Use of the BOINC system for distributed data collection in the "Internet of Things"

Marián Handrik[*,1], *Jana* Handriková[2], *Lenka* Jakubovičová[1]

[1]University of Žilina, Faculty of mechanical engineering, 010 26 Žilina, Slovak Republic
[2]Armed forces academy of general M. R. Štefánik, Department of informatics, 031 06 Liptovský Mikuláš, Slovak Republic

Abstract. The BOINC software package allows the creation of massive computational systems for distributed computing. The system is designed to use heterogeneous computing devices that are differing in hardware architecture and operating systems. The variability and versatility of the BOINC system allows its use to create a distributed system for measurement and data collection. This is done through the standard HTTP and HTTPS web communication protocol that is used on the Internet in connected devices, so called "Internet of things".

Keywords: data acquisition, BOINC, internet, http, https, image recognition

1 Introduction

In the field of distributed computing systems, BOINC is used to manage the computations in heterogeneous distributed computing systems. This system can also be used for ARM processor architecture. ARM processors are used, for example, in tablets, mobile phones, or single-rack minicomputer of Raspberry PI.

The single-board Raspberry PI computer is the credit-card sized computer. Raspberry PI 3 model B has a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, 1 GB RAM, with the connector for 6 megapixel camera. The graphic output is possible via HDMI and composite video. Internet connection is possible via 100 Mbit Ethernet, WIFI and Bluetooth. Raspberry PI 3 model B has 4 x USB to connect other devices, such as keyboard, mouse, memory media USB keys - disks, cameras, measuring devices. Thanks to USB, this device can be also connected to the 4G mobile networks. Raspberry PI 3 Model B includes even 40pin General-purpose input/output (GPIO) that is a generic pin on an integrated circuit or computer board whose behaviour-including whether it is an input or output pin—is controllable by the user at run time. The GPIO interface allows the connection of specific measuring devices designed for the purpose of certain measurement.

The official Raspberry PI operating system is based on the Linux Debian operating system, containing almost all packages of this system. The price of Raspberry PI 3 B model, including the camera, does not exceed 70€, therefore it is used for bulk data collection or measurement. Raspberry PI Model B is shown in Figure 1.

[*] Corresponding author: marian.handrik@fstroj.uniza.sk

Reviewers: *Robert* Grega, *Milan* Naď

The BOINC system enables simple system management for data collection via Raspberry PI. In the following chapter, the methods of gathering image data are described by image recognition methods in the field of machine vision through the BOINC system [1-4].

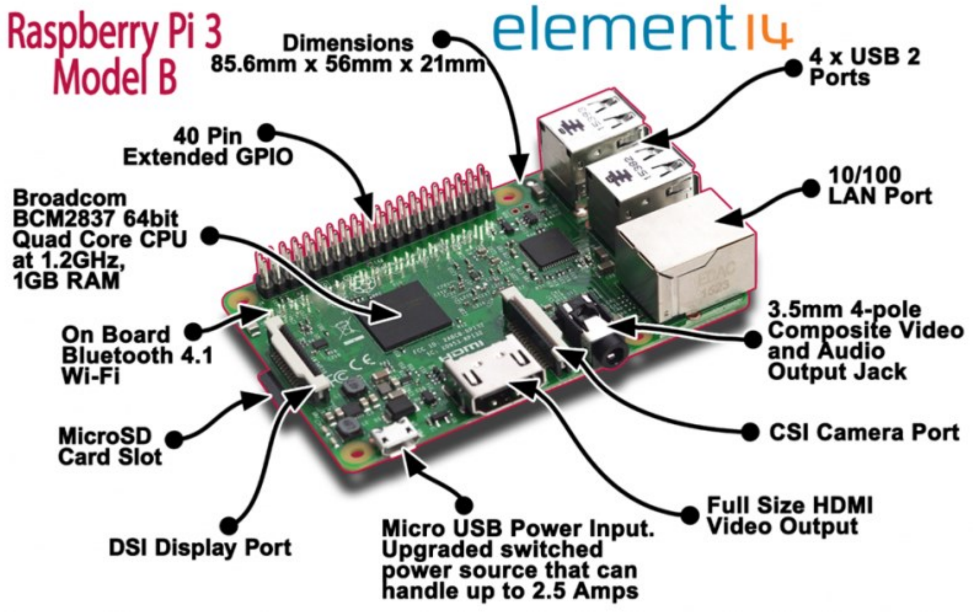
Raspberry Pi 3 Model B
element14
Dimensions 85.6mm x 56mm x 21mm
4 x USB 2 Ports
40 Pin Extended GPIO
10/100 LAN Port
Broadcom BCM2837 64bit Quad Core CPU at 1.2GHz, 1GB RAM
3.5mm 4-pole Composite Video and Audio Output Jack
On Board Bluetooth 4.1 Wi-Fi
CSI Camera Port
MicroSD Card Slot
Full Size HDMI Video Output
DSI Display Port
Micro USB Power Input. Upgraded switched power source that can handle up to 2.5 Amps

Fig. 1. Raspberry PI model B

2 Image recognition and computer vision

Image recognition systems can be designed for various specific tasks:

- For 3D reconstruction, the crucial step is to determine the relationships between the corresponding points across the image reliably. With the help of these relationships, it is possible to reconstruct the frames of cameras that recorded images as well as 3D spatial points corresponding to these points based on multi-view geometry and robust object detection methods.
- Object recognition can be divided into a specific object recognition and general object recognition of a certain type. A brief overview of object recognition methods is given in [5, 6]. The objective of category recognition is to classify an object into one of several predefined categories. The main task is to recognize the object from different angles, scales, lighting conditions or against cluttered backgrounds, etc. The aim of the object classification is to specify the object category or the group of objects to which the object belongs.
- The aim of the systems that choose the object based on the content is to return a number of most liked objects from a large database. Therefore, there are a huge number of local descriptors describing these objects as there are a huge number of objects. Special index structures, such as k-d trees, are used to search in these systems.
- The aim of simultaneous localization and mapping is to reconstruct the surrounding environment and locate the object within it in real time. These systems are based on various sensors and captured image by cameras. They are used in the automated navigation of vehicles and aircrafts, when creating extended reality.

The recognition process consists of two basic steps - the object detection and object or category recognition. The object is described (represented) by a descriptor, certain values that specifically describe the surroundings of a particular point in the image for later recognition. Descriptors can be used to find the match between objects in a database and an input image (recognizing a predetermined object), or serve to achieve a robust image representation that is the model of the given category (determining the group of objects to which the object belongs).

The procedure of recognizing a particular object can be demonstrated on pedestrian or vehicle recognition. When detecting the object from video sequence, the recording quality is as important as software for subsequent processing. In general, pedestrian and vehicle detection according to [6, 7] follow these steps:

- detection of a candidate region on which the object is likely to be,
- description of this region by using a descriptor,
- classification of this region, or verification whether a given candidate region contains the object of interest or not,
- post-processing.

The detection of candidate regions can be done, for example with the help recognizing patterns in the frame of a video sequence. In the case of a video sequence, the background subtraction method can be used - moving objects are separated from the background by calculating the difference between the current frame and the reference background.

The descriptor consists of features organized into the structure. It is expected that the descriptor is able to describe the object from different camera viewpoints and in different poses. Features are calculated for individual pixels (points of interest, key points) or for local image regions. Both types of descriptors can describe an object, in this case a pedestrian or a vehicle, as a whole or as a set of its individual parts (the parts do not have to correspond with the individual anatomical parts of the human body or vehicle components). Descriptors describing the parts of an object can be created independently in different processes.

The most common way how to create a descriptor is to link the local features into a multidimensional vector. The features used to create a descriptor can describe the shape of the object (e.g. edge detectors), object appearance (e.g. Local Binary Feature and its modification), object movement (histogram of optical flows).

The combination of multiple detection methods can increase system performance, improve its ability to detect an object. For example, in [8], the accuracy of vehicle detection at night is increased by the fusion of different features.

After gaining a descriptor from the candidate regions, the detention whether the candidate area contains an object or not follows.

Classification can use a generative or discriminative approach. Generative models of appearance are trying to provide us with as precise description as possible, they detect the data from the object category by creating a pattern - a template, with which the candidate regions are compared and if there is sufficient similarity to the pattern, they are classified as an object of a certain category. Discriminative models detect the object with the help of binary classification and robust supervised methods and algorithms, such as Support Vector Machine or cascaded AdaBoost classifier.

The classifier training process can be run incrementally - after processing each of the following tasks, the classifier is modified and is capable of classifying the object, and this corresponds to online learning tasks. If the classifier training process takes place over a set of training tasks before classifying an unknown object, it is a non-incremental learning that corresponds to offline learning tasks.

The training process can be run on a set of training tasks, which belong to a predetermined class (learning with a teacher). The goal is to detect the rule based on which an unknown object belongs to a particular class. In case of learning without a teacher, training tasks do not include information about certain classification. The goal is to divide objects with a specified combination of features into groups so that the similarity of combinations of object features assigned to one group was as large as possible, and vice versa, so that the difference between combinations of object features from different groups was greatest.

Image and video processing, which we can consider as the time sequence of individual image frames puts high demands on memory and the computing resources used. Therefore, it would be advisable to consider using parallel processing options, either using cloud or distributed systems.

Image processing methods, machine learning as well as other algorithms can be found in OpenCV free software package. Vehicle license plate recognition is available, for example, in free OpenALPR software package. There are also many other available image and video editing software packages on the Internet.

3 BOINC system

The BOINC (Berkeley Open Infrastructure for Network Computing) software platform is used for volunteer computing or grid computing creation.

BOINC is designed to support applications that have large computation requirements, large storage requirements, or both. The main requirement of the application is that it will be divisible into a large number (thousands or millions) of jobs that can be done independently. This task group also includes data collection by analyzing the image from the camera. BOINC is free software distributed under the Lesser General Public License (LGPL) version 3 or higher.

The main purpose of BOINC development is to support the public resource computing paradigm, to support the creation of many projects, to encourage a large community of computer owners to participate in one or more projects [9].

Other BOINC development goals are [9]:

Reduce the barriers of entry to public-resource computing. BOINC allows the scientists with moderate computer skills to create and operate a large public-resource computing project with about a week of initial work and an hour of maintenance per week. The server for a BOINC based project can consist of a single machine configured with common open-source software (Linux, Apache, PHP, MySQL, Python).

Share resources among autonomous projects. BOINC-based projects are autonomous; projects are not centrally authorized or registered. Each project operates its own servers and stands completely on its own. PC owners can easily participate in multiple projects and can determine how the resources will be divided among these projects. If most users are involved in multiple projects, then overall resource utilization is improved: if one project is closed for repairs, then other projects temporarily inherit its computing power. The CPU might work for one project while the network is transferring files for another.

Support diverse applications. BOINC accommodates a wide range of applications; providing flexible and scalable mechanism for distributing data and its scheduling algorithm intelligently connects, matches requirements with available resources. Existing applications in common, widely used languages can run as BOINC applications with little or no modifications. An application can consist of several files. New versions of applications can be deployed without the involvement of volunteers, centrally from BOINC servers.

Reward participants. Projects, which encourage the participation of volunteers, must provide them with some form of reward. The reward is provided on the numeric measure of the volunteer's contribution to the overall computation of the task. BOINC provides a credit accounting system that reflects the usage of the volunteer's computing resources in all projects. BOINC also makes it easy to add visualization graphics to volunteer applications that can be used as screensaver graphics. For example, the PlanetQest project allows a volunteer to name a planet found with the help of his or her computer.

The basic features of BOINC [10]:

- Project autonomy. Projects are independent; each one realizes computation and operates its own servers and databases. There is no central directory or approval process.
- Volunteer flexibility. Volunteers can participate in multiple projects. They control which projects they participate in, and how their resources are divided among these projects.
- Flexible application framework. Existing applications in common programming languages C, C++, Fortran can run as BOINC applications with little or no modification. An application can consist of several files.
- Security. BOINC protects against several types of attacks. For example, a digital signature based on public-key encryption protects against the distribution of viruses.
- Source code availability. BOINC code is distributed freely.
- Server performance and scalability. The BOINC server software is extremely efficient, so that a mid-range server can manage millions of jobs per day. The server architecture is highly scalable, making it easy to increase server capacity or availability by adding several computers.
- Support for large data. BOINC supports applications that produce or consume large amounts of data, or that use large amounts of memory. Data distribution and collection can be spread across many servers. Users can specify limits on disk usage and network bandwidth. Work is dispatched only to computers able to handle such data.
- Multiple participant platforms. The BOINC core client is available for most common platforms (Mac OS X, Windows, Linux and other Unix systems). The client can use multiple CPUs as well as graphics cards for calculations.
- Open, extensible software architecture. BOINC provides a documented interface to many of its key components, making it possible for third-party developers to create software and websites that extend BOINC.
- Volunteer community support. BOINC provides web-based tools, such as user profiles and private messaging that encourage volunteers to form an online community.
- Quick and easy integration of a new computer into the task solving process. Connecting a computer on Linux operating systems can be accomplished by running the appropriate script over the Internet, and the device does not need to be manually configured. In the case of manual configuration, the process is short and does not last more than one minute for each new computer.

The BOINC architecture is depicted in Figure 2.

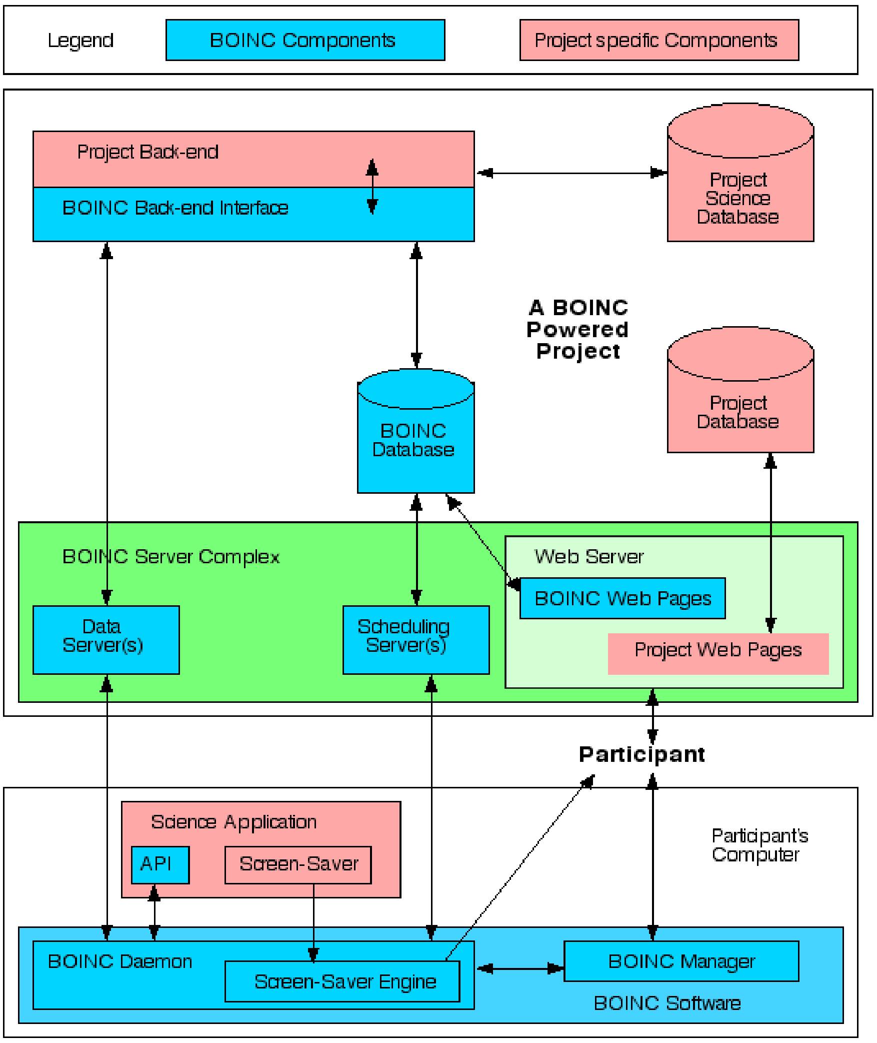

Fig. 2. BOINC architecture

Conclusion

The BOINC system is designed as a massive computational system in order to manage computing in high demand. BOINC can also be used when data collection from a large number of regions is needed. In the example presented in the text above, thanks to the single-board Raspberry PI 3 model B, it is possible to analyse an image from camera in a real time by using software Openalpr that is able to detect the vehicle license number from camera streams.

The presented examples, BOINC and Openalpr software, can be used, for example for automatic detection of the vehicle entry to toll roads, and still millions of devices for collecting data can be integrated into the system. The robustness of the BOINC system from the view of managing a large number of tasks is practically tested via large projects such as SETI@home, LHC@home, in which millions of computers around the world are involved in computing.

This work is supported by: KEGA 015ŽU-4/2017 Digital technologies for machine building study programs as a part of strategy "Internet of Things" and VEGA 1/0795/16 Development of effective methods for correction and optimization of coupled mechanical system.

References

1. P. Pecháč, M. Sága, *Memetic algorithm with normalized RBF ANN for approximation of objective function and sedondary RBF ANN for error mapping,* Procedia engineering **177**, 540 – 547, ISSN 1877-7058, (2017)
2. M. Sága, R. Bednár, M. Vaško, *Contribution to modal and spectral interval finite element analysis,* Springer proceedings in physics **139**, 269-274, ISSN 0930-8989, (2011)
3. P. Orsansky, B. Ftorek, P. Durcansky, *Mathematical model of a closed hot air engine cycle using MATLAB Simulink,* AIP conference proceedings **1608**, 173-176, ISSN 0094-243X, (2014)
4. J. Judova, M. Pajtasova, D. Ondrusova, *Effects on the bee colonies vitality in Slovakia*. International multidisciplinary scientific geoconference-SGEM **3**, 274-254, ISSN 1314-2704 (2016)
5. J. Valiska, S. Merchevsky, R. Kokoska, *Object tracking by color – based particle filter techniques in video sequences,* IEEE Radioelektronika 2014, ISBN 978-1-4799-3715-8, (2014)
6. C. M. Sukanya, G. Roopa, P. Vince, *A survey on object recognition methods.* IJCSET **6**, 48-52, ISSN 2231-0711, (2016)
7. M. Al-Smadi, K. Abdulrahim, R. A. Salam, *Traffic surveillance a rewiew of vision based vehicle detection, recognition and tracking*, International journal of applied engineering research **6**, 713-726, ISSN 0973-4562, (2016)
8. S. Sivaraman, M. M. Trivedi, *A review of recent developments in vision-based vehicle detection*, IEEE Intelligent vehicles symposium IV, 23–26 june 2013, Gold coast, Australia
9. K. Hulin, Ch. Long, G. Feng, C. Jiajie, C. Leanne, Y. Hong, *Combining region-of-interest extraction and image enhancement for nighttime vehicle detection,* IEEE Intelligent systems **31**, 57-61, (2016)
10. D. P. Anderson, Boinc: A system for public-resource computing and storage, https://boinc.berkeley.edu/grid_paper_04.pdf
11. Boinc web site, https://boinc.berkeley.edu