APPLICATION OF IMAGE PROCESSING IN FOG COMPUTING

Dušan Marković[1], Dejan Vujičić[2], Borislav Đorđević[3], Siniša Ranđić[2]
[1] Faculty of Agronomy Čačak, University of Kragujevac, Serbia
[2] Faculty of Technical Sciences Čačak, University of Kragujevac, Serbia
[3] Institute Mihailo Pupin d.o.o., Belgrade, Serbia

Abstract
The Internet of Things devices incorporate sensors for sensing and sending data, and Cloud infrastructure, for data analysis. Instead of using Cloud infrastructure, Fog computing could be also used as it brings the data analysis closer to the sensing area. In this paper, a model of the system based on Raspberry Pi is serving as a central node that is acquiring images from local sensor devices. Testing examples using images of lemon are transferred to Fog node where image segmentation was used to determine whether the lemon is damaged, by analyzing predefined range color. Fog node was loaded with various numbers of images and execution time for observed image processing from the selected sensor node estimated. The resulting delay when processing the image in this way could be used as an indicator for selecting the Fog node for particular data processing.

Keywords: Fog computing, Image processing, Internet of Things, Raspberry Pi.

INTRODUCTION

Information about living or working environment can make a significant contribution to everyday human activities, so their execution would be more efficient and conducted using a more effective way. Usually, information about conditions and status in the surrounding area could be obtained from data captured by appropriate sensors. Devices equipped with sensors could be interconnected and transfer data to the point where users could access them, save and analyze.

Wireless connection of sensor devices has led to the great application of wireless sensor networks in data collection. In the meanwhile, a new paradigm emerged and that was the Internet of Things (IoT). A concept that enables devices with adequate communication support to connect to the Internet and to exchange data between them without human intervention. IoT devices could transmit data across the Internet to the remote servers or in the spirit of new technologies to the Cloud platforms. Received data on Cloud platform could be processed to infer new information or knowledge or could be stored for later analysis.

The more sensor devices are connected to the network the more data are transferred to the cloud platforms which causes large amounts of data to be accepted and processed. The reduction of accepted data could be accomplished by the distribution of processing from Cloud to devices that are closer to the sensor's places. This relocation of data processing from Cloud to the devices in the lower level is known as Fog computing.

Data can be gathering from sensor devices in the area to the central computer-based devices that represent the Fog computing concept. Data would be analyzed on this level and there is no need for sending them to the Cloud, but the only result of the processing could be sent for further activities. Also, the results of processing are obtained close to the monitoring place and could be presented to users without much delay.

One of the very useful applications of sensor data processing is image analysis, where the current state could be determined by analysis of images from the selected locality. Image analysis in the concept of IoT could be found in articles such as an integrated fire detection system [1], intelligent parking system [2], or smart traffic density control [3]. Also, there are interesting applications of image processing in agricultural products such as classifiers of tomatoes [4], smart apple sunburn sensing system [5], or plant health monitoring system [6].

This paper aims to present a model of the Fog computing system which has the role of processing data collected from sensor devices and send results to the Cloud and nearby users. The presented example represents the capability of the central node device to process a set of images that could be obtained from camera-equipped devices and be used under the Fog concept. The model was tested by transferring images to the central Fog device to show processing characteristics by image analysis under loads from another node.

MATERIALS AND METHODS

The model of the Fog device is based on a Raspberry Pi 3 module which has a wireless connection with sensor devices in the surrounding area. These devices represent nodes with appropriate sensors and camera capturing state that was in interest by users. In this case, the image is transferred from sensor devices to the Fog node where it can be transmitted further to the Cloud or processed in the site.

Raspberry Pi (RPi) represents a computer that is characterized by small dimensions, relatively high processing power, low price, and all that on one board which was emerged on market in 2012. Also, the RPi module has a variety of interfaces which gave wide possibility in interaction with other devices. Adding necessary input and output peripheral units makes RPi a personal computer for users, or with connection to the Internet remote access could be established, which makes the RPi a standalone node device. RPi uses an SD card as a memory disk and operating system known as Raspberry PI OS or Raspbian as one of the customized Linux operating systems [7].

Raspberry Pi 3 represents the third generation of Raspberry Pi computers realized fully on one chip. It consists of a 64-bit quad-core microprocessor, 1 GB RAM, wireless communication by Wi-Fi and Bluetooth 4.1, HDMI interface, four USB ports, 40 general-purpose input/output pins, camera interface, and composite audio/video input [8].

Sensor nodes in a Fog node environment could be based on microcontrollers that have been connected with the camera module and send an image to the Raspberry Pi. Microcontrollers do not need to have enough processing power for image analysis; its main function would be to take a picture with a camera and send the captured image to the central Fog node. The model of the overall system is presented in Fig. 1 where data could be transferred between sensor nodes and Fog node. Also, Fog node could transmit data to the Cloud over the Internet and receive data and processing configuration from Cloud. It has a central role, it can take over data processing from the Cloud, run image analysis and the result of processing obtain to users.

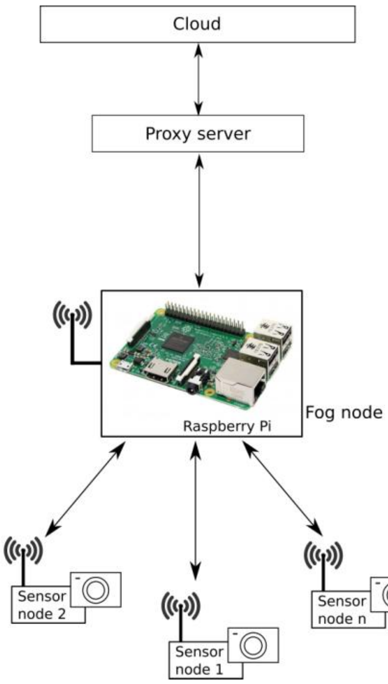

Fig. 1. *Model of the system with Fog node between sensor nodes and Cloud*

For testing purposes of the Fog node with image processing, the dataset from the Kaggle website was used. The dataset contains images of lemons prepared for the conducted process of fruit quality control by analyzing images [9]. A python script was implemented on Raspberry Pi to analyze images of lemon and determine pixels that belong to a range of colors marked as a bad part of the fruit. Firstly, the arrival image color model was changed from BGR to HSV. The reason for this conversion is unequivocally determination of color range in the HSV color model. The next step was the segmentation of HSV image by color range, using also Python library for image processing. In this process parts of the lemon image are distinguished by a predefined range of colors. These ranges mark and count pixels on the image that belong to the bad section and have the meaning of rotten or damaged part of the fruit.

An example of the testing purpose of data processing on Fog node could be presented in Fig. 2 where one node was observed and its data exchanged. Example aimed to show the amount of execution time for transmitted images towards Raspberry Pi, where time for processing and replaying results under various data loads on the Fog node is also considered.

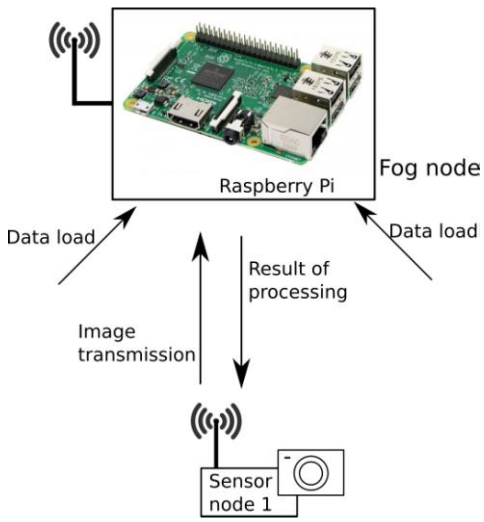

Fig. 2. *Data transfer between the sensor node and Fog node*

Data exchange is performed by the MQTT protocol intended for IoT devices and a comparison between HTTP and MQTT could be viewed in [10]. Data load was emulated by sending a predefined number of images per second. Under this condition observed node 1 calculates the difference between the time of sending the image and the time when the result of processing arrived from the Fog node.

RESULTS AND DISCUSSION

The central node, in addition to its basic role, was used as a Fog node on which image processing was performed. One of the results of image processing is shown in Fig. 3. The first image represents lemon with a rotten part that can be seen as a gray and dark green segment.

On the second image, every pixel that belongs to the rotten segment was count and replaced by a black pixel, while good yellow parts were replaced by white pixel. Using image segmentation in Python script, damaged or rotten lemon could be detected and their size calculated in pixel apart from black pixels in the background. Depending on the calculated number, the degree of lemon damage can also be determined.

This image processing was applied as a testing example to estimate usage of the central node or gateway based on Raspberry Pi for the role of the Fog node.

a) b)

Fig. 3. *Image of the damaged lemon (a) and image after segmentation (b)*

The focus of the test was on calculating the period between the time of sending an image from node 1 for analysis to the Fog node and the time when the result of segmentation was returned. The resolution of images used for testing was 640 x 480 pixels, and the same size of images is used for acquiring on Fog node.

The frequency of sending images from another node, which represent data load to the Fog node, was shown in Table 1, and also image processing delays. In the first row of the table, there are defined time intervals between sending data from the *n*-th sensor node. Value in the first cell means that one second has passed after the previous code execution that performed image transfer. The second cell represents a 0.5 second pass period between transfers and approximately it means that two images are processed per second on the Fog node. The second row in the table possesses values of time delays in seconds only for images transfers from sensor node 1. The third row shows values of time delay that contain periods of transfer image, image analysis, and receiving results.

Table 1. *Intervals of image sending and delays during processing*

Sending period (s)	1	0.5	0.1	0.05	0.02	0.01
Transfer image delay (s)	0.118	0.130	0.139	0.161	0.267	0.394
Image processing delay (s)	0.197	0.225	0.235	0.26	0.472	0.581

Values form table are also presented in Fig. 4 where the time of execution from node 1 to Fog node could be compared for various data loads from other nodes in the Wi-Fi network.

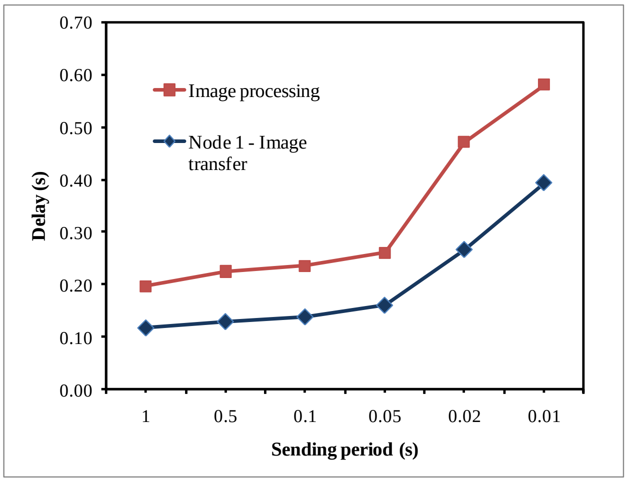

Fig. 4. *Node 1 image processing time on Fog node*

During the execution of the example when the rate of sending images to the Fog node has a higher value, which corresponds to the last two points on the chart, Raspberry Pi is restarted after some time. However, observing values for 0.1s or 0.05s intervals, processing delay from the sensor node side is 0.26 seconds. This can be a favorable value for the delay in the case of image processing on the Fog node. Timings of image transfer are also shown on the chart which could be considered in other processing cases that require more processing power and time on the side of the Fog node. In this case, the central Fog node which has some predefined purpose could also be used for moving some of the data processing from the Cloud.

CONCLUSION

In this paper model of the Fog computing system was presented with the reallocation of data processing from the Cloud to a central node near sensor devices. Testing example with image processing on Fog node applied on received images from surrounding sensor devices was presented without forwarding images to Cloud. Fog node could be used for image processing with a load that makes a sequence of images sending at every 0.1 to 0.05 seconds. Further research will be in performing test examples on the Fog node, involving acquired sensor data, which requires more complex processing.

ACKNOWLEDGEMENT

The work presented in this paper was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia, and these results are parts of the Grant with University of Kragujevac - Faculty of Agronomy Čačak and University of Kragujevac - Faculty of Technical Sciences Čačak, Grant No. 451-03-68/2020-14/200132.

REFERENCE

[1] Sharma A., Singh P. K., Kumar Y. An integrated fire detection system using IoT and image processing technique for smart cities. Sustainable Cities and Society 2020; 61: 102332.

[2] Rane S., Dubey A., Parida T. Design of IoT based intelligent parking system using image processing algorithms. 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, 2017, p. 1049-1053.

[3] Frank A., Khamis Al Aamri Y. S., Zayegh A. IoT based Smart Traffic density Control using Image Processing. 2019 4th MEC International Conference on Big Data and Smart City (ICBDSC), Muscat, Oman, 2019, p. 1-4.

[4] Clement, J., Novas, N., Gazquez, J.A., Manzano-Agugliaro, F. High speed intelligent classifier of tomatoes by colour, size and weight. Spanish Journal of Agricultural Research 2012; 10(2): 314-325.

[5] Shi G., Ranjan R., Khot L. R. Robust image processing algorithm for computational resource limited smart apple sunburn sensing system. Information Processing in Agriculture 2020; 7(2): 212-222.

[6] Pavel M. I., Kamruzzaman S. M., Hasan S. S., Sabuj S. R. An IoT Based Plant Health Monitoring System Implementing Image Processing. 2019 IEEE 4th International Conference on Computer and

Communication Systems (ICCCS), Singapore, 2019, p. 299-303.

[7] Vujović V., Maksimović M. Raspberry Pi as a Wireless Sensor node: Performances and constraints. 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2014, p. 1013-1018.

[8] Pagnutti A. M., Ryan E. R., Cazenavette V J. G., Gold J. M., Harlan R., Leggett E., Pagnutti F. J. Laying the foundation to use Raspberry Pi 3 V2 camera module imagery for scientific and engineering purposes. Journal of Electronic Imaging 2017; 26(1): 013014.

[9] Lemons quality control dataset. Available: https://www.kaggle.com/maciejadamiak/lemons-quality-control-dataset, 2020 (Accessed on: September 23, 2020).

[10] Yokotani T., Sasaki Y. Comparison with HTTP and MQTT on required network resources for IoT. 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, 2016, p. 1-6.