки данных для исследовательских и научных целей с использованием микрокомпьютера Raspberry Pi 3. Труды ИСП РАН, том 32, вып. 3, 2020 г., стр. 57-70 (на английском языке). DOI: 10.15514/ISPRAS–2020–32(3)–5

1. Introduction

Today, huge amounts of data are being generated, the source of which is social networks, meteorological organizations, corporate firms, scientific and technical institutions, web services, smart IoT devices [1], etc. Therefore, the development of tools for storing, processing and restoring information from huge volumes of data is today one of the most important issues in the research of information technology [2]. In order to meet the growing need for storage, manipulation and recovery of information, new data centers are being created.

Traditional data centers do their job well for commercial purposes, but have a number of disadvantages:

- consist of powerful hardware that is expensive;
- require a large amount of electricity to work;
- require powerful cooling;
- occupy a large area.

In addition, the use of powerful equipment provides for its continuous workload, since the operation of such systems without performing useful tasks is expensive.

Usually, big data means big sets of huge amounts of data that are difficult to work with using traditional data management tools, because of their huge size and complexity [3].

The inevitable problems of big data include the fact that the infrastructure necessary to process huge amounts of data must be created using limited resources and strictly limited processing time periods. In addition, extracting features from such data requires the use of clusters and complex data processing applications [4]. It is often necessary to work with similar data in real time.

In addition, these data centers must have capabilities such as extreme scalability, data distribution, load balancing, fault tolerance, etc. To solve these problems, Jeff Dean and Sanjay Ghemavat created the MapReduce model [5] for processing large amounts of data on large clusters.

Apache Hadoop [6, 7] is a project of the Apache Software Foundation, an open source and freely distributed set of utilities, libraries and frameworks for developing and running distributed programs running on clusters. Apache Hadoop v2.0 consists of four main modules – HDFS (Hadoop Distributed File System), Hadoop Common (a set of software libraries and utilities), YARN (Yet Another Resource Negotiator) and Hadoop MapReduce (software framework for easily writing applications which process vast amounts of data in-parallel).

Apache Hadoop is considered one of the main technologies for interacting with big data.

A single-board computer (SBC) is a universal computer that is built on one printed circuit board together with the required processor, memory, I/O ports and other functions necessary for a well-designed computer [8]. Raspberry Pi is an inexpensive and most common single board computer. An important contribution of this study is the use of the Raspberry Pi single-board computer with Hadoop clusters, which provides parallel and distributed processing with increased performance and fault tolerance.

The system under development is considered for academic purposes for research and scientific purposes. The goals also include training employees or students to work with cluster infrastructure.

In addition, it is important to provide the ability to verify the work of the developed data processing algorithms, including in enterprises, without using the capabilities of systems for production purposes.

The main goal of the project is to create a cheap solution for academic use. This is the main feature of the project, compared with the existing solutions under consideration.

Section 2 shows the related work. Section 3 shows system design. Section 4 shows the implementation process and result of the pilot project. Section 5 presents the conclusion.

2. Review of related works

First of all, let’s take a look at relatives works that use single-board computers (SBC) as a main computational unit.

2.1 Single-board computers review and selection

A single board computer (SBC) is a complete, self-contained computer. The difference between SBC and traditional personal computer is that SBC is assembled on one printed circuit board, on which all the devices necessary for the functioning of the device are installed:

- CPU;
- RAM;
- video memory;
- input-output interfaces;
- etc.

This approach to the manufacture of a single-board computer allows this to make it inexpensive and compact. In addition, the device becomes even cheaper through the use of systems on a chip (SoC). On the other hand, expanding capabilities by changing the processor, increasing the amount of memory and replacing other hardware components is impossible, since most of all these components are soldered to the board. On the other hand, these features of SBC make it possible to use them as industrial computers or as computers for embedded systems.

The big advantage of SBC is that they can easily be used as a system module from many of these modules, due to the fact that all the necessary components are on the same printed circuit board. This allows quick replacement of a broken assembly. It is enough to take a new SBC, insert an SD card with an operating system into it and connect the power and Ethernet wires.

Another advantage of using SBC is the general purpose input / output interface (GPIO) ports. A GPIO is an interface for interaction between components, for example, between a microcontroller or microprocessor and various peripherals. Most often, GPIO contacts can work both input and output, with some exceptions. The presence of GPIO ports allows you to use a SBC in embedded systems to read data from various external sensors (temperature, humidity, infrared radiation, angular speeds, accelerations, voltage, current, etc.) and control external devices (LCD displays, servos, DC motors, electric drives, LEDs and LED strips, etc.)

The following single-board microcomputers were selected for consideration:

- Banana Pi M1+;
- Orange Pi PC2;
- Raspberry Pi 3 Model B+.

There are a large number of different models of single-board computers. Table 1 compares several SBC with a similar price. As the comparison criteria were selected:

- CPU (K1);
- RAM (K2);
- network access interfaces (K3);
- supported operating systems (K4);
- price (K5).

Table 1. Comparative analysis of SBCs

	Banana Pi M1+	**Orange Pi PC2**	**Raspberry Pi 3 Model B +**
K1	A20 ARM Cortex -A7 Dual-Core 1GHz	Allwinner Cortex-A53 64-bit Quad-Core 1.2 GHz	Broadcom Cortex-A53 64-bit Quad-Core 1.4GHz
K2	1 Gb DDR3	1 Gb DDR3	1 Gb LPDDR2
K3	Wi-Fi, Ethernet	Ethernet	Wi-Fi, Ethernet
K4	Android Armbian Debian Other Linux	Android Ubuntu Debian	Raspbian Ubuntu Mate Ubuntu Core Win10 IoT
K5	3000 – 4000 rub.	2900 – 3400 rub.	3000 – 4000 rub

Based on a comparative analysis, the Raspberry Pi is the optimal choice. In addition, the Raspberry Pi is the most common SBC, which makes development easier due to community support. In addition, the availability of Raspberry Pi in many electronics stores is a significant advantage over other SBCs.

In addition to the presented single-board computers, there are also Odroid. These single-board computers have better characteristics and greater performance, but they have a higher cost, greater power consumption, greater heat dissipation and require better cooling. In addition, Odroid is less accessible and the developer community is many times smaller than the Raspberry Pi developer community.

Raspberry Pi clusters were implemented to solve some business, scientific and academic problems. The SBC Raspberry Pi offers competitive advantages: they are inexpensive, low power, and at the same time offer features similar to a simple personal computer.

2.2 Raspberry Pi clusters

Next we provide a review of several articles and projects that conduct research on the effectiveness of using Raspberry Pi based cluster.

2.2.1 Beowulf cluster

The paper [9] presents a performance benchmarking of a Raspberry Pi 2 cluster (fig. 1). The research project shows the design and construction of a high performance cluster of 12 Raspberry Pi 2 Model B single-board computers. The Raspberry Pi 2 Model B is the second-generation Raspberry Pi. It has:

- ARM Cortex-A7 CPU 900 MHz;
- 1 Gb RAM.

All of the nodes are connected over an Ethernet 100 Mbps Network in a parallel mode. Test performed using High Performance Linpack (HPL) benchmark.

As a result of their research, the authors collected cluster performance metrics in GFlops for different number of nodes and different problem sizes.

Fig. 1. Construction of Beowulf Cluster [9]

2.2.2 Iridis-Pi cluster

The project of Iridis-Pi [10] used the Raspberry Pi (one) Model B microcomputer (fig. 2). The cluster had the following characteristics:

- 64 Raspberry Pi Model B nodes;
- Broadcom BCM2835 700 MHz;
- 512 Mb RAM.

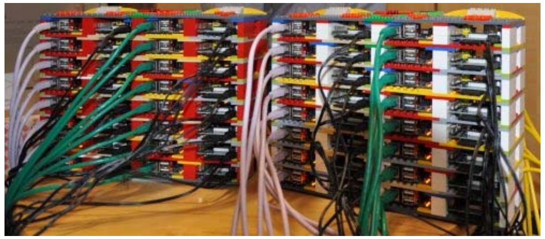


Fig. 2. Construction of Irdis Pi [12]

Like the previous project, this one uses LINPACK to test single-node performance and High-Performance LINPACK (HPL) to test cluster performance. In addition, SD card performance was measured. This is also important since the operating system and all files with which raspberry pi works are recorded on these cards.

2.3.3 Raspberry Pi Hadoop cluster and FAST algorithm

In the project [11], the Raspberry Pi Hadoop cluster is used in more realistic conditions. To test cluster performance, the authors run on it the SURF algorithm from the OpenCV library. The SURF algorithm is used to search for fixed objects (retaining their external attributes) in images using the characteristic points of the object. The similar use of the Raspberry Pi cluster in a similar task is a very illustrative example, since image processing requires high performance.

In their research, the authors compared the work of an ordinary desktop computer and a raspberry pi cluster with a different number of nodes and a different amount of data.

The result showed that the effectiveness of the built cluster occurs only with large amounts of data (in the case of the project, the required amount of data was from 64,000 and above).

2.3.4 Traditional DPC and single-board computer DPC

An important part of the research is the comparison of traditional data processing centers and data processing centers based on single-board microcomputers. As projects for comparison the last article (B) and the cluster of the higher school of software engineering of St. Petersburg Polytechnic University (A) were taken.

For comparison, the following criteria were identified:

- types of tasks to be solved (K1);
- examples of using (K2);
- hardware (K3);
- software (K4);
- hardware price (K5);
- areas of use (K6).

Comparison is presented in the Table 2.

Table 2. Traditional cluster and SBC cluster comparison

	Traditional cluster	**SBC cluster**
	A	***B***
K1	Students laboratory and course works on big data processing	Big data processing using computer vision algorithms
K2	Hadoop Distributed data processing and storage	Hadoop Image processing using the SURF algorithm

K3	- 4 Intel Xeon 6 core E5 2620 2GHz - 32 TB HDD - 256 Gb RAM	- 5 – 10 RPi 3 Model B 1.2 GHz * - 80 – 320 Gb + HDD * - 5 – 10 Gb RAM
K4	Linux Hadoop	Ubuntu Hadoop, OpenCV
K5	~ 450 000 rub.	~ 25 000 - 45 000 rub
	Traditional cluster	**SBC cluster**
	A	***B***
K6	Production Research Commercial use	Research Education Science Academic use
* depending on the number of nodes		

As we can see from the table, the advantages of using a cluster on Raspberry Pi for research and academic purposes, since it is economically more profitable. Such a system fully fulfills the functionality of a software-hardware system for distributed storage and processing of data, and high performance is not so important for research and academic purposes, unlike commercial use.

In addition, a rather important task is to create a system that is cost-effective during downtime, as traditional data centers use hardware that consumes a large amount of energy and generates a large amount of heat.

3. System description

This section describes the architecture of the project. Below will be described the hardware and software that are used.

3.1 Hardware: Raspberry Pi 3 Model B +

Raspberry Pi [12] 3 Model B + (fig. 3) is the third generation of Raspberry Pi SBCs.

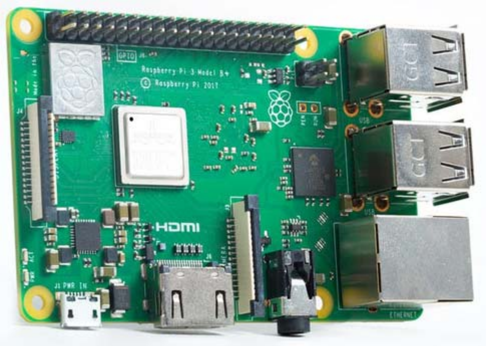


Fig. 3. Raspberry Pi 3 Model B plus[1]

Raspberry Pi was originally developed as a budget platform for learning computer science, but later gained wider fame and scope.

[1] https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

3.2 Software

A cluster requires a certain set of software. First of all, nodes need an operating system. It is necessary to install a set of programs on the operating system that will allow you to create a distributed file system and perform distributed computing. Next will be described the software that was used during the research.

3.2.1 Raspbian OS

Raspbian OS [13, 14] is the main operating system for the Raspberry Pi based on the Debian Linux operating system. Raspbian was originally created by Mike Thompson and Peter Green as an independent project. The system is optimized for operation on low-performance ARM processors. PIXEL (Pi Improved Xwindows Environment, Lightweight) is used as the desktop environment.

3.2.2 Hadoop

Apache Hadoop is the open source project by the Apache Software Foundation. Hadoop is used for reliable and scalable distributed computing, but can also serve as a distributed storage of large amounts of data. Many companies use Hadoop for production and scientific purposes.

Hadoop consists of four key components:

- HDFS. Hadoop Distributed File System is a distributed file system that is responsible for storing data on a Hadoop cluster;
- MapReduce system, which is designed for computing and processing large amounts of data on a cluster;
- Hadoop Common – this module provides the tools (written in the Java language) needed on the user's operating systems (Windows, Unix, or others) to read data stored in the Hadoop file system;
- YARN module manages the resources of systems that store data and perform analysis.

HDFS. Hadoop Distributed File System (HDFS) is the primary storage system used by Hadoop. HDFS repeatedly copies data blocks and distributes these copies to the computing nodes of the cluster, thereby ensuring high reliability and speed of calculations:

- data is distributed across several machines at boot time;
- HDFS is optimized more for streaming file reads than for irregular, random reads;
- files in the HDFS system are written once and making any arbitrary entries in the files is not allowed;
- applications can read and write HDFS files directly through the Java programming interface.

MapReduce. MapReduce is a programming model and framework for writing applications designed for high-speed processing of large amounts of data on large parallel clusters of computing nodes:

- provides automatic parallelization and distribution of tasks;
- has built-in mechanisms to maintain stability and performance in case of failure of individual elements;
- provides a clean level of abstraction for programmers.

Other tools. However, Hadoop has a number of other tools. Here is some of them:

- HBase – NoSQL database that supports random read and write;
- Pig – data processing language and runtime;
- Spark – a set of tools for implementing distributed computing;
- Hive – data warehouse with SQL interface;
- ZooKeeper – storage of configuration information.

It is important that the Hadoop software allows you to use horizontal scaling. Horizontal scaling allows to reduce the execution time of the same tasks.

4. Implementation

A cluster of four Raspberry Pi nodes was created for research and a pilot project. Fig. 4 shows the assembled cluster. It was decided to use different models of the Raspberry Pi single-board microcomputer, thereby creating a heterogeneous cluster.

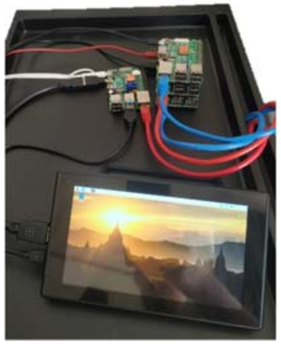


Fig. 4. Example of an assembled system of four nodes

On the created cluster, the word count algorithm using Hadoop MapReduce was tested. The Hadoop license file was used as a test file.

Table 3 shows some collected metrics from the test bench.

Table 3. Hadoop cluster performance metrics

Metric name	Value
HDFS number of bytes read	147239
HDFS number of bytes written	34796
HDFS number of read operations	8
HDFS number of write operations	2
Total time spent by map tasks (ms)	66853
Total time spent by reduce tasks (ms)	21310
Map input / output records	2746 / 21463
Combine input / output records	21463 / 2965
Reduce input / output records	2965 / 2965

The presented metrics were obtained for the operation of the cluster from one node, the second node did not participate in data processing due to configuration settings.

Next, we collected the time metrics for the algorithm for counting words in the text with different system configurations.

Tests were carried out in several stages. The algorithm was launched taking into account the fact that the text file was not divided into blocks. Further, the file system configuration was configured so that the file was divided into 3 blocks, the work with which was distributed across different nodes. Various configurations of the operating modes of the nodes were also tested. Three single-board computers performed a different role at each stage. At each stage there was 1 «master node» – it is engaged in the distribution of tasks and monitoring nodes. In addition, the number of «work nodes» that are responsible for data processing has changed. At stages 1, 3 and 5, the “master node” was at the same time a «working node».

Table 4 shows the processing time for a 38.5-megabyte file with various system configurations.

Table 4. The collected metrics of the test algorithm for calculating words in the text

	1 block	3 blocks
1 worker	3min 37sec	---
1 master 1 worker	2min 38sec	2min 13sec
2 workers	2min 44sec	2min 15sec
1 master 2 workers	2min 46sec	2min 14sec
3 workers	2min 49sec	2min 13sec

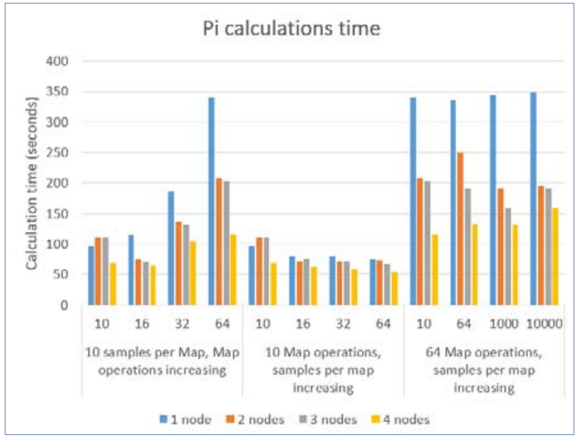
As can be seen from the table, when working on one node, the time of work with one block is the greatest. The best time was shown by the configuration of one «master node» and one «work node». In other conditions, since the file is not divided into blocks, we only lose time on the work of the «master node» for the distribution of tasks. In the case of splitting the file into 3 blocks, the execution time is approximately the same, since in each case all 3 blocks were immediately distributed to all nodes, however, the reduction in the operating time of the algorithm compared to 1 block is visible.

The second basic algorithm for checking the operation of distributed computing systems is distributed computing the value of Pi. Table 5 shows parameters of Pi calculation tests.

Table 5. Test Parameters

Constant parameters	Variable parameters
10 samples per Map operation	10
	16
	32
	64
10 Map operations	10
	16
	32
	64
64 Map operations	10
	64
	1000
	10000

The first results will present a graph of the dependence of the execution time of calculations on a different number of nodes and various settings that are indicated in the previous table. The graph is shown in fig. 5.



Fig, 5. Pi calculation time

In addition, a comparison was made of the time spent on one Map operation. The results are presented in fig. 6.

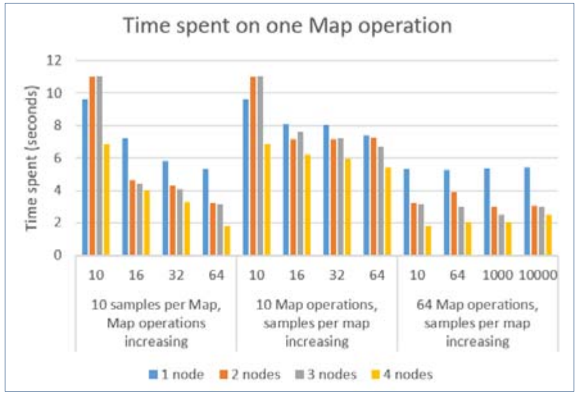


Fig. 6. Time spent on one Map operations

A distinctive feature of this work is the use of distributed computing systems on single-board microcomputers for academic purposes for research and educational tasks of students with minimal cost and ease of creating and using the system.

In addition, as a result of the study, a number of features and disadvantages of using the Raspberry Pi were identified.

- Such a cluster is effective in performing lightweight tasks, for which there is the possibility of splitting them into a large number of small tasks that do not require large computing power. The same feature leads to the fact that this cluster is not effective in such tasks where high performance is needed, for example, when working with graphics.
- SD card. Since the operating system is on an SD card, this can be a vulnerability, since the SD cards do not differ in high performance;
- During the tests, it was noticed that during prolonged operation of the cluster, the number of errors that occur during the execution of tasks increases, which may be associated with the accumulation of garbage files. To restore stability, a reboot of the cluster was required. This leads to the need for a more detailed approach to cluster configuration.
- In a situation when the number of nodes increases, errors may occur due to the fact that the node does not have enough memory to allocate resources that the main node requests from the node. The solution to this problem is the addition of certain configuration parameters to the files of the main node. It is necessary to indicate to the main node about the need to check the availability of the requested both virtual and hardware resources. In addition, you should correctly configure the operating system itself on each node, including the amount of allocated memory for Java, which may affect the operation of the cluster. It is necessary to approach in detail the configuration of various nodes that differ in hardware characteristics.
- For comfortable operation, the Raspberry Pi requires active cooling. However, a single fan is sufficient to cool two SBC’s. The fan used was powered from 5 volts and consumed 0.06 amperes, which equals a power of 0.3 watts, which is energy efficient.

5. Conclusion

As a result of the research, the following tasks were completed:

- research and analysis of existing projects on the use of SBCs within the cluster;
- comparison of the cost-effectiveness of a traditional data center and data center using SBCs for research and academic purposes. Based on this comparison, the relevance of developing a system on SBCs was revealed;
- comparison of SBCs. As a result of the comparison, the Raspberry Pi 3 Model B + single-board computer was chosen for the project, since it is the most optimal, due to its characteristics, price, and also availability and prevalence.

From the results of the test benchmarks it can be seen that the created system supports horizontal scalability, which meets the system requirements. In addition, based on the results, it can be concluded that the goal of creating a cheap, scalable distributed computing system for academic purposes has been achieved.

5. Future work

The project is under development. In the future, it is planned to perform the following tasks:

- increase the number of nodes for analysis to increase productivity;
- increase the number of metrics;
- analyze the performance of SD cards from different manufacturers, as their characteristics affect the operation of the Raspberry Pi;
- use an external USB (flash / HDD / SSD) drive (s) as storage media;
- analyze the efficiency of using a cluster on single-board computers in various tasks;
- use software tools for testing distributed applications and systems [15].

References / Список литературы

[1]. P. Pankov, I. Nikiforov, Y. Zhang. Hardware and software system for collection, storage and visualization meteorological data from a weather stand. In Proc. of the International Scientific Conference on Telecommunications, Computing and Control (TELECCON-2019), 2019 (in printing).

[2]. A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: research problems in data center networks. ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, 2008, pp. 68-73.

[3]. P. Zikopoulos, C. Eaton, D. deRoos, T. Deutsch, and G. Lapis, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw-Hill, 2011, 176 p.

[4]. L. Xue, J. Ni, Y. Li, and J. Shen. Provable data transfer from provable data possession and deletion in cloud storage. Computer Standards & Interfaces, vol. 54, 2017, pp. 46-54.

[5]. J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. in Proc. of the 6th Symposium on Operating System Design and Implementation (OSDI), 2004, pp. 137-150.

[6]. C. Lam. Introducing Hadoop. In Hadoop in Action. Manning Publications, 2011, pp. 3-20.
[7]. Tom White. Hadoop: The Definition Guide. 4th Edition, O'Rilley Media Inc., 2015, 688 p.
[8]. W. P. Birmingham and D. P. Siewiorek. MICON: a knowledge based single board computer designer. In Proc, of the 21st Conference on Design Automation, 1984, pp. 565-571.
[9]. Dimitrios Papakyriakou, Dimitra Kottou and Ioannis Kostouros. Benchmarking Raspberry Pi 2 Beowulf Cluster. International Journal of Computer Applications, vol. 179, no. 32, 2018, pp. 21-27.
[10]. Simon J. Cox et al. Irdis-Pi: A low-cost, compact demonstration cluster. Cluster Computing, vol. 17, no. 2, 2013, pp. 349–358.
[11]. Kathiravan Srinivasan et al. An Efficient Implementation of Mobile Raspberry Pi Hadoop Clusters for Robust and Augment Computing Performance. Journal of Information Processing Systems, vol.14, no. 4, 2018, pp. 989-1009.
[12]. Molloy Derek. Exploring Raspberry Pi. Interfacing to the Real World with Embedded Linux. Wiley, 2016, 720 p.
[13]. William Harrington. Learning Raspbian. Packt Publishing, 2015, 154 p.
[14]. Roberto Morabito. Virtualization on Internet of Things Edge Devices with Container Technologies: a Performance Evaluation. IEEE Access, vol. 5, 2017, pp. 8835-8850.
[15]. Kobyshev K.S., Nikiforov I.V., Prokofyev O.V. Tool for automating testing components of a distributed application using an esb bus. In Proc. of the Scientific and Practical Conference on Modern Technologies in Theory and Practice of Programming, 2019, pp. 212-214 (in Russian) / Кобышев К.С., Никифоров И.В., Прокофьев О.В. Инструмент для автоматизации тестирования компонент распределенного приложения, использующего ESB-шину. Труды научно-практической конференции «Современные технологии в теории и практике программирования», 2019 г., стр. 212-214.

Information about authors / Информация об авторах

Pavel Aleksandrovich PANKOV is a graduate student at the Higher School of Software Engineering at the Institute of Computer Science and Technology. Research interests: big data, data analysis, computer vision, robotics, automation of technological processes, embedded systems.

Павел Александрович ПАНКОВ – студент магистратуры Высшей школы программной инженерии Института компьютерных наук и технологий. Сферы научных интересов: большие данные, анализ данных, компьютерное зрение, робототехника, автоматизация технологических процессов, встраиваемые системы.

Igor Valerievich NIKIFOROV – candidate of technical sciences, associate professor of the Higher School of Software Engineering at the Institute of Computer Science and Technology. Research interests: parallel data processing, distributed data storage systems, big data, software verification, test automation.

Игорь Валерьевич НИКИФОРОВ – кандидат технических наук, доцент Высшей школы программной инженерии Института компьютерных наук и технологий. Сферы научных интересов: параллельная обработка данных, системы распределенного хранения данных, большие данные, верификация программного обеспечения, автоматизация тестирования.

Dmitry Fedorovich DROBINTSEV is a senior lecturer at the Higher School of Software Engineering at the Institute of Computer Science and Technology. Research interests: big data, information analytics, software verification.

Дмитрий Фёдорович ДРОБИНЦЕВ – старший преподаватель Высшей школы программной инженерии Института компьютерных наук и технологий. Сферы научных интересов: большие данные, аналитика информации, верификация программного обеспечения.