7th International Scientific Conference
Technics and Informatics in Education
Faculty of Technical Sciences, Čačak, Serbia, 25-27th May 2018
Session 3: Engineering Education and Practice
UDC: 004.5

Raspberry Pi module clustering and cluster application capabilities

Dragana Mitrović[1*], Dušan Marković[2], Siniša Ranđić[1]
[1]University of Kragujevac, Faculty of Technical Sciences in Čačak, Serbia
[2]University of Kragujevac, Faculty of Agriculture in Čačak, Serbia
[*] dragana.mitrovic.94@gmail.com

Abstracts: *The development of semiconductor technology and, consequently, the development of microcomputers created the conditions for adequate support for embedded parallelism in procedures for solving various problems in science and technology. With the simultaneous development of computer networks, conditions have been created for microcomputers to connect to clusters, which today are one of the pillars for the realization of parallel calculations. Raspberry Pi computer modules represent a good basis for cluster formation and support for parallel computing. The support of these computing practice in parallel computing is most often carried out with the support of the MPI (Message Passing Interface) concept.*

Keywords: *Parallel Computing; computer cluster; Raspberry Pi; message passing interface*

1. INTRODUCTION

Although parallel computing is often regarded as an objective in achieving more efficient problem solving, it can also be viewed from a completely opposite angle. This second angle relates to the real possibility that the built-in parallelism in the process of solving various problems is supported by computers. The ability to solve the problem, that is, calculations supported by parallel computer systems, has become real with the intensive development of semiconductor technology and its impact on the development of computing [1].

Particularly significant progress in parallel computing was achieved through computer clusters realized within the TCP/IP network [2]. In such systems, the parallel computation process takes place through the exchange of messages. The best known systems for parallel computing support within the TCP/IP cluster are MPI (Message Passing Interface) [3] and PVM (Parallel Virtual Machine) [4].

A large number of microcomputers, realized on the principle of one board computer, can be found on the market. Such computers, as a rule, have standard communication interfaces, which allow them to connect to the necessary peripheral devices, including connecting to various types of computer networks. One of the most famous modern microcomputer families are Raspberry Pi modules [5]. The latest generation of this module, Raspberry Pi 3 model B has the following features:

- CPU: Quad-core 64-bit ARM Cortex A53, 1.2GHz
- GPU: 400MHz VideoCore IV multimedia
- Memory: 1GB LPDDR2-900 MHz SDRAM
- USB: 4 ports
- Video outputs: HDMI, composite video (PAL and NTSC)
- Network: 10/100Mbps Ethernet and 802.11n Wireless LAN
- Peripherals: 17 GPIO, HAT ID Bus
- Bluetooth: 4.1
- Power source: 5V via MicroUSB or GPIO header.

Figure 1 shows the appearance of the Raspberry Pi module.

Figure 1. *Raspberry Pi module*

Due to processor and memory characteristics, as well as the ability to connect to the Ethernet network, Raspberry Pi is a good basis for the formation of the appropriate computer cluster. The aim of this paper is to demonstrate the possibility of realization of the computer cluster based on the Raspberry Pi 3 module connected in the TCP/IP network.

2. THE TERM OF COMPUTER CLUSTER

Computer clusters were developed in order to overcome limitation of computers available on the market with most cost efficient method. The term computer cluster can be defined in many ways, but essentially it represents multiple computers connected in a way that, from the "outside", they form a single unit. Inside computer cluster each computer is called a node, and one cluster can contain from two to few hundred nodes. Usually, nodes are commonly connected through fast local area networks (LANs) and they can form tight or loose connections depending on the way they are connected.

There are three main types of computer clusters: high performance (HP), load balancing, and high availability (HA) clusters. HP clusters are used for solving high and advanced computation problems, when result of one node effects future result of another node. Load balancing clusters are used when there is a need for more servers/nodes to host one website. In this case nodes serve multiple users with same resources at the same time. High availability clusters are used in cases when security of data stored on them needs to be active with minimum of down-time.

Some of major advantages of using computer clusters include better processing speed, flexibility, and high availability of resources. Multi-core processors have enabled the execution of the code to be parallelized by using the MPI concept. The question arises whether it is possible to achieve the same effect by using multiple computers connected to the LAN. Practice has shown that this can be achieved very successfully, as evidenced by numerous systems based on MPI network clusters [6]. One of the most famous systems is the Beowulf cluster [7, 8].

Figure 2 shows the basic scheme of a computer cluster in which computers are connected to a local computer network. Compute Node represents computers running parallel computing branches. Head Node represents the computer through which the cluster connects with the environment. In principle, this computer needs to have more powerful hardware than Compute Node computers. As a rule, the network that needs to provide the information exchange within the cluster should be based on Gigabit Ethernet. The cluster configuration in Figure 2 is sufficient to support lower-cost calculations. At the same time, this cluster can be a good basis for getting acquainted with the principles of parallel computing using the MPI concept. Accordingly, the appropriate cluster structure can be realized using the Raspberry Pi module.

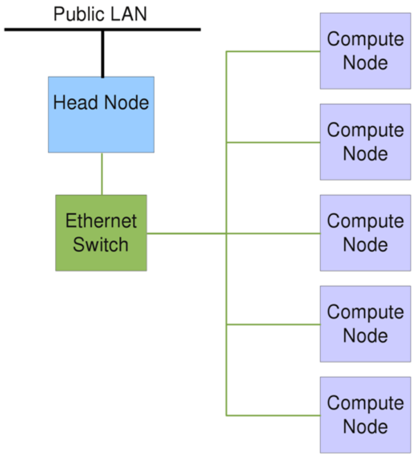

Figure 2. *Typical structure of MPI cluster [9]*

This paper deals with the possibilities of using clusters realized using four Raspberry Pi modules. In a given cluster, one Raspberry Pi module has the Head Node function, and the other three are the Compute Nodes.

3. POSSIBILITIES OF NETWORKING THE RASPBERRY PI MODULE

Raspberry Pi is very powerful but small computer module, created by developer team in England for educational purposes. Latest version of this module is Raspberry Pi 3 Model B+ with 1.4GHz 64-bit quad-core processor and dual-band wireless LAN, Bluetooth 4.2, faster Ethernet and Power-over-Ethernet support.

Possibilities of using Raspberry Pi module are many. It can be used as a computer, media server, gaming console, IoT device, and much more. It is possible to install many OS on the Raspberry Pi module, so configuration is mostly easily done through terminal and additional libraries. For those that do not want to use external hardware like monitor, keyboard or mouse, can just insert SD cards with pre-installed OS and use Bluetooth to operate Raspberry Pi module. Most recent version of Raspbian OS has all software needed for setting up and running Bluetooth connection.

Using Bluetooth connection, Raspberry Pi can control external hardware, like Bluetooth speakers, actuators, or even make connection with mobile phones. As for Wi-Fi module, it is usually used for connecting Raspberry Pi to other devices and actuators over Wi-Fi network. With this modules it is possible to build Wi-Fi router, 3D printer, make smart home devices or send collected data to online server. Another type of connection is through Ethernet. Raspberry Pi modules inside clusters are usually connected with Ethernet cables to secure data and get fastest exchange of data [10].

4. RASPBERRY PI CLUSTER BASED ON MPI CONCEPT

MPI or Message Passing Interface is communication protocol for programming parallel computers. Originally, MPI was designed for distributed memory architectures, but now it can run on almost any hardware platform, distributed memory, shared memory, hybrid, and so on. Some advantages of using MPI are: support for full asynchronous communication, processes can be grouped based on context, it is flexible and portable.

Hardware requirements for building one Raspberry Pi cluster are: four Raspberry Pi 3 modules, four 16GB micro SD cards, four USB to Micro USB cables, port switch, router, and Ethernet cable. For building cluster there has to be at least two or three nodes (Raspberry Pi) and if it is not enough it is possible to add more later on. To start on configuring the cluster, first the Raspbian Jessie image is downloaded to each micro SD card. Raspbian Jessie is operating system based on Debian Jessie, specifically designed for Raspberry Pi modules. After booting Raspbian on each module, take note of their IP addresses to be able to generate SSH keys. SSH stand for secure shell and it is used as encrypted remote login protocol and a way to communicate with other nodes on the same network. SSH can be configured over Wi-Fi and once configured you can use SCP (Secure Copy) and SFTP (Secure File Transfer Protocol) for transferring files and directives directly from one node to another. Via SSH it is also possible to directly run commands on selected node, change host name of nodes or even shutdown a node. At this point it is good to have another SD card with more memory to service as a disk for cluster. Depending on the type of cluster you are making, this additional memory can be available to all nodes or just the Head Node.

After configuring the operating system, it is necessary to install MPI software on the SD card of each of the nodes in the cluster. In this case, the MPICH3 version is installed. Also, you need to install the MPI4PY library, which allows programming nodes in the Python programming language.

In order to use identical system software on all nodes, it is necessary to perform cloning of the formed SD card. Finally, each node needs to be configured by assigning a unique name (HostName) and enabling it to work with the SSH protocol. Also, it is important that each node has its own IP address, which has to be stored in the so-called machine file. This file contains the IP addresses of all the nodes on which processes are started. This file must be located at each node and uses MPICH3 for communication and sending/receiving messages between the nodes.

Finally, it is necessary to generate SSH keys to allow management of each Raspberry Pi module without using the username and password. Figure 3 shows a schematic representation of the MPI cluster structure with hostname, IP addresses, and SSH keys used.

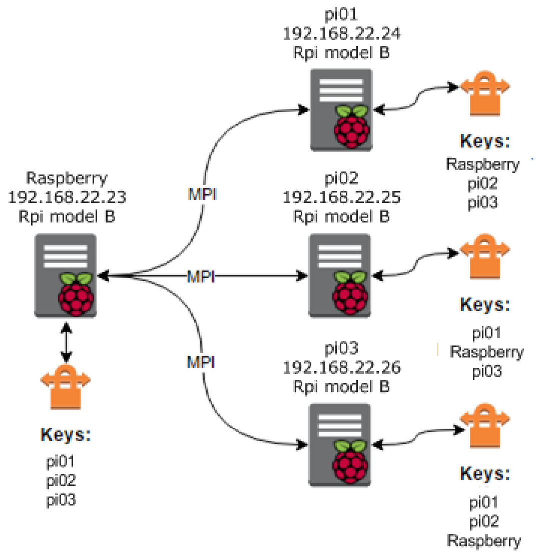

Figure 3. *Example of hostname and SSH keys in the MPI cluster based on the Raspberry Pi 3 module [11]*

5. POSSIBILITIES OF USING RASPBERRY PI MPI CLUSTER

When one program requires running multiple tasks that can be run separately it is good idea to run it on cluster. This shortens execution time of programs, because each node inside cluster can run one task, creating parallel computing. Not only this, but using clusters can improve redundancy and fault tolerance. When one node or server crashes inside cluster, process is distributed on rest of available nodes. Knowing this, there are many possibilities using Raspberry Pi cluster. Some possible uses are: machine learning, 3D processing, some types of simulations, automated software testing, media encoding, network load distribution, and much more.

In practice, there are examples of the use of Raspberry Pi-based clusters for the development of parallel applications. At the Cornel University, a project was realized to develop Symmetric Multi – Processing Platform (SMP) and Asymmetric Multi – Processing (ASMP) parallel applications [15]. The goal of realization of this project is to obtain support for Lab projects, within the course "Design with Embedded Operating Systems". The project was realized using the Raspberry Pi 2 Model B computer. With four cores that share physical memory this computer represents a good platform for Symmetric Multi-Processing. For

Asymmetric Multi-Processing, a cluster was developed in which four Raspberry Pi modules are connected via an Ethernet router. As an application for analyzing the performance of the realized cluster, Sobel Filter Edge Detection was selected. The Sobel filter is based on calculations over two-dimensional series, which is an excellent basis for the development of a parallel program. The results of the analysis showed that using Symmetric Multi - processing and Asymmetric Multi – Processing software achieve four to ten times better performance than with sequential programming.

OpenMP (Open Multi Processing) technology was used to develop parallel applications within the Symmetric Multi-Processing approach. This technology is adapted to multiprocessing based on shared memory. The OpenMP concept is based on compiler directives and libraries that define the framework for parallel programming [16]. Through OpenMP, the user receives a simple interface for the development of parallel applications. The platforms on which these applications can be used range from standard personal computers to supercomputers. This technology realizes multithreading, i.e. a parallelization method to which the main thread, which is a series of instructions, executes consecutively in a certain number of threads, thus achieving the parallelization of tasks, as shown in Figure 4.

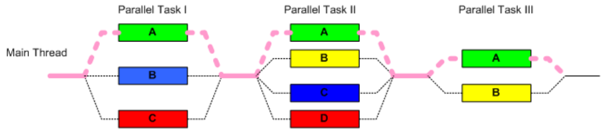

Figure 4. *Multithreading using openMP*

In the case of Asymmetric Multi-Processing, MPI technology is used. It is a standardized messaging system that defines the syntax and semantics of library routines that use a wide range of users to write programs in C, C++, and FORTRAN programming languages. Accordingly, MPI can be considered as a communication protocol for parallel programming. In the concrete project, the MPICH option was used, because it is widely accepted in relation to other realizations of the MPI concept. It should be noted that the MPI4PY achieves the MPI concept for programmers in the PYTHON programming language. Otherwise, within this project, C programming language was used for realization of both types of multiprocessing (SMP and ASMP).

At the Midwest Instruction and Computing Symposium 2017 (MICS 2017), researchers from the University of Wisconsin – La Crosse presented a cluster based on eight Raspberry Pi 3 Model B modules [17]. As in the previous case, the Raspbian Jessie based on Debian Linux was used as an operating system. Multiprocessing is realized using MPI standards and applications written in the Python programming language. The realized cluster was tested using the Monte Carlo method for calculating the value of Pi. Calculation was performed on eight nodes with one to four processes per node.

During the testing of the MPI application on the Raspberry Pi cluster, it was determined that only 25% of the computational power on each Raspberry Pi module was used. This is due to the use of only one core within the Raspberry Pi module. Consequently, the idea was to include, in addition to the MPI approach, the local multiprocessing at the level of each Raspberry Pi module within the cluster in the calculation. Since the MPI variant used the Python programming language, local multiprocessing was implemented through standard multiprocessing libraries of this programming language. Thanks to this, Raspberry Pi node computing resources are fully used, i.e. each of its four cores. Table 1 shows the mean times of calculating Pi using Monte Carlo approximation in 500 points.

Table 1. *Mean computing time within clusters with MPI and multiprocessing.*

Cluster MPI		Cluster Multiprocess	
Processes	**Time [s]**	**Processes**	**Time [s]**
1	78,61	1	98,41
2	47,35	2	49,56
3	32,01	3	33,07
4	24,39	4	24,85

This research applied the multiprocessing concept instead of multithreading to avoid the limitation due to the Global Interpreter Lock (GIL). Because of the GIL Python interpreter, it does not allow two instructions in the Python program to run at the same time, causing the threading concept to not increase the performance. As can be seen from Table 1, the mean calculation times are approximately the same. Improvements of approximately 35% are achieved by increasing the level of multiprocessing. Researchers assume that similar times are calculated for the consequences of a small number of messages that are exchanged between nodes.

Cluster building experiences, based on the Raspberry Pi module, have initiated the implementation of similar clusters with a large number of nodes. Examples of such clusters are "Beast v1", with 120 and "Beast v2" with 144 Raspberry Pi modules. These clusters are primarily designed to test software produced by reisin.io, designed to support IoT and embedded devices.

A group of professors and students from the University of Southampton created "Iridis – Pi" Cluster [18]. The cluster consists of 64 Raspberry Pi nodes that are interconnected over 100 Mbit Ethernet network. The cluster was created as a result of an attempt to provide a low – cost starting platform that would enable students to get to know them and can apply high-performance computing and data processing in a range of engineering and scientific problems.

Maybe biggest Raspberry Pi cluster made so far is 256 node “SeeMore” cluster. This project is developed by Kirk Cameron, professor of computer science and Sam Blanchard, an assistant professor of sculpture at Virginia Tech University. Idea was to create something to which will bring art and engineering together, while showcasing significance of parallel computing to viewers. Blanchard's idea was to create a kinetic sculpture that will show what's going on inside the computer.

Figure 5. *CSLab Raspberry Pi cluster*

Having in mind the aforementioned experiences in creating a cluster based on the Raspberry Pi module, in the Laboratory for Computer Science, the Faculty of Technical Sciences in Čačak has started to create its own computer cluster. Primarily, the goal was for students of master studies to be familiar with the problems of parallel computing, computer clustering and the Message Passing Interface concept. Also, they also wanted to gain experience in working with Raspberry Pi modules, installing the appropriate operating system and programming them, primarily using Python programming language.

The cluster was created using four Raspberry Pi 3 B modules, which are connected to a local computer network using a 5-port Ethernet router (Figure 5). The power supply of the Raspberry Pi 3 B module was achieved using the 5-port Power USB Hub. Raspberry OS has been installed on each Raspberry Pi module and the desired network parameters (IP addresses) have been set. Also, MPICH3 is installed on each module, allowing clustering based on the MPI concept.

Checking the correctness of the work of the created Raspberry Pi cluster was performed by executing a simple test program written in the Python programming language. This test program shows that some processes are performed on different computers within a cluster. This demonstrated that the created system is available for performing an arbitrary parallel application.

In the continuation of the research it is planned to develop various parallel applications and compare their performance with respect to the sequential execution of programs on one computer, or in the multithreading environment on computers with multi-core processors. It is also planned to expand the existing cluster by adding new Raspberry Pi 3 B modules.

6. CONCLUSION

The formation of MPI clusters has become particularly interesting with the emergence of computer modules such as Raspberry Pi or BeagleBone [14]. Good process and network characteristics of these modules allow clusters with a number of nodes to be formed based on them. This paper presents the realization of the MPI cluster from four nodes based on the Raspberry Pi 3 module.

Through the conducted research, within which the cluster shown was formed, the aspects of using the Raspberry Pi 3 module for this purpose, the installations of the required software, and the method and requirements for configuring nodes within the cluster were examined. Also, experience has been gained in terms of basic knowledge in programming such a computing complexity using the MPI concept. The realized research has shown that the construction of this cluster helped students of master studies in acquiring additional knowledge in the field of parallel processing. Consequently, the developed cluster, as well as future research on this plan, will provide a good basis for the education of students in the field of parallel processing.

It is planned that as part of further research, a cluster with a large number of Raspberry Pi nodes to be formed. It is also planned to use the programming of a more complex algorithm to use such a cluster and MPI concept and to conduct an appropriate comparative analysis in relation to the use of the sequential program.

Also, it will be interesting for further research to compare performance times and other performance factors for different platforms in which software applications for the same algorithms would be executed. Further research implies the use of different programming languages for the development of parallel applications and the implementation of an appropriate comparative analysis of the results obtained.

ACKNOWLEDGEMENTS

The paper presents the results of the research within the Hardware – Software CoDesign course at the Master studies in the field of Computer Engineering at the University of Kragujevac, Faculty of Technical Sciences in Čačak. Material support to the research was realized through the TR32043 project funded by the Ministry of Education, Science, and Technological Development of the Republic of Serbia.

REFERENCES

[1] Czech, Z. J., "Introduction to Parallel Computing", 1st Edition, Cambridge University Press, 2017

[2] Yeo, C. S., Buyyua, R., Pourreza, H., Eskicioglu, R., Graham, P., Sommers, F., "Cluster Computing: High – Performance, High – Availability and High – Throughput Processing on a Network of Computers", In Zomaya, A. Y. (eds) Handbook of Nature – Inspired and Innovative Computing, Springer, Boston, MA, 2006, pp 521 – 551

[3] Levesque, J., Wagenbreth, G., „High Performance Computing: Programming and Applications‟, 1st edition, Chapman and Hall/CRC, 2010

[4] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., Sunderam, V. S., „PVM: Parallel Virtual Machine: A User's Guide and Tutorial for Network Parallel Computing‟, The MIT Press, 1994

[5] Robinson, A., Cook, M., „Raspberry Pi Projects‟, 1st Edition, Wiley, 2013

[6] Gropp, W., Lusk, E, Skjellum, A., "Using MPI: Portable Parallel Programming with the Message – Passing – Interface", 3rd Edition, The MIT Press, 2014

[7] Sterling, T., "Beowulf Cluster Computing with Linux", 1st Edition, The MIT Press, 2001

[8] Sterling, T., "Beowulf Cluster Computing with Windows", The MIT Press, 2001

[9] „Qlustar Cluster OS 9.1 – Administration Manual‟, Qlustar Documentation Team, Q – Leap Networks GmbH, 2017

[10] Golden, R., "Raspberry Pi Networking Cookbook", Packt Publishing, 2013

[11] Iyer, K., "Learn to Build your Own Supercomputer with Raspberry Pi 3 Cluster", Post on TechWorm, 2018

[12] Dennis, A. K., "Raspberry Pi Super Cluster", Packt Publishing, 2013

[13] Morrison, C. R., "Build Supercomputers with Raspberry Pi 3", Packt Publishing – ebooks Account, 2017

[14] Molloy, D., "Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux", 1st Edition, Wiley, 2014

[15] Govindaraj, V., "Parallel Programming in Sapberry Pi Cluster", A Design Project Report, Schole of Electrical and Computer Engineering, Cornel University, 2016

[16] Kiessling, A., "An Introduction to Parallel Programming with OpenMP", The University of Edinburgh, A pedagogical Seminar, 2009

[17] Dorr, G. et al, "Introduction to Parallel Processing with Eight Node Raspberry Pi Cluster", Midwest Instruction and Computing Symposium (MICS), The University of Wisconsin – La Crosse in La Crosse, 7 – 8 April 2017

[18] Cox, S. J. et al "Iridis-pi: a low-cost, compact demonstration cluster", Cluster Computing, 2013