Clupiter: a Raspberry Pi mini-supercomputer for educational purposes

Alonso Rodríguez-Iglesias
Computer Architecture Group
CITIC, Universidade da Coruña
A Coruña, Spain
alonso.rodriguez@udc.es

María J. Martín
Computer Architecture Group
CITIC, Universidade da Coruña
A Coruña, Spain
maria.martin.santamaria@udc.es

Juan Touriño
Computer Architecture Group
CITIC, Universidade da Coruña
A Coruña, Spain
juan@udc.es

***Abstract*—The main objective of this work is to bring supercomputing and parallel processing closer to non-specialized audiences by building a Raspberry Pi cluster, called Clupiter, which emulates the operation of a supercomputer. It consists of eight Raspberry Pi devices interconnected to each other so that they can run jobs in parallel. To make it easier to show how it works, a web application has been developed. It allows launching parallel applications and accessing a monitoring system to see the resource usage when these applications are running. The NAS Parallel Benchmarks (NPB) are used as demonstration applications. From this web application a couple of educational videos can also be accessed. They deal, in a very informative way, with the concepts of supercomputing and parallel programming.**

***Index Terms*—Supercomputer, Parallel Programming, MPI, Raspberry Pi, NAS Parallel Benchmarks (NPB)**

I. Introduction

Supercomputers are computers comprised of hundreds or thousands of processors, along with massive amounts of memory, to provide high speed, computational and data processing power. However, these machines and their mechanisms and processes are often beyond the understanding of the general public. For this reason, the aim of this work is to bring this field of computer science closer to people who are not familiar with it.

For this purpose, a small cluster has been built with Raspberry Pis under the name of Clupiter (a mixture of the words Cluster and Pi), which is intended to be a small-scale replica of a supercomputer.

The Raspberry Pi is a low-cost, low-power, small-board computer commonly used in educational environments [1], [2]. Specifically, the Raspberry Pi 4B, the newest version of the *Standard* form factor that was available at the time of this work, was used to build Clupiter.

The Raspberry Pi boards have been connected and configured so that they can work collaboratively as if they were a single computer. Its performance was evaluated and a specific web application was developed to help show how supercomputers can be used to accelerate the execution of computationally intensive applications.

The paper is structured as follows. In section II we give a brief summary of related work. In section III we outline the requirements analysis performed, as well as the hardware and software design decisions and the configuration process of the resulting infrastructure. A performance evaluation, using the NAS Parallel Benchmarks (NPB) [3], is carried out in section IV. Section V describes the web application developed to help explain the internal operation of Clupiter. The article ends with the main conclusions.

II. Related work

Raspberry Pi boards include all the essential circuits, such as CPUs (Central Processing Unit), GPUs (Graphical Processing Unit) and input/output circuits, making them well suited for use in computer-related educational projects. They have been used, for example, to teach topics such as image processing [4], signal processing [5], real-time control algorithms [6], or even cybersecurity [7].

More closely related to the topic of this article, there are previous experiences in the literature where several Raspberry Pis have been used to build low-cost, low-power clusters to help understand concepts related to high-performance computing. Examples include the *Iridis-pi* cluster [8], consisting of 64 Raspberry Pis connected via Ethernet and housed in a chassis built from Lego blocks; the *Wee Archie* infrastructure (https://www.archer2.ac.uk/community/outreach/materials/wee_archie), built with 18 Raspberry Pi boards packaged in a transparent box, and intended to be the scaled-down version of the EPCC (Edinburgh Parallel Computing Center) ARCHER2 supercomputer; or the *cluster coffer* project [9], where 16 Raspberry Pis are housed in a portable metal case and interconnected by a Gigabit Ethernet network.

Previous proposals differ fundamentally in the final appearance of the cluster and the monitoring information that can be obtained during the execution of parallel applications. In this sense, Clupiter's key strengths are: its organization, since its hardware has been assembled to emulate the structure of a real supercomputer; and its web application, which allows the execution and monitoring of all NPB applications in a simple and visual way.

Outside of the educational field, Raspberry Pi clusters have been used for a number of different applications. To mention a few examples, in [10] they are used to run data mining algorithms, in [11] to speed up the execution of a parallel

TABLE I: Total cost of Clupiter

Units	**Material**	**Cost (€)**
8	Raspberry Pi 4B	392.00
1	Gigabit Switch	33.60
1	USB Ethernet	23.00
2	RPI Towers	54.00
1	120mm Fan	18.00
1	Power Supply	27.00
8	32GB MicroSD	104.00
10	Magnetic USB-C cables	26.16
1	MT3608 Step-up	1.79
Total		**679.55**

watermarking algorithm, or in [12] to monitor agricultural land.

III. Clupiter Design

Before deciding on the hardware and software components of our cluster, a requirements analysis was carried out and the following conclusions were drawn:

- The cluster must be **small** and **manageable**, discarding structures where nodes are "free" and "dispersed".
- It must be **visually pleasing** and **comprehensible**, with easily identifiable parts, as isolated and distinguishable as possible.
- All nodes must be connected to each other in an N-to-N topology, i.e., the typical internal structure of an Ethernet switch.
- The cluster must be able to run generic **parallel applications** that make use of all nodes.
- It must be **reasonable**, with quantity and quality of materials adequate to its expectations, and using modern components with a **good value for money**.

A description of the hardware and software components that have been selected to meet these requirements and their configuration is given below.

A. Hardware components

Table I summarizes the hardware components used and their cost in euros (excluding VAT).

Eight Raspberry Pi 4B devices have been chosen as cluster nodes. The CPU of this model is the Broadcom BCM2711, a processor with ARMv8-A architecture and 4 Cortex-A72 cores, running at 1.5 GHz and backed by 2 GB of memory. The choice of this board was based on its small form factor and low-power consumption and cost, which makes it an ideal solution for this work, as it does not require very powerful hardware, but rather a lot of low-power hardware to simulate the structure of a supercomputer. A MicroSD card was purchased for each Raspberry Pi, which is needed to store the operating system.

Cluster components receive the necessary power for operation from a 5V, 30A power supply, resulting in 150W of power, which is three times the minimum power requirements, and twice as much as recommended.

To connect the nodes, an 8-port Gigabit Ethernet switch operating at 5V has been used, which allows it to be connected directly to the power supply. It should be noted that the cluster must be accessible from the outside, so we actually need nine ports. To solve this drawback, a USB 3.0 to Gigabit Ethernet adapter is added, which is connected to the Cluster master node and bridged with the internal interface, thus creating a virtual 9-port switch (see Figure 1).

The Raspberry Pi boards are stacked vertically in two towers of 4 units each, so that they can be easily wired, accessed and cooled. Between the two towers sits a 120mm fan, which while not essential for heat dissipation purposes, serves as an interesting analogy to the importance of cooling in real supercomputers. The chosen fan operates at 12V; however, feeding it at this voltage would cause it to run at full power constantly. To avoid this, a variable step-up is used to keep the fan at a fixed low speed to reduce the noise level. The electrical connection of these components can be seen in Figure 2.

The final result of the assembly of all these hardware modules can be seen in the render in Figure 3. In that image, the multiple zones of the chassis can be identified. Starting with the bottom area, we have the power supply and the DC and AC cable connections. The switch, which interconnects the eight devices with each other at 1 Gbps in full duplex mode, is located in the upper area. The two Raspberry Pi towers are placed in the middle area, oriented with the input/output interfaces facing outwards. They are cooled by the fan running through them. Figure 4 shows an actual picture of Clupiter and its connections.

B. Software components

The Arch Linux operating system was installed on the hardware described in the previous section. It is a lightweight and flexible Linux distribution with extensive documentation on the Arch Wiki (https://wiki.archlinux.org).

In order to run parallel codes on Clupiter we will use MPI (Message Passing Interface) [13], the *de facto* standard for programming distributed-memory parallel systems. A parallel MPI program consists of several processes, each one with associated local memory, that can communicate through the interconnection network by using send and receive routines. In this work we use a free implementation of MPI, OpenMPI (https://www.open-mpi.org), since it is the official supported library provided by Arch Linux in its repositories.

In addition, a web application was developed to monitor the status and historical data of the cluster in real time, as well as to provide interactive explanations about the operation of the MPI programs. This web application will be explained in detail in Section V.

C. Cluster configuration

In order to manage the cluster, each node must be assigned an IP address. Due to its characteristics, and especially to the fact that the connection from the outside will be variable and not always available, static IP addresses are configured on each

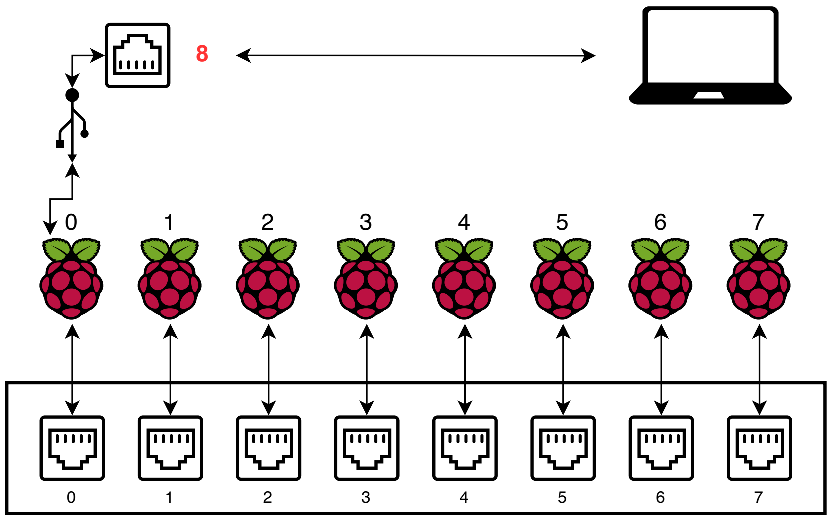


Fig. 1: Clupiter physical network diagram

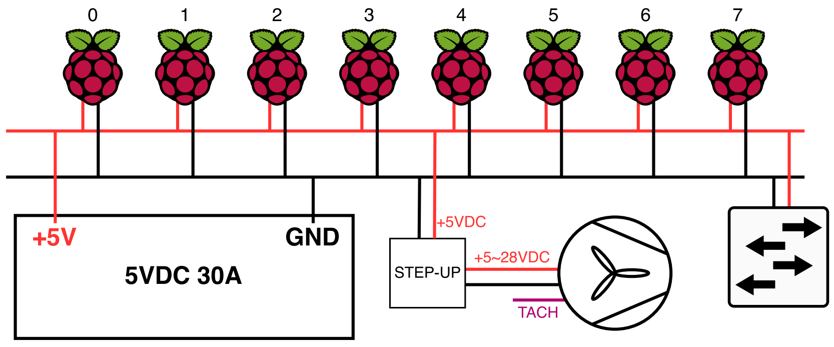


Fig. 2: Clupiter electrical diagram

of the nodes, as well as a default gateway address, which will sometimes be active and sometimes not.

In order to launch MPI applications, a user account named `mpiuser` is created. In addition, non-interactive authentication is configured for SSH by using a public and private key pair. Non-interactive authentication via private key is required so that the `mpiuser` can establish unattended SSH connections to the other nodes during the execution of MPI programs.

Finally, since calls to the `mpirun` command for running MPI jobs on multiple hosts execute the same command on all of them, there must be some kind of shared storage mounted under the same path on all the hosts. To satisfy this need for shared storage, NFS (Network File System) is chosen. The server will be the master node and all other nodes will be clients.

The entire configuration process is documented in more detail in [14], where all the Linux commands used for this purpose are specified.

IV. Performance Evaluation

Once the cluster is up and running, we test its ability to run MPI applications and its performance. Although performance is not a priority, it is convenient to perform these tests, especially to be able to observe the impact of the communication network between the cores of a single CPU (remember that each CPU has four cores), or between multiple CPUs and memories. This is done by using the NPB benchmarks [3], a set of numerical computational kernels designed by the NASA's Supercomputing Division for measuring supercomputer performance. Figures 5, 6 and 7 show the achieved results, expressed in millions of operations per second (MOPS), for a representative subset of the NPB applications. The results obtained with the other NPB applications can also be found at [14].

Specifically, results are shown for:

- CG (Conjugate Gradient): It solves systems of linear equations of symmetric and positive definite matrices using the conjugate gradient iterative method.

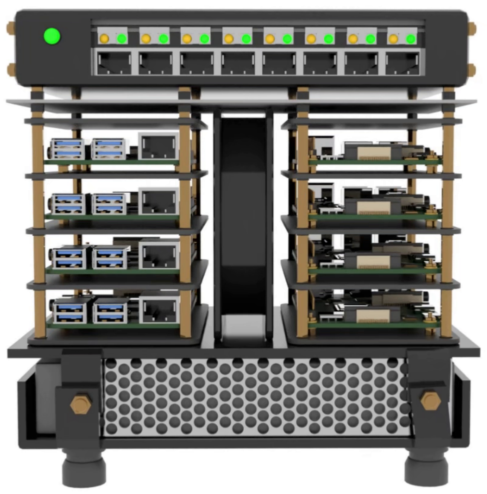


Fig. 3: Clupiter infographic

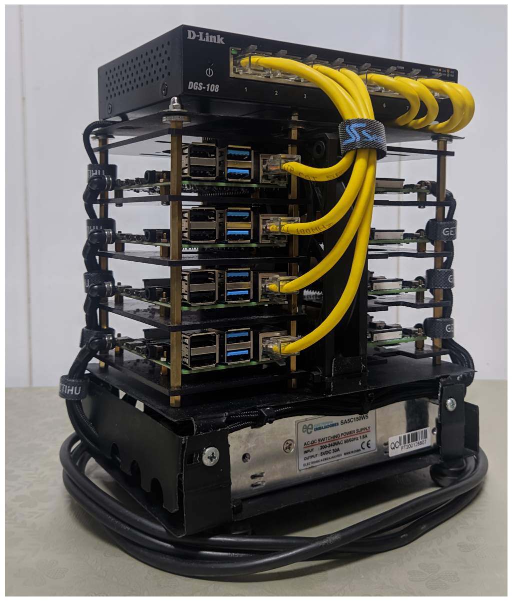


Fig. 4: Photo of Clupiter and its connections

- EP (Embarrassingly Parallel): It contains a massively parallel kernel that serves to provide an estimate of floating-point performance bounds.
- IS (Integer Sort): It sorts integer number and it is useful to test both the speed of computation with integers and the performance of communications.

The three applications were run using the class `C` size. Five runs of each were performed and the average number of MOPS was calculated. The dashed vertical line in the graphs shows the transition from single-node execution (using all four cores) to multi-node execution. Intra-node communications are performed via shared memory, while inter-node communications use the Gigabit Ethernet network.

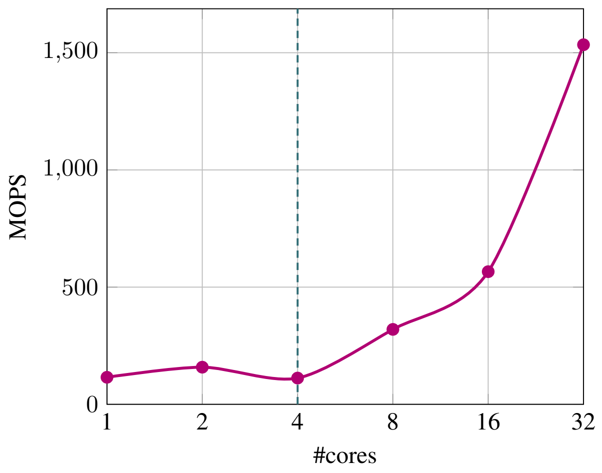


Fig. 5: Performance for the CG kernel

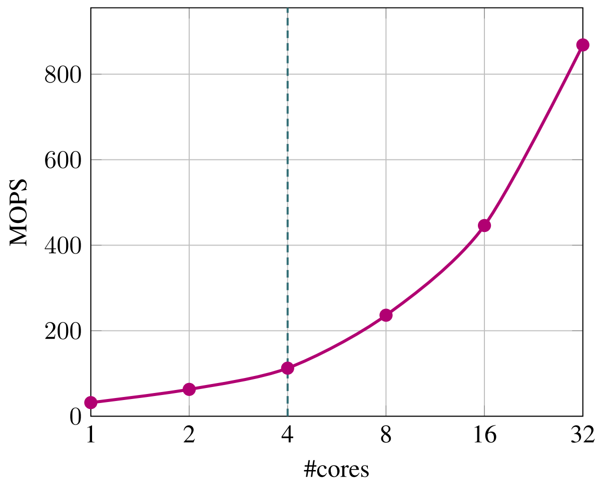


Fig. 6: Performance for the EP kernel

As it can be seen, the results are excellent for EP (see Figure 6) which, by its nature, allows obtaining a very good floating-point performance. On the other hand, there is a noticeable impact when running benchmarks that make intensive use of inter-node communications. This can be observed in the IS graph of Figure 7, where performance drops sharply when running in a two-node configuration (i.e., eight cores). As for CG (Figure 5), it can be seen that not only performance is not improved by going from two to four cores, but it actually gets worse. This effect is due to the small bandwidth offered by the single memory chip built into each Raspberry Pi. Despite this behavior on single-node runs, performance does increase when running kernels in distributed memory. In some cases, even more than doubling the previous performance, as it occurs when going from 16 to 32 cores in CG and EP.

V. Web application

The main objective of this work is to build a small-scale replica of a supercomputer that can be used to explain the

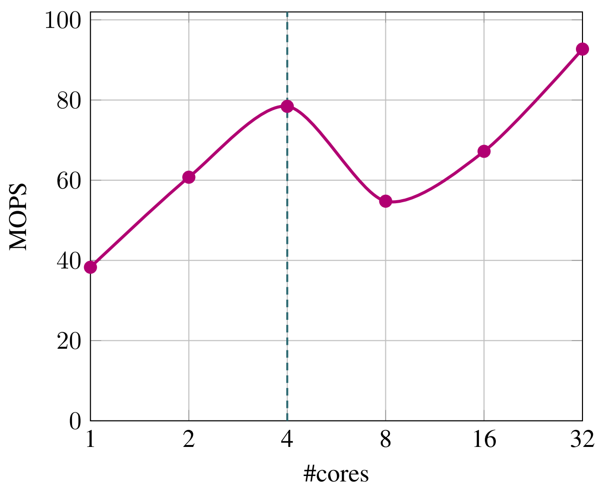


Fig. 7: Performance for the IS kernel

operation of a parallel system, as well as to carry out practical demonstrations of its operation, both live and through videos. For this purpose, a web application has been developed from which the benchmarks described in the previous section can be run and monitored, and videos about parallel computing can be watched.

We have chosen to implement this functionality as a web application because it offers universality and simplicity (the web application can be accessed from any operating system and any browser). To run this application, it will only be necessary to connect through a browser to the address of the master node, which is the one that hosts the application.

The technology on which the backend of the web application is programmed is *nodejs*, in particular, *expressjs* (https://expressjs.com/). Moreover, it makes use of the APIs of *Netdata* and *socket.io*. The choice of *socket.io* is due to the simplicity and elegance it brings to the code. Using *socket.io* we can make calls to the backend from the web application in the same programming style as in Android, making it familiar and straightforward. On the other hand, Netdata [15] is a free and open source monitoring software, which is easily configured and integrated, and allows obtaining information from the Clupiter nodes in real time. To use the data provided by this software, it must first be installed and activated on each of the cluster nodes.

The web application has three tabs: “Home”, “Monitoring” and “About”. From the “Home” tab two educational videos, created specifically for this project, can be accessed. The first one lasts about 6 minutes and introduces, in a very informative way, the concept of a supercomputer, providing historical context and figures about the capacity of these infrastructures, as well as their utilities and achievements. This is followed by a description of Clupiter and its fundamental parts, which are related to their counterparts in a real supercomputer. The second video, a more technical one, lasts about seven minutes and introduces, through hand-illustrated animations, the inner workings of a supercomputer, the need for them on a day-to-day basis, and the constraints that parallel programs have to overcome, using simple examples and avoiding technicalities. This is followed by a brief introduction to the most basic MPI operations. Both videos have also been hosted and subtitled in both English and Spanish on the YouTube platform. The videos can be played from the following URLs: https://youtu.be/o76-VP6WFCo and https://youtu.be/if0MWI_9xzM.

The “Monitoring” tab provides access to a page from which each of the NPB benchmarks (class `C`) can be run, namely LU, CG, FT, IS, MG and EP. The page shows the real-time impact that the execution of the benchmark has on the load and network traffic of each node, and its output is displayed on a terminal. Figure 8 shows an english translated version of this monitoring page. At the top, there are the CPU usage and network traffic meters for each of the Raspberry Pi boards, while at the bottom there are the buttons to run the different test programs, allowing us to observe in real time their impact on the different nodes. Specifically, the figure shows the output during the execution of the FT (Fast Fourier Transform) kernel. Thanks to this monitorization, it can be observed that during the execution of this application all nodes work in parallel, and the communication and computation phases alternate.

Finally, the “About” tab includes, among other information, a brief description of Clupiter.

VI. Conclusions

This paper has presented Clupiter, a miniature supercomputer with all its sections well differentiated and extrapolated to a real supercomputer.

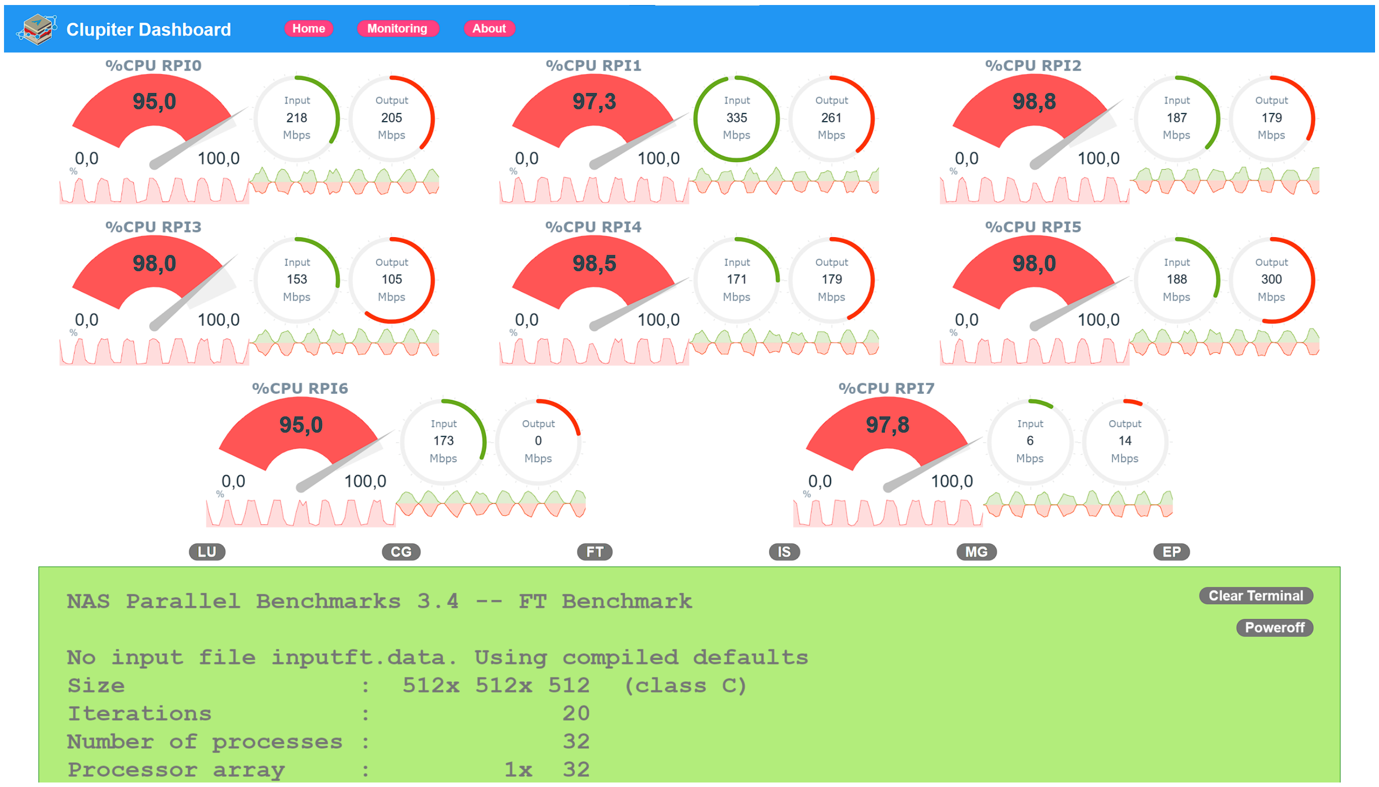
Although performance and scalability are not good (note that Raspberry Pi boards are not designed to be particularly efficient computers in this regard), Clupiter is able to run parallel programs and test how the different hardware components of the system can work together to accelerate the execution of scientific and engineering applications. That was, after all, the ultimate goal of this project.

Clupiter is currently available at the ”Technology Demonstrator Room” of the Center for Information and Communications Technology Research (CITIC, https://citic.udc.es/en/technology-demonstrator/) of the Universidade da Coruña as an educational tool. It is used during visits to the center by, among other audiences, high school students interested in technology and engineering, in order to engage them in the supercomputing arena.

All resources related to this work, including the source code of the web application, are available under the MIT license at https://github.com/forcegk/GEI_TFG.

Acknowledgments

Clupiter has been supported by grants EDC431C 2021/30 (Xunta de Galicia, Consolidation Program of Competitive Reference Groups) and PID2022-136435NB-I00, funded by MCIN/AEI/ 10.13039/501100011033 and by “ERDF A way of making Europe”, EU.



```
NAS Parallel Benchmarks 3.4 -- FT Benchmark

No input file inputft.data. Using compiled defaults
Size                :  512x 512x 512  (class C)
Iterations          :            20
Number of processes :            32
Processor array     :        1x  32
```

Fig. 8: Clupiter monitoring page

References

[1] B. Balon and M. Simić, "Using Raspberry Pi computers in education," in *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019, pp. 671–676.

[2] N. S. Yamanoor and S. Yamanoor, "High quality, low cost education with the Raspberry Pi," in *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, 2017, pp. 1–5.

[3] NASA, "NAS Parallel Benchmarks," https://www.nas.nasa.gov/software/npb.html.

[4] J. Marot and S. Bourennane, "Raspberry Pi for image processing education," in *25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 2364–2366.

[5] G. Pasolini, A. Bazzi, and F. Zabini, "A Raspberry Pi-based platform for signal processing education [SP Education]," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 151–158, 2017.

[6] J. Sobota, R. PiŜl, P. Balda, and M. Schlegel, "Raspberry Pi and Arduino boards in control education," *IFAC Proceedings Volumes*, vol. 46, no. 17, pp. 7–12, 2013.

[7] P. Legg, A. Mills, and I. Johnson, "Teaching offensive and defensive cyber security in schools using a Raspberry Pi cyber range," *Journal of The Colloquium for Information Systems Security Education*, vol. 10, no. 1, 2023, 9 pages.

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

[9] P. Gschwandtner, A. Hirsch, P. Thoman, P. Zangerl, H. Jordan, and T. Fahringer, "The cluster coffer: teaching HPC on the road," *Journal of Parallel and Distributed Computing*, vol. 155, pp. 50–62, 2021.

[10] J. Saffran, G. Garcia, M. A. Souza, P. H. Penna, M. Castro, L. F. Góes, and H. C. Freitas, "A low-cost energy-efficient Raspberry Pi cluster for data mining algorithms," in *Euro-Par 2016: Parallel Processing Workshops, Lecture Notes in Computer Science, vol. 10104*. Springer, 2017, pp. 788–799.

[11] K. M. Hosny, A. Magdi, N. A. Lashin, O. El-Komy, and A. Salah, "Robust color image watermarking using multi-core Raspberry Pi cluster," *Multimedia Tools and Applications*, vol. 81, no. 12, pp. 17 185–17 204, 2022.

[12] A. Aher, J. Kasar, P. Ahuja, and V. Jadhav, "Smart agriculture using clustering and IOT," *International Research Journal of Engineering and Technology*, vol. 5, no. 03, pp. 4065–4068, 2018.

[13] Message Passing Interface Forum, "MPI: A message-passing interface standard version 4.0," https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf, 2021.

[14] A. Rodríguez-Iglesias, "Hardware and software implementation of a Raspberry Pi-based mini-supercomputer (in Spanish)," https://github.com/forcegk/GEI_TFG/releases/download/memoria/MemoriaTFG.pdf, 2021, B.S. Thesis in Computer Engineering, Universidade da Coruña.

[15] Netdata, Inc., "Monitoring everything in real time for free," https://www.netdata.cloud.