J. Yaghob (Ed.): ITAT 2015 pp. 17–22
Charles University in Prague, Prague, 2015

Assessing Applicability of Power-Efficient Embedded Devices for Micro-Cloud Computing

Martin Kruliš, Petr Stefan, Jakub Yaghob, and Filip Zavoral

Parallel Architectures/Algorithms/Applications Research Group
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské nám. 25, Prague, Czech Republic
{krulis,yaghob,zavoral}@ksi.mff.cuni.cz, ptr.stef@gmail.com

Abstract: Distributed computing and cloud phenomenon have become an intensively studied topic in the past decade. These technologies have been enveloped with attractive business models, where the customer pays only for the resources or services which have been actually utilized. Even though this popularity lead to rapid development of distributed algorithms, virtualization platforms, and various cloud services, many issues are still waiting to be solved. One of these issues is the question of power efficiency. In this paper, we investigate possibilities of applying single-board computers as platform for distributed systems and cloud computing. These small devices (such as Raspberry Pi) are quite power efficient and relatively cheap, so they may reduce the overall cost for cloud services. Furthermore, they may be employed to create small clusters that could replace traditional enterprise servers and achieve lower cost and better robustness for some tasks.

Keywords: reliability, distributed systems, cloud computing, micro cloud, power efficiency, Raspberry PI

1 Introduction

Distributed computing has been an intensively studied topic since the dawn of computer science. The idea of utilizing multiple ordinary devices instead of a single powerful one brought many advantages, such as much easier scaling, possibly higher robustness, or better utilization of spare hardware. On the other hand, distributed computing is encumbered with many challenges that include the question of efficiency, communication and synchronization overhead, or the necessity of handling failures of individual nodes.

In combination with modern technologies and hardware virtualization, the distributed computing lead to the inception of the cloud phenomenon, where large complex systems are presented to users not in a form o a distributed system, but as virtual hardware, programming platform, or even specialized services. In this form, the user is completely shielded from tedious details of system design. Furthermore, the concept of cloud allows much more efficient allocation of hardware resources, from which benefits both the cloud providers (since they have less hardware to buy and maintain) and cloud customers (who pay only for resources they really utilize).

With the growth of the cloud infrastructure, the power efficiency become a more and more important problem. Despite the fact that the cloud services utilize underlying hardware more efficiently than it could have been used by individual users, the pressure to reduce power consumption of this infrastructure is raising steadily. One of the possibilities is to utilize more efficient hardware that requires less power to perform the same task. A quite promising platform are the ARM CPUs which are currently utilized in mainstream mobile and other handheld devices. However, the majority of enterprise servers and professional solutions use CPUs based on x86 architecture which have more computational power. Nevertheless, these solutions are considered less efficient at least for some tasks.

Some specialized problems cannot utilize cloud solutions for various reasons such as security or domain-specific constraints, hence they must be hosted on privatized clusters. Beside the power efficiency issues, small clusters may benefit from small ARM-based devices in other ways. For instance, utilizing many single-board computers instead of a few enterprise server may be cheaper. Furthermore, using many devices allows more fine-grained performance scaling.

In this paper, we study issues of power efficiency in distributed systems, clouds, and micro-cloud solutions. We have selected the Raspberry Pi single-board computer as a representative of power efficient hardware based on ARM platform. We have tested performance of this device using our own application benchmark and compare the results with a commodity desktop PC and an enterprise server to determine the power-to-performance ratio and relative applicability for various problems. Even though the results are only approximate, the Raspberry Pi seems to be a viable candidate for green micro-cloud solutions.

The paper is organized as follows. More detailed overview of distributed systems and cloud solutions is provided in Section 2. Section 3 revises related work on micro-cloud systems. In Section 4, we present details about our tested platform – the Raspberry Pi device. Section 5 summarizes our empirical evaluation, Section 6 outlines possible applicability of these technologies, and Section 7 concludes the paper.

2 Distributed Applications and Cloud Solutions

Among the most important computing technologies that are in use nowadays are Distributed Systems and Cloud Computing Systems. Distributed system [1] is a collection of computers that work together and appear as one large computer. These computers cooperate to solve usually complex tasks; they are mutually interconnected to provide a massive computing power.

The basic advantages of distributed systems are:

- High performance
- Transparency
- Resource sharing
- Reliability and availability
- Incremental extensibility

On the other hand, the disadvantages that we may face in distributed systems are complexity, software development difficulties, networking problems, and security issues.

Contemporary cloud solutions has evolved from the earlier distributed systems. Cloud computing (despite the term has no exact definition) can be considered as a specialized form of distributed computing where virtualized resources are available as a service over the internet. These services usually include infrastructure, platform, applications, storage space and many other vendor-specific modules, libraries and frameworks. The users pay only for the services or resources they actually use. The underlying resources, such as storage, processors, memory, are completely abstracted from the consumer. The vendor of the cloud service is responsible for the reliability, performance, scalability and security of the service.

Cloud computing has many benefits, but cases exist where some data cannot be moved to the cloud for various reasons. In some cases, data may be generated at rates that are too big to move or at rates that exceed transfer capacity, for example in surveillance, operations in remote areas, and telemetry applications. In other cases, security concerns or regulatory compliance requirements might limit the use of the cloud.

Green computing [2] [3] refers to the environmentally responsible use of computers and any other technology related resources. Green computing includes the implementation of best practices, such as energy efficiency central processing units (CPUs), peripherals and servers [4]. Green Cloud is a computing facility that is entirely built, managed and operated on green computing principles. It provides the same features and capabilities of a typical cloud solution but uses less energy and space, and its design and operation are environmentally friendly.

3 Micro-Cloud Solutions

The recent introduction of the Raspberry Pi, a low-cost, low-power single-board computer, has made the construction of miniature green cloud systems more affordable.

Glasgow Raspberry Pi Cloud [5] is a model of a micro-cloud solution composed of clusters of Raspberry Pi devices. The PiCloud emulates every layer of a cloud stack, ranging from resource virtualisation to network behaviour, providing a full-featured cloud computing research and educational environment.

Iridis-pi [6] cluster consists of 64 Raspberry Pi Model B nodes each equipped with a 700 MHz ARM processor, 256 Mbit of RAM and a 16 GiB SD card for local storage. The cluster has a number of advantages that are typical for micro-clouds, such as low total power consumption, easy portability due to its small size and weight, and passive, ambient cooling. These attributes make Iridis-Pi ideally suited to educational applications, where it provides a low-cost starting point to inspire and enable students to understand principles of high-performance computing.

Sher.ly [7] builds a network-attached storage (NAS) device, the Sherlybox, that comes with its own peer-to-peer virtual private network and file server. The Sherylbox is built around the Raspberry Pi Model B computer. It comes with 512 MB of RAM, two USB 2.0 ports, 802.11n Wi-FI, and a 100mb Ethernet port. Instead of just the naked board, the Sherylbox comes with a case, a 4GB eMMC flash drive, and an optional 1 TB hard-drive. The company claims that with external USB drives, it can support up to 127 USB drives.

Tonido [8] offer a compelling alternative to public cloud file services allowing consumers to leverage their existing computers or IT infrastructures to keep control over their own data. It is available for a wide list of operating systems running on different hardware including Raspberry Pi using Raspbian or Raspbmc OS. Nimbus [9] is another example of a micro-cloud solution.

Although all of the abovementioned solutions are intended especially to personal or educational use (and a majority of scientific papers expect such use-cases), we claim that, under certain conditions, there may exist a wider range of possible applications. Some of them are discussed in Section 6.

4 Single-board Computers

Single-board computers constitute a special brand of computational devices which aim for compactness and power efficiency. These devices have various applications in robotics, intelligent household devices, smart monitoring stations, and many other domains. Even though their performance cannot compete with mainstream desktop PC and servers, they may achieve better power to performance and power to cost ratios. In this section, we present a few examples of compact single-board devices and revise the properties of Raspberry Pi device, which was selected as a representative for our research.

4.1 Computer Examples

Arandale Board [10] is a single-board computer powered by Samsung Exynos 5, which is an ARM CPU. The board is equipped with 2GB of RAM and various common peripherals such as USB 3.0, WiFi, GPS module, or interface for LCD display. The board is mainly designed for tablets and embedded computers; however, it also provide adequate performance for cloud computing. On the other hand, most of its peripherals are undesired for a solely computational solution and they may increase the overall cost.

AMD presented a *Gizmo 2* board [11], which is also a single-board computer that is compatible with x86 architecture. It comprises specialized double core APU (clocked at 1 GHz), which is a single chip that integrates power efficient CPU and Radeon GPU, and 1 GB of DDR3 RAM. The board is designed to provide all-in-one PC solution, so it is equipped with traditional interfaces such as USB, Gigabit Ethernet, or HDMI. The integrated GPU may provide excellent performance (with respect to power consumption); however, the price of the board is rather high in comparison to similar devices.

Intel entered the domain of power efficient single-board devices with *Galileo* development board [12]. It is equipped with Intel Quark X1000 CPU, which is a single-core Pentium-based 32-bit processor clocked at 400 MHz, and 256 MB of DRAM. The board is compatible with Arduino [13] device specification, which allows it to share peripherals and extensions designed for this platform.

Another similar platform is *Intel Edison*. It also contains Intel Quark CPU, but the Edison platform aims mainly at wearable devices and extensive miniaturization.

The *Parallela* board [14] is a relatively novel accomplishment in the field of efficient parallel hardware. Unlike many other devices, Parallela was designed by a small company Adapteva. It is equipped with ARM Cortex-9 CPU, FPGA, and a Epiphany coprocessor. The coprocessor is perhaps the most intriguing part of this hardware, since it is a specialized power-efficient parallel processing unit which organizes the cores in a 2D grid. This device may be the most promising alternative for a Raspberry Pi in the terms of power efficiency and total performance. On the other hand, Parallela is approximately $3\times$ more expensive than Raspberry Pi.

4.2 Raspberry Pi

Raspberry Pi [15] is one of the first low-cost devices that is capable of running a traditional operating system (in this case Linux), so it can be used as a modest desktop PC. It was originally created as a cheap platform that would allow children to learn basics of programming, but it was quickly adopted for various applications, such as embedded devices, simple audio and video players, etc.

At present, there are several configurations available (models A, B, and B+) and a new version called Raspberry Pi 2 was introduced to the market. In this work, we present (and measure) the properties of Raspberry Pi model B+, which is the newest revision of the original Raspberry Pi (before its second version was released).

The device is powered by Broadcom BCM2835 CPU, which is an ARMv6 processor clocked at 700 MHz. The graphics is rendered by VideoCore IV GPU clocked at 250 MHz. The GPU is capable of decoding a full-HD video in real time; however, there is currently no API (such as OpenCL) provided for computations. The system holds 512 MB of DDR2 RAM, which is shared both by CPU and GPU. Persistent memory is not integrated on the board, but it contains an interface for memory cards. We have used commodity 32 GB Kingston MicroSDHC card (class 10) as the persistent data storage.

Raspberry Pi has many external interfaces. Beside traditional USB or HDMI connector, it also holds custom GPIO port or I2C bus, which make the device suitable as a high-level controller for many electronic devices. The most important interface for our intentions is the 100 Mb Ethernet. Unfortunately, the Ethernet interface is internally connected via USB 2.0 bus. This bridged solution does not reduce the overall throughput, but slightly increases the communication latency.

The system is designed mainly for Linux operating system, but it can accomodate virtually any system that can run on ARM CPU (e.g., RiscOS). For the convenience of the users, the community has prepared modified distribution of Debian Linux called Raspbian and some other distributions based on Ubuntu or Fedora are also available. We have used the Raspbian in our experiments, since it is the recommended system.

5 Experimental Results

We have subjected Raspberry Pi to a custom set of performance tests to assess its applicability for distributed computing and cloud applications. The performance results are compared with results from a desktop PC and commodity server in the perspective of the power consumption. Let us emphasize that the results measured for Raspberry Pi and for full-sized computers are not directly comparable and provide only approximate comparison since our measurements of power consumption does not use same methodology and our benchmark is only single-threaded.

5.1 Experimental Setup

The parameters of Raspberry Pi are detailed in Section 4.2. The referential desktop PC is equipped with Intel Core i7 870 CPU, which has four physical (8 logical) cores clocked at $2{,}93$ GHz, and 16 GB DDR3-1600 RAM. The persistent storage is represented by two 100 GB SSD disks connected in RAID 1.

The referential server is Dell PowerEdge M910. It is 4-way cache-coherent NUMA[1] system, where each node has 8 physical (16 logical) cores clocked at 2 GHz. Each node manages 32 GB RAM – i.e., the whole system comprises 64 logical cores and 128 GB of internal memory. The server was connected to Infortrend ESDS 3060 disk array comprising two 400 GB SSD disks and 14 magnetic disks of 4 TB each. Both desktop PC and server are running Red Hat Enterprise Linux 7 as an operating system.

To asses the performance, we measure the real execution time of prepared tests. All tests are executed on the same data inputs and the size of the input is selected so that the test takes reasonable time on Raspberry Pi and at least a few seconds on desktop PC and server. Each test was repeated 10× and the average time is presented as the final result. The values were processed by statistical methods to remove outliers (times tainted with errors of measurement).

The power consumption of the Raspberry Pi was determined by KCX-017 device, which measures voltage and current on an USB power cord, since Raspberry Pi is powered via USB. The power consumption is equal to voltage times current ($P = UI$) and we employ additional correction factor of $1/0.8$, which simulates loss on power source with efficiency of 80%. The power consumption was between 1.2 W (idle device) to 1.7 W (performing cryptographical tests).

The power of our server was measured on its power controller embedded in server chassis. We also include estimated partial consumption of the chassis itself and additional equipment (such as cooling infrastructure), hence we will operate with aggregated approximate consumption of 500 W. The power consumption of the desktop PC was calculated from the component specifications since we were not able to measure this value with reasonable effort. For our purposes and intentions, we will operate with the value 250 W.

5.2 Tests

The performance experiments were design to test various aspects of the device. Since we are trying to determine applicability of Raspberry Pi as a platform for distributed system and cloud infrastructure, we have selected algorithms that cover many different domains:

- *aes* – The Rijndael (Advanced Encryption Standard) algorithm [16] for symmetric cryptography.
- *scrypt* – Computing scrypt [17] hash function.
- *sha256* – Computing SHA256 hash function.
- *dijkstra* – Finding shortest path in a sparse graph using Dijkstra algorithm [18] with regular heaps.
- *hash* – Simulation of database hash-join operation using integer keys.
- *merge* – Simulation of database merge-join on sorted data streams using integer keys.
- *levenshtein* – Wagner-Fischer dynamic programming algorithm [19] that computes Levenshtein edit distance
- *multiply* – Naïve ($O(N^3)$) algorithm for matrix multiplication on float numbers.
- *strassen* – Strassen algorithm for matrix multiplication on float numbers.
- *quicksort* – Quicksort [20] in memory sorting algorithm implemented in C++ `std::sort` routine applied on integers.
- *zlib* – DEFLATE [21] compression algorithm implemented in Zlib.

Beside these application tests, we have performed additional tests designed to determine the speed of internal memory, effectivity of its CPU caches, and performance of the persitent storage (i.e., the SD flash card). However, we do not present detailed results of all these tests for the sake of the scope.

5.3 Results

The application benchmark results are presented in Figure 1. The results depict computational power efficiency normalized relatively to Raspberry Pi (individually for each algorithm) – i.e., higher value means greater power consumption with respect to computational performance. Hence, we can directly determine, which platform is better and which is worse for a particular problem. Let us note that we have adjusted the results so that they take the multi-core and multi-processor nature of the desktop PC and the server, since our benchmark is only single threaded. The performance of the full-sized computers were multiplied by the number of their physical cores.

The results indicate that Raspberry Pi is quite efficient for memory-intensive tasks. For some tests (especially database merge joins), the Raspberry Pi even outperforms both desktop PC and server. On the other hand, number crunching operations (such as the matrix multiplication on float numbers) are more suitable for x86 architecture, since it may employ SIMD instructions. We have performed additional synthetic memory-oriented experiments and they have confirmed this observation.

In addition to application tests, we have measured performance of the persistent storage. The throughput of individual operations is presented in Table 1. Let us emphasize that the Raspberry Pi has only a commodity SD card, while the server uses enterprise disk array.

[1] Nonuniform Memory Architecture

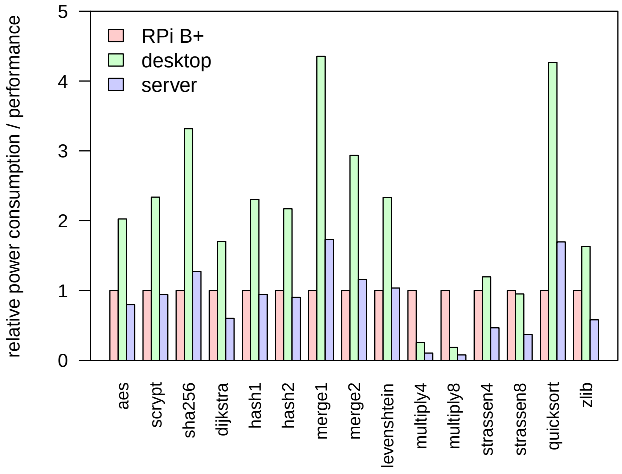

Figure 1: Relative efficiency of application tests

	Rasbperry Pi	desktop PC	server
rand. read	0.7	9.3	5.3
seq. read	17.3	171.9	404.5
seq. write	1.7	58.1	86.9

Table 1: Persistent storage performance (MB/s)

The results indicate that the performance of the Raspberry Pi is approximately 10-100× worse than the performance of other two platforms. On the other hand, if the data are distributed evenly among the devices, each Raspberry Pi has to handle two orders of magnitude smaller amount of data, so the performance is comparable. Furthermore, the devices may also utilize external disk array connected via 100 Mbit ethernet, which should provide data transfers around 5 MB/s.

6 Applicability

In this section, we would like to outline possible applicability of single-board devices for various problems. Besides the obvious cost issue, the presented solutions are expected to take advantage of two greatest benefits over traditional servers or desktop PCs:

- increased robustness
- and better heat dissipation.

The robustness is one of the expected properties of many distributed systems. However, when one server fails, the total drop of performance could be significant, especially in case of smaller and mid-sized clusters. When small devices such as Raspberry Pi are used, the failure of a single device is hardly noticable on the overall performance and the faulty hardware could be replaced more quickly. Furthermore, small devices permit more fine-grained redundancy in the system.

The heat dissipation presents a challenging problem for modern servers as most powerful x86 processors easily produce over a hundred watts of thermal power. Hence, the servers, their chassis, and the server racks employ sophisticated cooling mechanism to drive the undesired heat out off the server room. In case of smaller devices, the produced heat has much lower watt per area ratio, so it is much easier to cool these devices.

6.1 Replacing Tradional Servers

A direct applicability of a Raspberry Pi cluster could be to replace traditional enterprise servers. Based on the scale, this solution could work for a small cluster within one server room or as a large distributed system that provides cloud services. In any case, the main advantage of such solution is the more evenly distributed heat output. Hence, the system does not to have a server room with powerful cooling system.

It may even be considered to place most of the hardware outside of a server room and integrate the single board computers into the infrastructure of a building or into regular rooms (offices, etc.). The Raspberry Pi does not require a cooling fan, hence such solution would not increase background noise inside the building. Furthermore, the heat produced by the devices may be used as part of internal heating system and the I/O ports (USB or GPIO) could be used to operate building sensors.

6.2 Outdoor Micro-Clouds

The compactness and low consumption of single-board computers may be utilized in many applications which could be characterized as *outside the server room* projects. Such projects would include robotics, autonomous vehicles and aircraft, probes and intelligent exploration devices, etc. A micro-cloud solution could increase robustness of these devices, which could be important since their hardware is subjected to much harsh physical conditions than hardware located in a server room or in an office.

Let us use an autonomous car (which is a domain that spawned an intensive research in the past few years) as an example of such outdoor device that required nontrivial computational power. A cluster of single-board computers may provide much scalable hardware for navigation computations. For instance, when the car is driving on a straight road in an unpopulated area, it requires much less computational power to track and analyse surrounding environment. Hence, it may shut down most of the devices in the cluster to save energy. On the other hand, when driving inside a city, it may turn on the whole cluster to get necessary computational power. Finally, the decentralized nature of the hardware may provide enough computational power even in extreme cases, such as when part of the vehicle is compromised in a car crash.

7 Conclusions

In this paper, we have addressed the issue of power efficiency in cloud systems. Many systems would benefit greatly from a hardware that provide less computational power, but which is more power efficient and has lower initial and maintenance costs. We have designed an application benchmark for small devices that tests various known algorithms. The benchmark was applied on the Raspberry Pi, which is one of the first single-board computers. The Raspberry Pi is very power efficient and cost around $30, which makes it a good candidate to be a worker in a green cluster or a micro cloud. The benchmark results indicate that current version of Raspberry Pi is competitive with desktop PC as well as an enterprise server in tasks that can be idealy distributed.

In our future work, we would like to test other similar devices, especially the second version Raspberry Pi and the Parallela board with Epiphany coprocessor. Furthermore, we are planning to build a small cluster from these devices to measure the total consumption more precisely and to determine the communication overhead of various distributed algorithms.

Acknowledgment

This paper was supported by Czech Science Foundation (GACR) projects P103/13/08195 and P103/14/14292P and by SVV-2015-260222.

References

[1] Tanenbaum, A. S., Van Steen, M.: Distributed systems: principles and paradigms (2nd Ed.). Prentice-Hall, 2006

[2] Priya, B., Pilli, E., Joshi, R.: A survey on energy and power consumption models for greener cloud. In: Advance Computing Conference (IACC), 2013 IEEE 3rd International, 2013, 76–82

[3] Kumar, S., Buyya, R.: Green cloud computing and environmental sustainability. In: Harnessing Green It, John Wiley and Sons, Ltd, UK, 2012, 315–339

[4] Yanovskaya, O., Yanovsky, M., Kharchenko, V.: The concept of green cloud infrastructure based on distributed computing and hardware accelerator within fpga as a service. In: Design Test Symposium (EWDTS), 2014 East-West, 2014, 1–4

[5] Tso, F. P., White, D. R., Jouet, S., Singer, J., Pezaros, D. P.: The glasgow raspberry pi cloud: A scale model for cloud computing infrastructures. In: Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on, IEEE, 2013, 108–112

[6] Cox, S. J., Cox, J. T., Boardman, R. P., Johnston, S. J., Scott, M., O'Brien, N. S.: Iridis-pi: a low-cost, compact demonstration cluster. In: Cluster Computing **17** (2014), 349–358

[7] sher.ly, "Private Cloud Solution for Sensitive Data Sharing with Secure Access Control." [Online]. Available: http://sher.ly/

[8] Tonido, "Turn your Raspberry PI into your personal cloud." [Online]. Available: http://www.tonido.com

[9] Cloudnimbus, "Nimbus - Personal Cloud for Raspberry Pi." [Online]. Available: www.cloudnimbus.org

[10] "Arandale board." [Online]. Available: http://www.arndaleboard.org/

[11] Gizmosphere, "Amd gizmo 2." [Online]. Available: http://www.gizmosphere.org/products/gizmo-2/

[12] Intel, "Galileo Gen 2 Development Board, url=http://www.intel.com/content/www/us/en/do-it-yourself/galileo-maker-quark-board.html,."

[13] M. Banzi, D. Cuartielles, T. Igoe, G. Martino, and D. Mellis, "Arduino uno." [Online]. Available: http://www.arduino.cc/

[14] Adapteva, "Parallela." [Online]. Available: https://www.parallella.org/

[15] Raspberry Pi Foundation, "Raspberry Pi single-board computer." [Online]. Available: https://www.raspberrypi.org/

[16] Gladman, B.: A specification for rijndael, the aes algorithm. at fp. gladman. plus. com/cryptography_technology/rijndael/aes. spec **311** (2001), 18–19

[17] Percival, C.: Stronger key derivation via sequential memory-hard functions. Proceedings BSD Canada, 2009

[18] Dijkstra, E. W.: A note on two problems in connexion with graphs. Numerische Mathematik **1 (1)** (1959), 269–271

[19] Wagner, R. A., Fischer, M. J.: The string-to-string correction problem. Journal of the ACM (JACM) **21 (1)** (1974), 168–173

[20] Hoare, C. A.: Quicksort. The Computer Journal **5 (1)** (1962), 10–16

[21] Deutsch, L. P.: Deflate compressed data format specification version 1.3, 1996