Energy and Cost Considerations for Single Board Computers Usage in Citizen Science Scenarios

Pedro Verdugo, Joaquín Salvachúa and Gabriel Huecas
Grupo de Internet de Nueva Generación, DIT, ETSIT, Universidad Politécnica de Madrid, Madrid, Spain

Keywords: Citizen Science, Green Computing, Energy Efficiency, Single Board Computers.

Abstract: The rich availability of single board computers as an expansion of the traditional embedded system provides a low cost, easily managed, execution ready and scalable entry-level infrastructure, resulting in a new computing paradigm termed as Fog Computing. Simultaneously, the current expansion of Citizen Science initiatives along with the generalization of IOT projects precludes the need for individually managed, low complexity data processing systems, generating a new user-oriented ecosystem that has come to be defined as the Edge Cloud. In this document the authors will briefly study the adequacy of the current user-grade SBC hardware offerings to cover the needs set by the Citizen Science paradigm, starting by studying current Citizen Science projects in order to define the aforementioned set of specific requirements, and subsequently analyzing the hardware selection and provisioning process given the current non-enterprise user-grade supply for SBC computers, taking special consideration to power usage and total cost of ownership factors from a green computing perspective.

1 INTRODUCTION

The main goal of Citizen Science is to achieve an increase of citizen participation and involvement in scientific research. This user-based research needs to be supported by a cost-driven, energy-conscious and flexible hardware infrastructure. A raise in the popularity and availability of **Single Board Computers** (from now on, SBCs) sets these systems as possible candidates to host Citizen Science projects. In order to verify the adequacy of the SBC infrastructure approach to Citizen Science projects, we will begin by briefly studying the history and current state of the Citizen Science paradigm, then we will identify the required needs for a common Citizen Science project, and finally we will study the current SBC hardware offerings capacity to fulfill the discovered needs.

1.1 The Citizen Science Paradigm

Widely understood as the collaboration of volunteers in the scientific process, since Silvertowns' seminal paper (Silvertown, 2009), Citizen Science has been proven a valuable method to deal with otherwise unattainable wide-scoped projects.

The number of samples and observations required to obtain significant results for some areas of knowledge, such as the biology(Kelling et al., 2015) and biochemistry (Kawrykow et al., 2012) fields, makes it impossible to resort to traditional expert data gathering and categorizing techniques, whereas motivated and loosely trained private individuals can provide sufficient research data with quality standards comparable to those attained by trained experts.

Illustrating the previous point, most common Citizen Science projects to date have been focused on data collection via Crowdsourcing, defined as the gathering of data and interactions from a wide variety of users and sources in a distributed fashion(Panchariya et al., 2015), leaving the processing side to field experts and data scientists. This traditional data chain for Citizen Science projects (Newman et al., 2012) sets *typical collaborator tasks* as data collection and data transcription whereas *typical data scientist tasks* include problem definition, data cleanup, data analysis, data processing and results dissemination. Taking into consideration the previous point, it is not strange that the most commonplace current concern in Citizen Science projects is the increase of the involvement and motivation of the individual volunteer in all aspects of the scientific process, widening the collaborator's scope of influence, interaction and participation for all the defined stages. This participation cannot be unbound; to warrant observation quality there

Verdugo, P., Salvachúa, J. and Huecas, G.
Energy and Cost Considerations for Single Board Computers Usage in Citizen Science Scenarios.
In *Proceedings of the 7th International Joint Conference on Pervasive and Embedded Computing and Communication Systems (PECCS 2017)*, pages 35-40
ISBN: 978-989-758-266-0

are available widely studied and defined participation protocols, as seen in (Sprinks et al., 2017).

In order to set the desired characteristics for a sustainable citizen-driven hardware infrastructure, in the next section we will summarize the most popular current Citizen Science initiatives.

1.2 Current Citizen Science Projects

The **Berkeley Open Infrastructure for Network Computing** (BOINC) is a widespread middleware system for distributed processing, which hosts a variety of projects and enjoying a long lifetime (Anderson, 2004). Some of the disadvantages of BOINC include the need to port code to run in heterogeneous architectures, as well as the common dependency management and user authentication issues.

The BOINC based SETI@home project is the most long lived of the herein presented (Korpela et al., 2015), but continues to grow and refresh its experiment set as shown by the CASPER child project (Werthimer, 2015), where FPGAs and dedicated GPUs are used to increase processing power.

Also based on the aforementioned BOINC project, Asteroids@home (Ďurech et al., 2015) aims for the reconstruction of asteroid shapes based on volunteer processing power, and, as commented above, must include binaries with native support for all common operating systems.

The Folding@home project has also had a long running trajectory (Beberg et al., 2009), and presents a robust infrastructure based on machines with dedicated roles. However, it seems to have decreased in growth, at least in the academic environment related to our context of study.

The **GROMACS** project (Abraham et al., 2015), is an active and impressive attempt at generating parallellizable algorithms for chemistry application in a local environment. With over 2 million lines of code, it requires a compilation toolchain for supported platforms.

A mix of available resources from GROMACS and Folding@home is possible as shown in (Lawrenz et al., 2015), combining fast local resources with possibly abundant remote ones.

We can also note the existence of independent Crowd Computing efforts, as illustrated in (Kovacs and Lovas, 2014), also commonly based on the BOINC platform and thus suffering from its limitations.

Once the advantages and shortcomings of the given projects are known, in the next section we will abstract from them a set of desirable characteristics for a common Citizen Science infrastructure.

Table 1: Single Board Systems.

Name	*Processor*	*Price ($)*
ODroid-C1+	Cortex-A5	37
ODroid-U3	Cortex-A9	59
Beaglebone Black	Cortex-A8	55
Banana-Pi BPI M2	Cortex-A7	39.50
Banana pi BPI-M3	Cortex-A7	75
Hummingboard-i1	Freescale GC880	70
Minnowboard Max	Intel Atom	99
Raspberry Pi 2B	Cortex-A7	35

1.3 Detected Citizen Science Hardware Requirements

The focus of our current work is the generation of a simple, cheap and flexible enough environment and infrastructure to allow individuals to set up, use and obtain immediate results for their own science projects, while also enabling the sharing of data or processing power in distributed endeavors. The study of the initiatives presented in the previous section lets us detect a series of common defining factors needed to achieve our goal of a user-level autonomous hardware environment, that will be detailed below:

- **Initial Cost**: The environment component cost should be flexible enough to warrant any desired level of initial user economical environment.
- **Efficiency**: The components should be as less energy wasteful as possible, following the green computing paradigm. Thermal considerations and power usage will be of paramount importance, and will also reflect on the accumulated running cost.
- **Simplicity.** The integration of the environment components should be as simple as possible, not requiring complex hardware setups or custom equipment. For that purpose, when given the choice we will opt for common solutions.
- **Flexibility.** The system should be as scalable as required for the user needs from a physical view as well as price-wise. Related to the previous point, the system components should be easily interchangeable and of widespread availability.

2 HARDWARE CONSIDERATIONS

Taking into consideration the resulting factors from the previous section, the use of single board systems

Table 2: Hardware Specifications.

Board	CPU			DMIPS		RAM		Net	Power	
Name	*Family*	*Cores*	*GHz*	*Single*	*Total*	*Mb*	*Mhz*	*Mbps*	*W-Full*	*W-Idle*
ODroid-C1+	Cortex-A5	4	1.5	1570	6280	1024	900	1000	2.31	0.73
ODroid-U3	Cortex-A9	4	1.7	1780	7120	2048	900	1000	4.25	1.75
Beaglebone Black	Cortex-A8	1	1.0	2000	2000	512	600	100	2.44	0.42
Banana-Pi BPI M2	Cortex-A7	2	1.0	1270	2540	1024	432	100	2.01	0.43
Banana pi BPI-M3	Cortex-A7	4	2.0	1270	5080	2048	480	100	2.35	1.32
Raspberry Pi 2B	Cortex-A7	2	0.9	1180	2360	1024	900	100	2.25	1.15

is obviously a reasonable choice to increase the simplicity and flexibility of our setup, as shown on the excellent performance comparison given in (Lencse and Répás, 2015). In Table 1 we will update the mentioned table to detail some of the current user grade SBC offerings.

Table 1 moves us to consider the idea of using ARM (Advanced RISC Machine) based architecture, which has recently been proven to be a cost-oriented viable alternative to the traditional x86 architecture as shown in Keipert (Keipert et al., 2015). These ideas have already been successfully taken to practice in the OLPC project as illustrated in Gahire (Gaihre, 2015). Restricting our selection to ARM architectures, we can now proceed to compare individual board characteristics, always taking into consideration the restrictions for our particular goal.

2.1 CPU Comparison

For starters, table 2 details the number of cores and work frequencies along with the different ARM families for each of our SBCs of interest.

These different values assert the need to compare the CPU performance for each board across different processor families and frequencies. For that purpose, we will use the Dhrystone tool, which lets us obtain an standardized MIPS (Millions of Instructions Per Second) value independently from the processor family or architecture, allowing simpler inter-system comparison.

The results shown in table 2 are mostly self-explanatory, but we can note how the Cortex-A9 system seems to have the better score at raw processing, surprisingly closely followed by the older A5 architecture.

2.2 RAM Comparison

RAM size and I/O throughput is of high importance in data-oriented environments. This may probably be the biggest flaw of the current board offerings. As OS and applications will have to share the meager memory space, there will constantly be a lack of fast access RAM, and maybe pagination issues.

Table 2 shows how the Odroid-U3 system seems to be the best technical offering, noting the use of LPDDR3 technology, a standard in mobile applications.

2.3 Network Comparison

Given the common use cases for our system we can expect very high network I/O throughput between local machines and also with remote systems, making cabled ethernet a desirable item.

Table 2 shows that the ODroid family takes the lead here with an integrated 1Gbps ethernet NIC.

2.4 Storage Comparison

In table 3 we will very briefly consider the storage capacities of the different boards, with the parameter *c* corresponding to the class of the used microSD card. This class corresponds directly to the speedup over a reference 1MBps transfer speed, and given the current exponential cost of higher than 10-class microSD cards will be the main practical limitation to our data transfer capabilities.

Table 3: Storage Specifications (Mbps).

Name	*Interface*	*Read*	*Write*
ODroid-C1+	microSD	*c*	*c*
-	eMMC	250	90
ODroid-U3	microSD	*c*	*c*
-	eMMC	250	90
Beaglebone Black	microSD	*c*	*c*
Banana-Pi BPI M2	SD	*c*	*c*
-	SATAII	300	300
Banana pi BPI-M3	SD	*c*	*c*
-	SATAII	300	300
Raspberry Pi 2B	microSD	*c*	*c*

There is also room to notice the eMMC interface provided by the ODroid family, again inherited from the mobile industry and comparable in transfer speeds

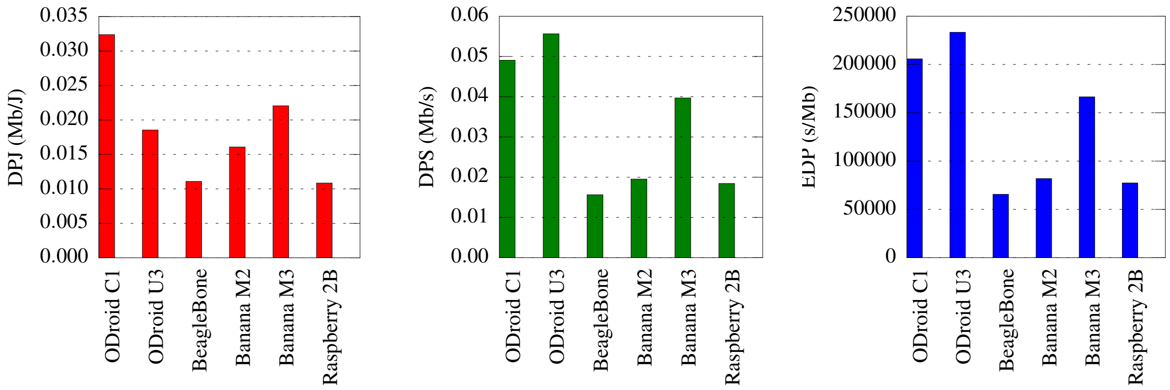

Figure 1: DPJ, DPS and EDP.

to SSD technology. The high cost and dedicated nature of such a solution, precludes us from further consideration.

3 POWER CONSIDERATIONS

The **Thermal Design Power** (TDP), or thermal design point, has been historically used to generate power usage metrics(Hennessy and Patterson, 2012). Being defined as the maximum amount of heat generated during typical computer operation (Gough et al., 2015), this concept results inadequate as a measure of a full system processing power.

The work of Hennessy clearly states that any useful metric for computer power must necessarily be tied to energy usage, and introduces the following terms:

- **Data Processed Per Second** (DPS): data processed by the system in a given time (in our case, one second).
- **Data Processed Per Joule** (DPJ): amount of data processable by the system with a given energy budget of 1 Joule.
- **Energy-Delay Product** (EDP): As introduced by Horowitz(Horowitz et al., 1994) in the transistor performance environment, and adapted by Laros III(Laros III et al., 2013), defines the time taken by the system to output a given amount of data for a fixed energy budget.

Thus, from a thermal and power usage standpoint, we can refer to the equations provided by Malik (Malik and Homayoun, 2015), used to transform the power values in table 2 into the more useful metrics given in Figure 1.

In Figure 1 we can appreciate how the ODroid family has the highest DPJ rate, mainly due to the high core speeds. These speeds, along with the elevated MIPS previously remarked, are also responsible for the DPS measurement. The EDP graph from Figure 1 shows us the price to pay for both previous high figures: an elevated EDP signifying that the ODroid family is not as energy-efficient as its lower-power counterparts.

4 COST CONSIDERATIONS

The most common tool for complete cost calculation is the **Total Cost of Ownership** (TCO) metric, as presented in Martens (Martens et al., 2012). For the system under study, we will refine the previous formulas with multinode extended considerations as developed by Barroso (Barroso et al., 2013).

To begin with, the terms of interest will be defined:

- n: Number of boards (7)
- t: Runtime (5 years)
- C_{pi}: Provisioning Cost per node ($37)
- C_{ei}: Total Electricity Cost per node ($ To be determined)
- C_h: Electricity Cost per hour (0.11 KWh)
- P_f: Full Power Usage per node (2.3W)
- P_i: Idle Power Usage per node (0.73 W)
- U: Usage Factor (0.15% (default) / 0.95% (high))

The main cost formula is obtainable by means of the following equation:

$$TCO = \sum_{i=1}^{n}(Cpi * Cei) \qquad (1)$$

Where C_e can be expanded as follows:

$$Ce = t * Ch * (U * Pf + (1 - U) * Pi) \qquad (2)$$

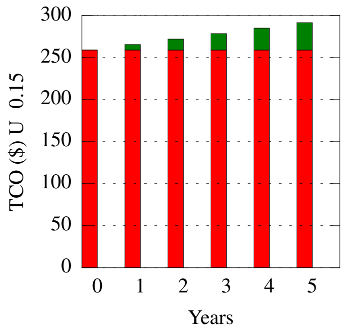

Figure 2: Accumulated Results per Year for Regular Usage Factor.

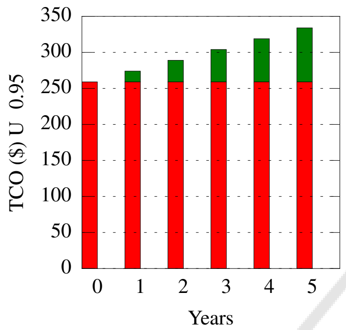

Figure 3: Accumulated Results per Year for High Usage Factor.

The results obtained from applying the given values to the previous formulas can be seen in figures 2 and 3, and show that even for a low number of boards and very high usage factors the provisioning costs (red) are constantly higher than the yearly accumulated costs (green), indicating that for the present selection a 100% full time use would still be profitable in the long run.

5 CONCLUSIONS AND FUTURE WORK

Based on all the previous points, it seems clear that the ODroid family provides superior CPU and network performance, with medium RAM and storage technology. Between both studied offerings, the U3 board is undoubtedly the best technical choice from a mere energy-based standpoint. Nonetheless, taking into consideration the initial provisioning cost objectives, the C1+, at half the cost of the U3 board, covers all our requirements in a more reasonable fashion.

In the authors' consideration the system choice as presented accomplishes the goals set in the introduction, as detailed below:

- **Cost Oriented**: Technical preferences are given a secondary plane in order to focus on flexible user budgets.
- **Simple**: Minimal software/hardware knowledge is needed to deploy the infrastructure. As a final user, and depending on the project of choice, no previous expertise is required.
- **Scalable**: Growth complexities are reduced by ensuring the use of lightweight infrastructure choices.
- **Efficient**: As shown in the previous power and cost studies, the proposed hardware infrastructure warrants near-optimal resource usage for a minimal initial inversion. For high workload scenarios, efficiency increases.

A note for future deployment could be that even if the current software offerings simplify the development of collaborative infrastructures, the worldwide expansion of Citizen Science still needs a more commonplace point of entry for the casual private user. Mobile platforms seem to be optimally placed for that role(Panchariya et al., 2015), and as such, should be given the utmost importance when considering the given infrastructure interactions.

To summarize the innovations introduced by the presented proposal:

- It adopts an inclusive view from several seemingly distant state of the art knowledge fields, as current Cloud Computing offerings are given the main focus in industry and academic-oriented works(Penzel et al., 2015)
- It subsequently reasons that the main point of interest for the individual Citizen Science user can be more clearly focused in IOT, Edge Computing initiatives(Kido and Swan, 2014) providing a direct application and immediate results.
- Finally, it expresses an integral solution oriented to give the individual enthusiast, volunteer or casual user a simple opportunity to take active participation in the scientific process with very low entry-point costs.

REFERENCES

Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., and Lindahl, E. (2015). GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25.

Anderson, D. P. (2004). Boinc: A system for public-resource computing and storage. In *Grid Comput-*

ing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on, pages 4–10. IEEE.

Barroso, L. A., Clidaras, J., and Hölzle, U. (2013). The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis lectures on computer architecture*, 8(3):1–154. 01105.

Beberg, A. L., Ensign, D. L., Jayachandran, G., Khaliq, S., and Pande, V. S. (2009). Folding@home: Lessons from eight years of volunteer distributed computing. In *IEEE International Symposium on Parallel Distributed Processing, 2009. IPDPS 2009*, pages 1–8.

Ďurech, J., Hanuš, J., and Vančo, R. (2015). Asteroids@home—A BOINC distributed computing project for asteroid shape reconstruction. *Astronomy and Computing*, 13:80–84.

Gaihre, K. P. (2015). ARM Based Computing Technology for Sustainable Development (Performance Analysis on E-Learning). In *Proceedings of IOE Graduate Conference*, pages 298–304.

Gough, C., Steiner, I., and Saunders, W. (2015). CPU Power Management. In *Energy Efficient Servers*, pages 21–70. Apress.

Hennessy, J. L. and Patterson, D. A. (2012). *Computer Architecture: A Quantitative Approach*. Elsevier.

Horowitz, M., Indermaur, T., and Gonzalez, R. (1994). Low-power digital design. In *Low Power Electronics, 1994. Digest of Technical Papers., IEEE Symposium*, pages 8–11. IEEE. 00362.

Kawrykow, A., Roumanis, G., Kam, A., Kwak, D., Leung, C., Wu, C., Zarour, E., Sarmenta, L., Blanchette, M., Waldispühl, J., and others (2012). Phylo: a citizen science approach for improving multiple sequence alignment. *PloS one*, 7(3):e31362.

Keipert, K., Mitra, G., Sunriyal, V., Leang, S. S., Sosonkina, M., Rendell, A. P., and Gordon, M. S. (2015). Energy-Efficient Computational Chemistry: Comparison of x86 and ARM Systems. *Journal of chemical theory and computation*, 11(11):5055–5061. 00001.

Kelling, S., Fink, D., La Sorte, F. A., Johnston, A., Bruns, N. E., and Hochachka, W. M. (2015). Taking a ‘Big Data’approach to data quality in a citizen science project. *Ambio*, 44(4):601–611.

Kido, T. and Swan, M. (2014). Know thyself: Data driven self-awareness for understanding our unconscious behaviors-Citizen Science Genetics for sleep and wellness research.

Korpela, E. J., Siemion, A. P., Werthimer, D., Lebofsky, M., Cobb, J., Croft, S., and Anderson, D. (2015). The next phases of SETI@ home. In *SPIE Optical Engineering+ Applications*, pages 96060B–96060B. International Society for Optics and Photonics.

Kovacs, J. and Lovas, R. (2014). Crowd computing to support EGI scientists. In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 325–331.

Laros III, J. H., Pedretti, K., Kelly, S. M., Shu, W., Ferreira, K., Vandyke, J., and Vaughan, C. (2013). Energy delay product. In *Energy-Efficient High Performance Computing*, pages 51–55. Springer. 00005.

Lawrenz, M., Shukla, D., and Pande, V. S. (2015). Cloud computing approaches for prediction of ligand binding poses and pathways. *Scientific reports*, 5.

Lencse, G. and Répás, S. (2015). Method for benchmarking single board computers for building a mini supercomputer for simulation of telecommunication systems. In *Telecommunications and Signal Processing (TSP), 2015 38th International Conference on*, pages 246–251. IEEE.

Malik, M. and Homayoun, H. (2015). Big data on low power cores: Are low power embedded processors a good fit for the big data workloads? In *Computer Design (ICCD), 2015 33rd IEEE International Conference on*, pages 379–382. IEEE. 00001.

Martens, B., Walterbusch, M., and Teuteberg, F. (2012). Costing of cloud computing services: A total cost of ownership approach. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 1563–1572. IEEE. 00050.

Newman, G., Wiggins, A., Crall, A., Graham, E., Newman, S., and Crowston, K. (2012). The future of citizen science: emerging technologies and shifting paradigms. *Frontiers in Ecology and the Environment*, 10(6):298–304.

Panchariya, N. S., DeStefano, A. J., Nimbagal, V., Ragupathy, R., Yavuz, S., Herbert, K. G., Hill, E., and Fails, J. A. (2015). Current Developments in Big Data and Sustainability Sciences in Mobile Citizen Science Applications. In *2015 IEEE First International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 202–212.

Penzel, D., Kryvinska, N., Strauss, C., and Gregu, M. (2015). The Future of Cloud Computing: A SWOT Analysis and Predictions of Development. In *2015 3rd International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 391–397.

Silvertown, J. (2009). A new dawn for citizen science. *Trends in ecology & evolution*, 24(9):467–471.

Sprinks, J., Wardlaw, J., Houghton, R., Bamford, S., and Morley, J. (2017). Task Workflow Design and its impact on performance and volunteers’ subjective preference in Virtual Citizen Science. *International Journal of Human-Computer Studies*, 104:50–63.

Werthimer, D. (2015). What’s new in SETI and CASPER? In *Radio Science Meeting (Joint with AP-S Symposium), 2015 USNC-URSI*, pages 293–293.