

# Clúpiter: un mini-supercomputador de Raspberry Pis con fines divulgativos

Alonso Rodríguez-Iglesias, María J. Martín y Juan Touriño<sup>1</sup>

*Resumen*— El principal objetivo de este trabajo es acercar la supercomputación y el procesamiento paralelo a públicos no especializados mediante la construcción de un clúster formado por Raspberry Pis, al que hemos llamado Clúpiter, que emula el funcionamiento de un supercomputador. Está formado por ocho nodos, ocho Raspberry Pis interconectadas entre sí para que puedan ejecutar trabajos en paralelo. Para que sea más sencillo mostrar cómo funciona, se ha desarrollado una aplicación web desde la que se pueden lanzar aplicaciones paralelas y acceder a un sistema de monitorización que permite ver el uso de los recursos cuando se están ejecutando. Desde dicha aplicación web también se puede acceder a un par de vídeos educativos que tratan de manera muy divulgativa los conceptos de supercomputación y programación paralela.

*Palabras clave*— Supercomputador, programación paralela, MPI, Raspberry Pi

## I. INTRODUCCIÓN

Los supercomputadores son equipos informáticos que están compuestos por cientos o miles de procesadores, junto con cantidades ingentes de memoria, para ofrecer una elevada velocidad y capacidad de cálculo y de procesamiento de datos. Sin embargo, estas máquinas y sus mecanismos y procesos suelen quedar muchas veces fuera de la comprensión del público general. Por eso, el objetivo de este trabajo es acercar este ámbito de la informática a personas ajenas al mismo.

Para ello, se ha construido un pequeño clúster con Raspberry Pis con el nombre de Clúpiter (mezcla de las palabras Clúster y Pi) que pretende ser una réplica a pequeña escala de un supercomputador.

La Raspberry Pi es un computador de placa reducida de bajo coste y consumo usado muy habitualmente en entornos educativos [1], [2]. En concreto, para la construcción de Clúpiter se ha utilizado la Raspberry Pi 4B, la versión más nueva del formato *Standard* que estaba disponible en el momento de la realización de este trabajo.

Las placas de Raspberry Pis se han conectado y configurado para que puedan trabajar de forma colaborativa como si se tratase de un único equipo, se ha evaluado su rendimiento y se ha desarrollado una aplicación web específica que ayuda a demostrar cómo se pueden utilizar los supercomputadores para acelerar la ejecución de aplicaciones computacionalmente costosas.

El trabajo está estructurado de la siguiente manera. En la sección II se hace un breve resumen de los trabajos relacionados. En la sección III se expone el

análisis de requisitos realizado, así como las decisiones de diseño hardware y software y el proceso de configuración de la infraestructura resultante. En la sección IV se lleva a cabo una evaluación de rendimiento, utilizando para ello los NAS Parallel Benchmarks (NPBs) [3]. La sección V describe la aplicación web desarrollada para ayudar a explicar el funcionamiento de Clúpiter. El artículo finaliza exponiendo las conclusiones.

## II. TRABAJO RELACIONADO

Las placas de Raspberry Pis incluyen todos los circuitos esenciales, tales como CPUs (*Central Processing Unit*), GPUs (*Graphical Processing Unit*) y circuitos de entrada y salida, lo que las hacen muy apropiadas para ser utilizadas en proyectos educativos relacionados con la informática. Se han utilizado, por ejemplo, para enseñar tópicos como el tratamiento de imágenes [4], el procesamiento de señal [5], algoritmos de control en tiempo real [6] o ciberseguridad [7].

Más vinculado con la temática de este artículo, existen en la bibliografía experiencias previas en las que se han utilizado varias Raspberry Pis para construir clústeres de bajo coste y consumo que ayuden a entender los conceptos relacionados con la computación de altas prestaciones. Algunos ejemplos son el clúster *Iridis-pi* [8], que consiste en 64 Raspberry Pis conectadas a través de Ethernet y alojadas en un chasis construido con bloques Lego; la infraestructura *Wee Archie* ([https://www.archer2.ac.uk/community/outreach/materials/wee\\_archie](https://www.archer2.ac.uk/community/outreach/materials/wee_archie)), construida con 18 Raspberry Pis empaquetadas en una caja transparente y que pretende ser la versión reducida del supercomputador del EPPC (Edinburgh Parallel Computer Center) ARCHER2; o el proyecto *cluster coffee* [9], 16 Raspberry Pis ubicadas en un maletín metálico portátil e interconectadas por una red Gigabit Ethernet.

Las propuestas anteriores se diferencian fundamentalmente en el aspecto final del clúster y la información de monitorización que se puede obtener durante la ejecución de aplicaciones paralelas. En este sentido, los puntos fuertes de Clúpiter son dos: su organización, ya que su hardware ha sido ensamblado para emular la estructura de un supercomputador real; y su aplicación web, que permite la ejecución y monitorización de forma sencilla y visual de todas las aplicaciones de los NAS Parallel Benchmarks (NPBs) [3].

Fuera del ámbito educativo, los clústeres de Raspberry Pis han sido utilizados para un gran número de

<sup>1</sup>Universidade da Coruña, CITIC, Grupo de Arquitectura de Computadores, e-mail: {alonso.rodriguez,mariam,juan}@udc.es.

Tabla I: Coste total de Clúpiter

Uds.	Material	Coste (€)
8	Raspberry Pi 4B	392,00
1	Switch Gigabit	33,60
1	USB Ethernet	23,00
2	Torres RPI	54,00
1	Ventilador	18,00
1	Fuente alimentación	27,00
8	MicroSD 32GB	104,00
10	Cables USB-C magnéticos	26,16
1	Step-up MT3608	1,79
<b>Total</b>		<b>679,55</b>

aplicaciones diferentes. Por mencionar algunos ejemplos, en [10] se utilizan para ejecutar algoritmos de minería de datos, en [11] para acelerar la ejecución de un algoritmo de marca de agua paralelo, o en [12] para monitorizar terrenos agrícolas.

### III. DISEÑO DE CLÚPITER

Antes de decidir los componentes hardware y software del clúster se lleva a cabo un análisis de requisitos llegando a las siguientes conclusiones:

- Debe ser **pequeño** y **manejeable**, descartando estructuras donde los nodos queden “libres” y “desperdigados”.
- Debe ser **visualmente agradable** y **comprensible**, con partes fácilmente identificables, lo más aisladas y señalables posible.
- Todos los nodos deben estar conectados entre sí en una **topología N a N**, es decir, la típica de un switch Ethernet.
- Debe ser capaz de ejecutar **aplicaciones MPI** [13] genéricas.
- Debe ser sensato, empleando una calidad y cantidad de materiales adecuada a las expectativas del mismo y utilizando componentes actuales con una **buena relación calidad/precio**.

A continuación se describen los componentes hardware y software que se han seleccionado para cumplir con esos requerimientos y la configuración que se ha llevado a cabo de los mismos.

#### A. Componentes hardware

La Tabla I resume los componentes hardware empleados y su coste (sin IVA).

Se ha elegido como nodos del clúster ocho Raspberry Pis 4B. La CPU de este modelo es la Broadcom BCM2711, un procesador con arquitectura ARMv8-A y 4 núcleos Cortex-A72, que funcionan a 1,5 GHz y cuentan con 2 GB de memoria. La elección de esta placa se ha basado en su reducido formato y su bajo consumo y coste, lo que lo hace una solución idónea para este trabajo, que no requiere de hardware muy potente, sino más bien de mucho hardware poco potente que permita simular la estructura de un supercomputador. Para cada Raspberry Pi se ha adquirido una tarjeta MicroSD, necesaria para almacenar el sistema operativo.

Los componentes del clúster reciben la energía necesaria para su funcionamiento de una fuente de alimentación de 5V y 30A, resultando en una potencia de 150W, que triplica los requisitos energéticos mínimos, y duplica los recomendados.

Para conectar los nodos se ha utilizado un switch Gigabit Ethernet de 8 puertos que funciona a 5V, lo que permite conectarlo directamente a la fuente de alimentación. Hay que tener en cuenta que el clúster debe poder ser accedido desde el exterior, por lo que en realidad se necesitan 9 puertos. Para solucionar este inconveniente se añade un adaptador USB 3.0 a Gigabit Ethernet que se conecta al nodo maestro de Clúpiter y es puenteado con la interfaz interna, creando así un switch virtual de 9 puertos (ver Figura 1).

Las Raspberry Pis se apilan verticalmente en dos torres de 4 placas cada una para poder ser cableadas, accedidas y refrigeradas de forma sencilla. Entre las dos torres se ubica un ventilador de 120 mm, que si bien no es imprescindible a efectos de disipación de calor, es interesante para realizar analogías con la importancia de la refrigeración en los supercomputadores reales. El ventilador elegido funciona a 12V, sin embargo, alimentarlo a esa tensión haría que funcionase a máxima potencia constantemente. Para evitar esto, se emplea un *step-up* variable con el que se mantiene el ventilador a una velocidad fija y lo más baja posible para rebajar el nivel de ruido. La conexión eléctrica de estos componentes se puede observar en la Figura 2.

El resultado final del ensamblaje de todos estos módulos hardware se puede observar en el *render* de la Figura 3. Sobre esa imagen se pueden identificar las múltiples zonas del chasis. Comenzando por la zona inferior tenemos la fuente de alimentación y las conexiones de los cables de corriente continua y alterna. En la zona superior se encuentra el switch, que interconecta los ocho dispositivos entre sí a 1 Gbps en modo *full duplex*. La zona intermedia está ocupada por las dos torres de Raspberry Pis, orientadas con las interfaces de entrada/salida hacia fuera, y que son refrigeradas por el ventilador que las atraviesa. La Figura 4 muestra una foto real de Clúpiter y sus conexiones.

#### B. Componentes software

Sobre el hardware descrito en la sección anterior se instaló el sistema operativo Arch Linux, una distribución de Linux ligera y flexible que dispone de una amplia documentación en la Arch Wiki (<https://wiki.archlinux.org>).

Para poder ejecutar códigos paralelos sobre Clúpiter utilizaremos MPI (Message Passing Interface) [13], el estándar de facto para la programación de los sistemas paralelos de memoria distribuida. En este trabajo se emplea una implementación libre de MPI, OpenMPI, ya que es la librería MPI que ofrece Arch Linux en sus repositorios oficiales.

Adicionalmente, se desarrolló una aplicación web para monitorizar el estado e histórico del clúster en

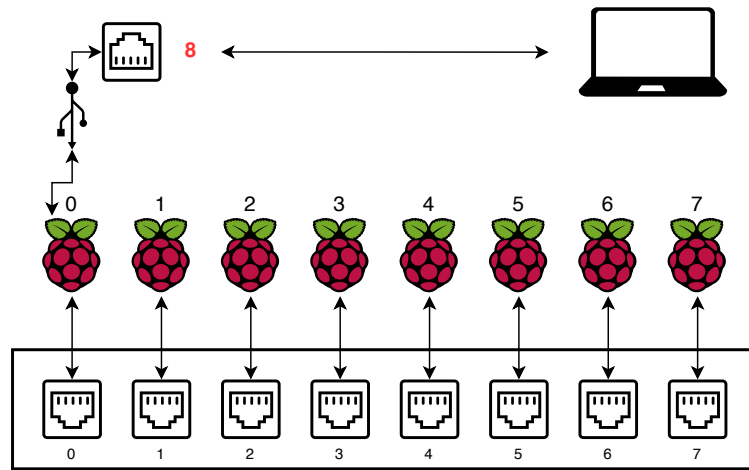


Fig. 1: Esquema físico de red de Clúptiter

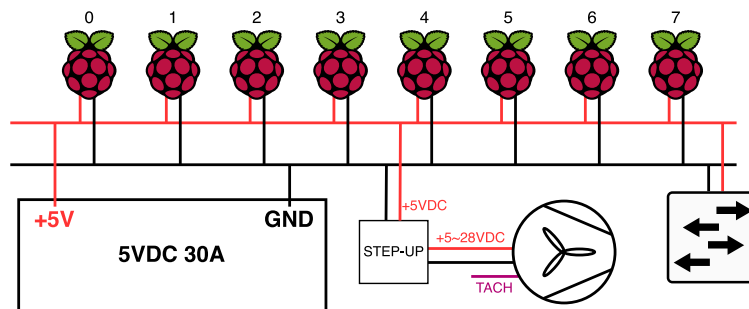


Fig. 2: Diagrama eléctrico de Clúptiter

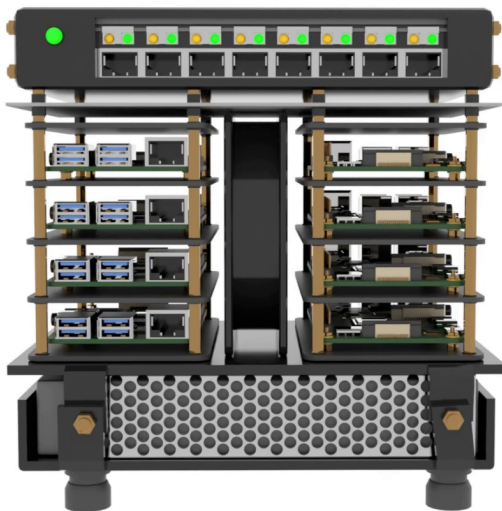


Fig. 3: Infografía de Clúptiter



Fig. 4: Foto de Clúptiter y sus conexiones

tiempo real, y poder realizar explicaciones interactivas acerca del funcionamiento de los programas MPI. Esta aplicación web se explicará en detalle en la Sección V.

### C. Configuración del clúster

Para poder administrar el clúster, es necesario que a cada nodo se le asigne una dirección IP. Debido a las características del mismo, y especialmente a que la conexión desde el exterior va a ser variable y no siempre va a estar disponible, se configuran direcciones IP estáticas en cada uno de los nodos, así como

una dirección de *gateway* predeterminada, que algunas veces estará activa y otras no.

Para poder lanzar aplicaciones MPI se crea una cuenta de usuario denominada `mpiuser`. Además, se configura la autenticación no interactiva para SSH mediante el uso del par de claves pública y privada. La autenticación no interactiva por clave privada es necesaria para que el usuario `mpiuser` pueda establecer conexiones SSH de forma desatendida con el resto

de nodos durante la ejecución de programas MPI.

Finalmente, debido a que las llamadas al comando `mpirun` para la ejecución de trabajos MPI en múltiples nodos ejecutan el mismo comando en todos ellos, debe existir algún tipo de almacenamiento compartido montado con el mismo nombre en todos los nodos. Para satisfacer esta necesidad de almacenamiento compartido elegimos NFS (*Network File System*). El servidor será el nodo maestro y el resto serán clientes.

En [14] se encuentra documentado en mayor detalle todo el proceso de configuración, especificando todos los comandos Linux utilizados para ello.

#### IV. EVALUACIÓN DEL RENDIMIENTO

Una vez puesto en marcha el clúster, se comprueba su capacidad para ejecutar aplicaciones MPI y su rendimiento. Si bien el rendimiento no es una prioridad, es conveniente realizar estas pruebas, especialmente para poder observar el impacto que tiene la red de comunicaciones entre los núcleos de una sola CPU (recordemos que cada una tiene 4 núcleos), o entre múltiples CPUs y memorias. Para ello se utilizan los NAS Parallel Benchmarks (NPBs) [3], un conjunto de tests de cálculo numérico diseñados por la División de Supercomputación de la NASA para la medida del rendimiento de supercomputadores. Las Figuras 5, 6 y 7 muestran los resultados obtenidos, expresados en millones de operaciones por segundo (MOPS), para un subconjunto representativo de las aplicaciones. Los resultados obtenidos con las otras aplicaciones de los NPBs se pueden consultar en [14].

En concreto, se muestran los resultados para:

- CG (Conjugate Gradient): Resuelve sistemas de ecuaciones lineales de matrices simétricas y definidas positivas utilizando el método iterativo del gradiente conjugado.
- EP (Embarrassingly Parallel): Contiene un *kernel* masivamente paralelo que sirve para proporcionar una estimación de los límites del rendimiento en punto flotante.
- IS (Integer Sort): Ordena números enteros. Es útil para comprobar tanto la velocidad de cómputo con enteros como el rendimiento de las comunicaciones.

Las tres aplicaciones se han ejecutado con clase C, se han realizado 5 ejecuciones de cada una y se ha calculado la media. La línea vertical discontinua de las gráficas señala la transición de ejecución en un solo nodo (utilizando los 4 núcleos) a ejecución en múltiples nodos. Las comunicaciones intranodo se realizarán vía memoria compartida, mientras que las comunicaciones internodo utilizarán la red Gigabit Ethernet.

Como se puede apreciar, los resultados son excelentes para EP que, por su naturaleza, permite obtener un muy buen rendimiento en punto flotante. Por otro lado, se nota un fuerte impacto al ejecutar benchmarks que hacen un uso intensivo de las comunicaciones entre nodos. Esto se puede apreciar en

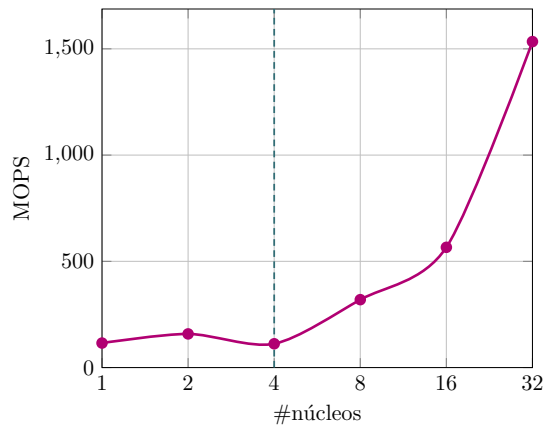


Fig. 5: Rendimiento para el kernel CG

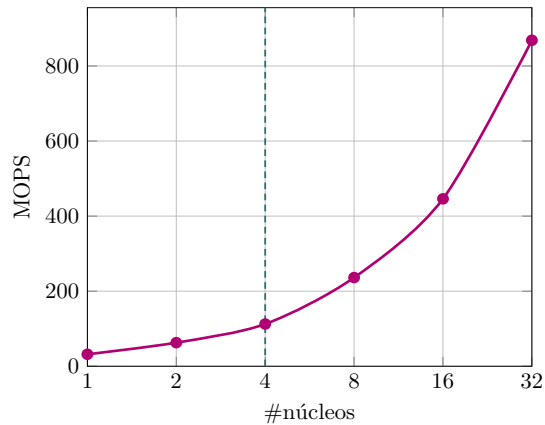


Fig. 6: Rendimiento para el kernel EP

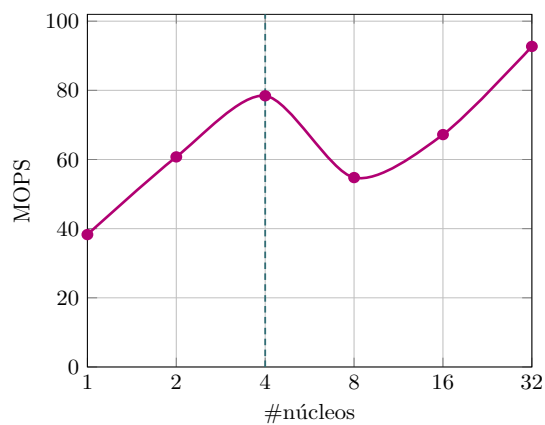


Fig. 7: Rendimiento para el kernel IS

la gráfica de IS, donde el rendimiento cae de forma pronunciada en la ejecución con dos nodos (es decir, 8 núcleos). En cuanto a CG, se puede ver que no solo no se obtiene un mayor rendimiento pasando de dos a cuatro núcleos, sino que se pierde. Este efecto es debido al pequeño ancho de banda que ofrece el único chip de memoria de cada Raspberry Pi. A pesar de este comportamiento en las ejecuciones en un único nodo, cuando se ejecutan los kernels en memoria distribuida, el rendimiento sí que aumenta. En algunos casos, incluso, más que duplicando el rendimiento anterior, como por ejemplo ocurre al pasar de 16 a 32 núcleos en CG y EP.

## V. APLICACIÓN WEB

El objetivo final de este trabajo es construir una réplica a pequeña escala de un supercomputador que sirva para explicar el funcionamiento de un sistema paralelo, así como realizar demostraciones prácticas, tanto en directo como mediante vídeos, de su funcionamiento. Para ello se ha desarrollado una aplicación web desde la cual se podrán ejecutar y monitorizar los benchmarks discutidos en la sección anterior, así como visualizar vídeos divulgativos sobre la computación paralela.

Se ha elegido implementar esta funcionalidad como una aplicación web porque proporciona universalidad (nos podremos conectar desde cualquier operativo) y sencillez. Para ejecutarla solamente será necesario conectarse mediante un navegador a la dirección del nodo maestro, que es el que aloja la aplicación.

La tecnología sobre la que se programa el *backend* de la aplicación web es *nodejs*, en concreto *expressjs* (<https://expressjs.com/es/>), y se hace uso de las APIs de *Netdata* y *socket.io*. La elección de *socket.io* se realiza debido a la sencillez y elegancia que aporta al código, ya que desde la propia aplicación web se pueden hacer llamadas al *backend* con el mismo estilo de programación que en Android, resultando familiar y sencillo. Por otro lado, *Netdata* [15] es un software de monitorización gratuito y de código abierto, de fácil configuración e integración, que permite obtener información de los nodos de Clúptiter en tiempo real. Para emplear los datos que proporciona este software, primero debe instalarse y activarse en cada uno de los nodos del clúster.

La aplicación web dispone de 3 pestañas: “Inicio”, “Monitorización” y “Acerca de”. Desde la pestaña de “Inicio” se puede acceder a dos vídeos didácticos desarrollados específicamente para este proyecto. El primero de ellos dura unos 6 minutos e introduce, de manera muy divulgativa, el concepto de supercomputador, aportando contexto histórico y cifras acerca de la capacidad de estas infraestructuras, así como de sus utilidades y logros. Tras ello, se realiza una descripción de Clúptiter y sus partes fundamentales, y se relacionan con sus análogos en un supercomputador real. El segundo vídeo, ya más técnico, dura unos 7 minutos y en él se introduce, mediante animaciones ilustradas a mano, el funcionamiento interno de un supercomputador, su necesidad en el día a día, y las restricciones que tienen que superar los programas paralelos, utilizando ejemplos sencillos y evitando los tecnicismos. Tras ello, se proporciona una breve introducción a las operaciones más básicas de MPI. Ambos vídeos se han alojado y subtitulado en YouTube y pueden reproducirse desde las direcciones: <https://youtu.be/o76-VP6WFCo> y [https://youtu.be/if0MWI\\_9xzM](https://youtu.be/if0MWI_9xzM).

La pestaña de “Monitorización” da acceso a una página desde la que se pueden ejecutar cada uno de los benchmarks de los NPBs (clase C), y observar su salida por terminal, así como ver en tiempo real el impacto que la ejecución de los mismos tiene sobre

la carga en los nodos y el tráfico de red. Esta página de monitorización se muestra en la Figura 8. En la parte superior se encuentran los medidores de uso de CPU y el tráfico de red de cada una de las Raspberry Pis. En la parte inferior se encuentran los botones para ejecutar los diferentes programas de prueba y poder observar en tiempo real el impacto sobre los diferentes nodos. En concreto, la figura está mostrando la salida durante la ejecución del kernel FT (Fast Fourier Transform). Gracias a esta monitorización se puede observar que durante la ejecución de esta aplicación todos los nodos trabajan en paralelo y se van alternando las fases de comunicación y computación.

Finalmente, en la pestaña “Acerca de” se incluye, entre otra información, una breve descripción de Clúptiter.

## VI. CONCLUSIONES

En este trabajo se ha presentado Clúptiter, un supercomputador en miniatura con todas sus secciones bien diferenciadas y extrapolables a un supercomputador real.

A pesar de que el rendimiento y la escalabilidad no son buenos (las Raspberry Pis no están diseñadas para ser un computador especialmente eficaz en este sentido), Clúptiter permite ejecutar programas paralelos y comprobar cómo son capaces de cooperar los diferentes componentes hardware del sistema para acelerar la ejecución de aplicaciones científicas y de ingeniería. Al fin y al cabo, esto es lo que se perseguía como objetivo final de este proyecto.

Clúptiter está actualmente disponible en la “Sala Demostrador Tecnológico” del Centro de Investigación en Tecnologías de la Información y las Comunicaciones (CITIC, <https://citic.udc.es/demostrador-tecnologico/>) de la Universidade da Coruña como herramienta divulgativa, y se utiliza en las visitas al centro de, entre otro público, estudiantes de secundaria.

Todos los recursos relacionados con este trabajo, incluido el código de la aplicación web, se encuentran disponibles en la dirección [https://github.com/forcegk/GEI\\_TFG](https://github.com/forcegk/GEI_TFG) bajo licencia MIT.

## AGRADECIMIENTOS

Clúptiter ha sido financiado por el Grupo de investigación en Arquitectura de Computadores de la Universidade da Coruña (<https://gac.udc.es>).

## REFERENCIAS

- [1] Branko Balon and Milenko 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] Narasimha Saii Yamanoor and Srihari 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.nasa.gov/software/npb.html>, Last updated: January 27, 2022.
- [4] Julien Marot and Salah Bourennane, “Raspberry Pi for image processing education,” in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 2364–2366.



Fig. 8: Página de monitorización de Clúptier

- [5] Gianni Pasolini, Alessandro Bazzi, and Flavio 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] Jaroslav Sobota, Roman Pišl, Pavel Balda, and Miloš Schlegel, "Raspberry Pi and Arduino boards in control education," *IFAC Proceedings Volumes*, vol. 46, no. 17, pp. 7–12, 2013.
- [7] Phil Legg, Alan Mills, and Ian 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] Simon J Cox, James T Cox, Richard P Boardman, Steven J Johnston, Mark Scott, and Neil S O'Brien, "Iridispi: a low-cost, compact demonstration cluster," *Cluster Computing*, vol. 17, pp. 349–358, 2014.
- [9] Philipp Gschwandtner, Alexander Hirsch, Peter Thoman, Peter Zangerl, Herbert Jordan, and Thomas Fahringer, "The cluster coffer: teaching HPC on the road," *Journal of Parallel and Distributed Computing*, vol. 155, pp. 50–62, 2021.
- [10] João Saffran, Gabriel Garcia, Matheus A Souza, Pedro H Penna, Márcio Castro, Luís FW Góes, and Henrique C Freitas, "A low-cost energy-efficient Raspberry Pi cluster for data mining algorithms," in *Euro-Par 2016: Parallel Processing Workshops*. Springer, 2017, pp. 788–799.
- [11] Khalid M Hosny, Amal Magdi, Nabil A Lashin, Osama El-Komy, and Ahmad Salah, "Robust color image watermarking using multi-core Raspberry Pi cluster," *Multimedia Tools and Applications*, vol. 81, no. 12, pp. 17185–17204, 2022.
- [12] Agraj Aher, Janhavi Kasar, Palasha Ahuja, and Varsha 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>, June 2021.
- [14] Alonso Rodríguez-Iglesias, "Diseño e implementación hardware y software de un mini-supercomputador con Raspberry Pi," [https://github.com/forcegk/GEI\\_TFG/releases/download/memoria/MemoriaTFG.pdf](https://github.com/forcegk/GEI_TFG/releases/download/memoria/MemoriaTFG.pdf), 2021, Trabajo Fin de Grado, Universidade da Coruña.
- [15] Netdata, Inc., "Monitoring everything in real time for free," <https://www.netdata.cloud>.