



Universidad de Castilla-La Mancha
Escuela Superior de Ingeniería Informática

Trabajo Fin de Grado
Grado en Ingeniería Informática
Tecnología específica de Ingeniería de Computadores

Monitorización y análisis del consumo de energía en un clúster HPC

Pablo Franco García-Filoso

Julio 2023



TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Tecnología específica de Ingeniería de Computadores

Monitorización y análisis del consumo de energía en un clúster HPC

Autor: Pablo Franco García-Filoso

Tutor: José Luis Sánchez García

Co-Tutor: Francisco José Alfaro Cortés

Julio 2023

Este trabajo va dedicado a mi familia y amigos que han sido mi apoyo a lo largo de toda la carrera y a lo largo de este proyecto. Así como a todos mis compañeros de Asador La Granja, mi otra familia.

Declaración de autoría

Yo, Pablo Franco García-Filoso, con DNI 74525483D, declaro que soy el único autor del trabajo fin de grado titulado Monitorización y análisis del consumo de energía en un clúster HPC, que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual, y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Albacete, a 11 de Julio de 2023

Fdo.: Pablo Franco García-Filoso

Resumen

El consumo de energía de los centros de datos y de supercomputación es un aspecto muy relevante desde diversos puntos de vista. El económico porque supone la mayor parte del presupuesto de este tipo de centros; el medioambiental por su gran impacto debido a las emisiones de CO₂; y el técnico pues puede imponer restricciones a su expansión y, como consecuencia, a su máximo rendimiento.

Por ello, se están diseñando y aplicando soluciones para mejorar la eficiencia energética sin comprometer el rendimiento del sistema. Se están usando mejores sistemas de refrigeración, gestionando mucho mejor los intervalos de tiempo sin actividad, utilizando de forma más eficaz la virtualización de los recursos, e incluso diseñando aplicaciones más eficientes energéticamente.

En este proceso de búsqueda de alternativas que consigan sistemas energéticamente más eficientes, es esencial disponer de herramientas que continuamente monitoricen y controlen el consumo de energía de dichos sistemas. Con estas herramientas los administradores de centros de datos y de supercomputación pueden disponer de datos muy valiosos para tomar decisiones inteligentes en la gestión de los recursos disponibles.

Entre las facilidades existentes para obtener datos de consumo, cabe mencionar aquellas que usan soporte hardware/software de los propios sistemas informáticos, y las que utilizan dispositivos externos, situados entre la toma general de corriente y el sistema o sus componentes.

En este Trabajo Fin de Grado se muestra el uso, sobre un clúster HPC, de un caso particular de cada una de esas facilidades. Se comparan los resultados obtenidos con ambas estrategias y se indican las ventajas e inconvenientes de cada una de ellas extraídas a partir de la experiencia de este trabajo.

Agradecimientos

Agradecimiento en especial a mis tutores por su entera disponibilidad y predisposición a la hora de la realización del proyecto. A mi familia y amigos por todo su apoyo durante los meses que me he visto inmerso. Por último mencionar también todos los organismos y recursos disponibles que nos proporciona la universidad. Agradecimiento en especial a Francisco Parreño por ayudarnos en todas las dudas que hemos ido teniendo.

Índice general

1	Introducción y motivación	1
1.1	Introducción.....	1
1.2	Motivación.....	3
1.3	Estructura de la memoria	4
2	Objetivos y metodología.....	5
2.1	Objetivos	5
2.2	Metodología.....	6
2.2.1	<i>Planificación.....</i>	7
2.3	Competencias.....	7
3	Estado del arte	9
3.1	Clúster HPC.....	9
3.1.1	<i>Arquitectura.....</i>	10
3.1.2	<i>Componentes.....</i>	12
3.2	Consumo de energía	16
3.2.1	<i>Estrategias para reducir el consumo.....</i>	17
3.2.2	<i>Monitorización y control del consumo en sistemas HPC.....</i>	19
4	Desarrollo.....	23
4.1	Intel Performance Counter Monitor	24
4.2	Unidad de distribución de potencia.....	26
4.2.1	<i>Protocolo SNMP</i>	28
4.2.2	<i>MG-SOFT MIB Browser</i>	30

4.2.3	<i>Cacti</i>	32
4.2.4	<i>Herramientas en línea de comandos.</i>	34
4.3	Discusión sobre Intel PCM y PDU.	35
4.3.1	<i>Particularidades de este Trabajo Fin de Grado.</i>	37
5	Pruebas y resultados	39
5.1	Entorno de pruebas	40
5.1.1	<i>Hardware.</i>	40
5.1.2	<i>Software</i>	42
5.2	Métricas	44
5.3	Configuración de las pruebas.	44
5.4	Resultados y análisis	45
5.4.1	<i>Consumo de los cores de la CPU.</i>	45
5.4.2	<i>Consumo de los nodos del clúster.</i>	53
5.4.3	<i>Consumo de un switch de la subred del clúster</i>	57
5.4.4	<i>Resumen</i>	58
6	Conclusiones y trabajo futuro	61
6.1	Conclusiones	61
6.2	Trabajo futuro	62
6.3	Competencias.	63
Bibliografía		67
A	Anexo 1	69
A.1	Instalación, compilación y ejecución de las aplicaciones HPC.	69
A.1.1	<i>Gadget.</i>	69
A.1.2	<i>Graph500.</i>	71
A.1.3	<i>Gromacs</i>	72
A.1.4	<i>Hpcg</i>	73
A.1.5	<i>Lammps</i>	74
A.2	Ejecución de las aplicaciones que registran el consumo	75

Índice de figuras

2.1	Diagrama de Gantt.	7
3.1	Ejemplo de Clúster [23].	10
3.2	Arquitectura clúster HPC (CETA Ciemat [8]).	11
3.3	Sistema de Almacenamiento DAS [13].	13
3.4	Sistema de Almacenamiento NAS [39].	14
4.1	Armario o rack con PDU.	26
4.2	Gráfica generada y visualizada en MIB Browser.	30
4.3	Crear una gráfica en MIB Browser.	31
4.4	Interfaz Cacti.	33
4.5	Opciones para añadir un dispositivo en Cacti.	33
5.1	Nodos y switches conectados a la PDU.	41
5.2	Topología de una subred del clúster CELLIA.	41
5.3	Evolución del consumo a lo largo del tiempo. Obtenido con la PDU para 16 tareas y 8 nodos. Obteniendo las medidas cada 0.5s segundos.	46
5.4	Consumo de un nodo en función del número de tareas. Medido con pcm-power.	47
5.5	Consumo de un nodo en función del número de tareas. Medido con la PDU.	48
5.6	Estimación del consumo de un nodo en función del número de cores. Aplicación Lammmps y datos recogidos con Intel PCM.	51
5.7	Estimación del consumo de un nodo en función del número de cores. Aplicación Lammmps y datos recogidos con la PDU.	52
5.8	Evolución del consumo al variar el número de nodos. Medido con pcm-power.	53
5.9	Evolución del consumo al variar el número de nodos. Medido con la PDU.	54
5.10	Estimación del consumo en función del número de nodos. Aplicación Lammmps y datos recogidos con Intel PCM.	55
5.11	Estimación del consumo en función del número de nodos. Aplicación Lammmps y datos recogidos con la PDU.	56
5.12	Puertos utilizados para medir el consumo de los switches.	57

5.13 Consumo en función del número de puertos activos de un switch. Medido con la PDU.	59
--	----

Índice de tablas

5.1	Número medio de vatios de un nodo en función del número de tareas cuando las medidas se toman usando Intel PCM.	49
5.2	Número medio de vatios de un nodo en función del número de tareas cuando las medidas se toman usando la PDU.	50
5.3	Consumo en función de los puertos activos del switch (PDU).	58

Índice de códigos

4.1	Ejemplo de uso de contadores en un código [15]	25
A.1	Script Slurm para ejecutar la aplicación Gadget4.	70
A.2	Script Slurm para ejecutar la aplicación Graph500.	71
A.3	Script Slurm para ejecutar la aplicación Gromacs.	72
A.4	Script Slurm para ejecutar la aplicación HPCG.	73
A.5	Script Slurm para ejecutar la aplicación Lammps.	75
A.6	Script para recoger el consumo de la aplicación Graph500.	76
A.7	Contenido del fichero llamadas.sh.	77
A.8	Contenido del fichero procesar.sh	78

1. Introducción y motivación

En este primer capítulo se introduce el contexto en el que se desarrolla el trabajo, se indican los motivos principales que han llevado a realizarlo, así como la forma en la que se han estructurado los diferentes capítulos y secciones de este documento.

1.1. Introducción

La industria de los centros de datos está creciendo rápidamente a medida que aumenta el número de aplicaciones que requieren computación y almacenamiento. Las cargas de trabajo de HPC (del inglés High Performance Computing) junto con las debidas a las aplicaciones de Inteligencia Artificial (IA) y de análisis de datos suponen una exigencia de recursos tal que están subiendo el listón en cuanto a rendimiento, necesidad de memoria y de E/S de los sistemas informáticos de los centros de datos.

Algo similar se puede decir de los centros de supercomputación que, aunque menos numerosos y destinados a un tipo de aplicaciones más restringido, también están siendo requeridos para mayor número de cargas de trabajo y con altas exigencias computacionales.

Los sistemas de ambos tipos de centros están aumentando su capacidad de procesamiento incorporando aceleradores, además de las CPU convencionales, introduciendo así una gama más amplia y heterogénea de configuraciones para estos sistemas. Están formados por miles de nodos de procesamiento y almacenamiento, unidos por una red de interconexión de altas prestaciones, siendo la arquitectura clúster ampliamente utilizada, casi el 90% de los sistemas en la lista Top500 [38].

Todos los nodos interconectados en el clúster actúan como un único ordenador dotado de una gran potencia de cálculo. La computación paralela permite a un clúster HPC gestionar grandes cargas de trabajo dividiéndolas en tareas computacionales separadas que se pueden desarrollar al mismo tiempo.

Una elevada potencia de cálculo permite resolver problemas de mayor envergadura y encontrar soluciones más rápidamente, con mayor precisión y a menor coste. Todo ello supone una ventaja competitiva, que en ciertos campos puede significar la diferencia entre ser el primero en publicar o no hacerlo; y en la industria, puede determinar quién llega primero a la oficina de patentes.

El clúster de alto rendimiento demuestra cada día su gran valor en una amplia variedad de usos: predicción meteorológica, diseño industrial, dinámica molecular, modelización astronómica, entre otros.

Este tipo de sistemas HPC también desempeñan un papel cada vez más importante en las empresas. El alto rendimiento es clave en la minería de datos, en el renderizado de imágenes, en la resolución de consultas a bases de datos, etc. Los grandes proveedores de servicios de Internet como Google utilizan estos sistemas.

En los sistemas actuales para computación de alto rendimiento la energía juega un papel cada vez más importante. Prueba de ello, al margen de la preocupación por el coste que supone el gasto en electricidad, es el hecho de que a las métricas de rendimiento tradicionales, como el tiempo o la productividad, se han unido otras como la potencia, la energía o la temperatura, en diversos contextos y en distintas combinaciones, para diversas aplicaciones. Es ya de obligado cumplimiento incluir datos en este sentido cuando se realizan propuestas para mejorar el funcionamiento de estos sistemas.

Así las cosas, el diseño de los nuevos sistemas basados en clúster HPC, que intentan alcanzar rendimiento exascale, busca no sobrepasar una potencia máxima de 20 MW [11]. Se estima que los centros de datos son responsables hasta del 3% del consumo mundial de electricidad y se prevé que alcancen el 4% en 2030. Una instalación media de un centro de datos consume anualmente entre 20 y 50 MW, electricidad suficiente para abastecer a más de 35.000 hogares.

Aunque el hardware más eficiente y las innovaciones en infraestructuras y sistemas de refrigeración han podido contrarrestar la creciente demanda de electricidad en los centros de datos, es bastante probable que esta demanda sea tan elevada que no pueda compensarse únicamente con mejoras en la eficiencia del hardware, el software o la infraestructura en su conjunto.

Aparte de la clasificación Top500, orientada al rendimiento, la lista Green500 clasifica los grandes sistemas de computación en función del rendimiento por vatio [37]. La adopción generalizada de los procesadores gráficos ha contribuido a aumentar este ratio, pero a la vez ha incrementado el consumo energético.

1.2. Motivación

Como se ha indicado anteriormente, el consumo energético de los centros de datos y de supercomputación es un aspecto de los mismos que preocupa a sus gestores. Es por ello que, desde hace ya algunos años, se están diseñando y aplicando estrategias para detener su gran aumento al ir incrementando el tamaño de los sistemas de esos centros. En este contexto, es imprescindible poder saber en cada momento cuál es el consumo energético en esos centros, tanto a nivel global como particular de los diferentes elementos que lo forman.

La importancia de medir el consumo, y hacerlo de la forma más precisa posible, es lo que ha motivado este trabajo, que se ha centrado en un elemento habitual en estos centros como es un clúster HPC. Las medidas de su consumo permiten tomar decisiones sobre diferentes aspectos que marcan su normal funcionamiento. Algunos de ellos son los siguientes:

- **Eficiencia energética.** Medir el consumo energético en un clúster HPC es fundamental para evaluar y mejorar la eficiencia energética. Un clúster HPC está compuesto por un gran número de nodos de computación, cada uno con varios procesadores de alto rendimiento. Las medidas de consumo ayudan a identificar el consumo excesivo en los nodos o el sistema completo. Con esta información se pueden implementar mejores estrategias para reducir el consumo energético sin comprometer el rendimiento.
- **Control del coste.** La factura de electricidad contribuye significativamente a los costes operativos de un clúster HPC. Con medidas precisas, es posible optimizar la asignación de recursos y planificar el futuro crecimiento del clúster HPC de manera más eficiente, ayudando a controlar los costes y hacer uso adecuado de los recursos financieros.
- **Planificación y capacidad.** Los datos de consumo en un clúster HPC son de gran valor para su planificación. Al conocer la demanda energética actual y el crecimiento esperado, los administradores del clúster pueden determinar si la infraestructura eléctrica existente es suficiente para soportar el sistema o si se requiere una actualización. Además, los datos de consumo permiten establecer perfiles de carga y picos de demanda, facilitando la gestión y la planificación de futuras expansiones o mejoras.
- **Sostenibilidad y responsabilidad ambiental.** La medición del consumo energético en un clúster HPC permite evaluar el impacto ambiental de la infraestructura y tomar medidas para reducirlo. Al identificar y corregir ineficiencias energéticas, se puede lograr una reducción significativa en la huella de carbono del clúster. Además, la medición del consumo proporciona datos cuantificables que pueden ser usados en informes y auditorías relacionados con la sostenibilidad, cumpliendo así con los estándares y regulaciones medioambientales.

En resumen, medir el consumo energético en un clúster HPC es fundamental para mejorar la eficiencia energética, controlar los costes operativos, planificar adecuadamente la infraestructura eléctrica y garantizar la sostenibilidad y responsabilidad ambiental. Esta práctica permite optimizar el rendimiento del sistema, reducir los gastos innecesarios y garantizar un uso eficiente y responsable de los recursos energéticos.

1.3. Estructura de la memoria

El resto de la memoria de este Trabajo Fin de Grado está organizado de la siguiente forma:

- Capítulo 2: **Objetivos y metodología.** Se indica el objetivo principal del trabajo y se incluye una lista de tareas que se han realizado para poder lograrlo.
- Capítulo 3: **Estado del arte.** Se incluyen los aspectos básicos necesarios para el desarrollo de este trabajo, especialmente los relacionados con sistemas basados en clúster de computación y el consumo de energía de los mismos.
- Capítulo 4: **Consumo de energía en un clúster HPC.** Se proporciona información detallada sobre las medidas de consumo realizadas en un clúster real, tanto mediante un dispositivo externo como a través de los contadores hardware/software disponibles en la propia infraestructura del clúster.
- Capítulo 5: **Pruebas y Análisis de Resultados.** Incluye detalles sobre la plataforma hardware/software utilizada, las aplicaciones HPC usadas, la configuración de las pruebas realizadas, los resultados de estas y su análisis.
- Capítulo 6: **Conclusiones y Trabajo Futuro.** Se resumen las principales conclusiones obtenidas a partir del trabajo realizado y se incluyen posibles propuestas para dar continuidad al mismo.
- **Bibliografía.** Incluye los documentos, manuales y webs que han sido consultadas para el desarrollo del trabajo.

2. Objetivos y metodología

En este capítulo se indican cuáles son los objetivos principales de este Trabajo Fin de Grado, y las diversas tareas que se han realizado para lograrlos. Para cada tarea se describe brevemente en qué consiste y un diagrama de Gantt muestra la forma en la que avanzan las tareas en el tiempo. En la parte final de este capítulo, se indican las competencias del Grado que se trabajan con el desarrollo de este trabajo.

2.1. Objetivos

Para obtener medidas de consumo energético en un clúster HPC hay diversas herramientas. Algunas se basan en recoger datos en contadores hardware incorporados en los propios componentes del clúster, procesadores por ejemplo. Otras, utilizan dispositivos externos, generalmente situados entre la toma general de corriente y el clúster o sus componentes.

En este TFG se van a usar las dos vías para recoger datos de consumo de un clúster, analizar sus diferencias y similitudes, y, en la medida de lo posible, realizar un estudio comparativo, determinando así cuál de ellas es más interesante utilizar, en términos de precisión, simplicidad y eficiencia.

Todo esto constituye el objetivo principal de este Trabajo Fin de Grado, el cual se alcanzará cubriendo este conjunto de objetivos particulares:

- Conocer con detalle las características, componentes y funcionamiento de un clúster HPC. Es necesario ejecutar aplicaciones en el clúster para tomar medidas de consumo, y por tanto hay que aprender a usarlo adecuadamente.
- Aprender a usar el dispositivo externo conectado a los elementos principales del clúster. Examinando la documentación disponible, se tiene que estudiar la forma de usar el medidor y cómo recoger los datos de consumo.
- Estudiar el soporte hardware/software de los componentes del clúster para realizar mediciones. También por medio de los manuales disponibles de los elementos del clúster se debe comprobar el soporte disponible para obtener datos de consumo.

-
- Saber obtener datos de consumo con los dos tipos de técnicas, es decir medidor externo y soporte interno. Con la información disponible, hay que aprender a usar ambas vías de obtención de datos de consumo.
 - Comparar y analizar los datos obtenidos. Hay que hacer un análisis de los datos recogidos por las dos vías, compararlos y analizar sus ventajas e inconvenientes.

2.2. Metodología

Las tareas que deben completarse para lograr el objetivo principal del trabajo se realizarán siguiendo la metodología utilizada habitualmente en este tipo de trabajos, que en general consiste básicamente en motivar y justificar la conveniencia de realizar este trabajo; elegir el material necesario para ello y aprender su manejo y los datos que ofrece; establecer las configuraciones hardware/software adecuadas; recoger datos y analizarlos para extraer las conclusiones más relevantes.

Al aplicar dicha metodología al caso particular de este Trabajo Fin de Grado, se obtienen las siguientes tareas principales a realizar:

1. Revisión de las características del clúster Cellia y del medidor de consumo disponible, con el propósito de conocer los aspectos necesarios para realizar el trabajo. Para realizar esta tarea se consultará diversa documentación del clúster y sus componentes, y el manual de usuario del medidor de consumo.
2. Estudio del soporte hardware/software disponible en/para los componentes del clúster que permita medir el consumo sin necesidad de medidor externo. En el caso de los nodos de cómputo ese soporte existe, mientras que habrá que averiguar a qué nivel lo hay en el caso de la red de interconexión.
3. Selección de un conjunto representativo de aplicaciones de centros de datos y de supercomputación que serán usadas en el estudio. Los datos de consumo serán recogidos durante la ejecución de dichas aplicaciones, y también cuando el sistema esté ocioso para comprobar los consumos base de los componentes.
4. Configuración de las pruebas y desarrollo de las mismas para la obtención de los datos de consumo de energía. Se considerarán, como se ha indicado, varias aplicaciones con diferentes cargas y se harán variaciones sobre algunos parámetros del clúster, como el número de nodos, puertos de los switches, NICs, etc.
5. Análisis de resultados y obtención de conclusiones. Se espera que los datos recogidos por las dos vías (soporte interno y medidor externo) sean muy numerosos y por tanto será necesario desarrollar código adecuados para su manejo, y/o paquetes estadísticos para la presentación y análisis de los mismos. Se deben extraer conclusiones sobre las ventajas e inconvenientes de cada método, y en la medida de lo posible establecer comparaciones justas que ayuden a decidir cuál de ellos resulta más adecuado utilizar.
6. Escritura de la memoria. Se irá registrando en un documento todo el trabajo realizado durante el desarrollo del trabajo.

2.2.1. Planificación

En la figura 2.1 se muestran las tareas a realizar y una estimación del tiempo de dedicación a cada una de ellas.

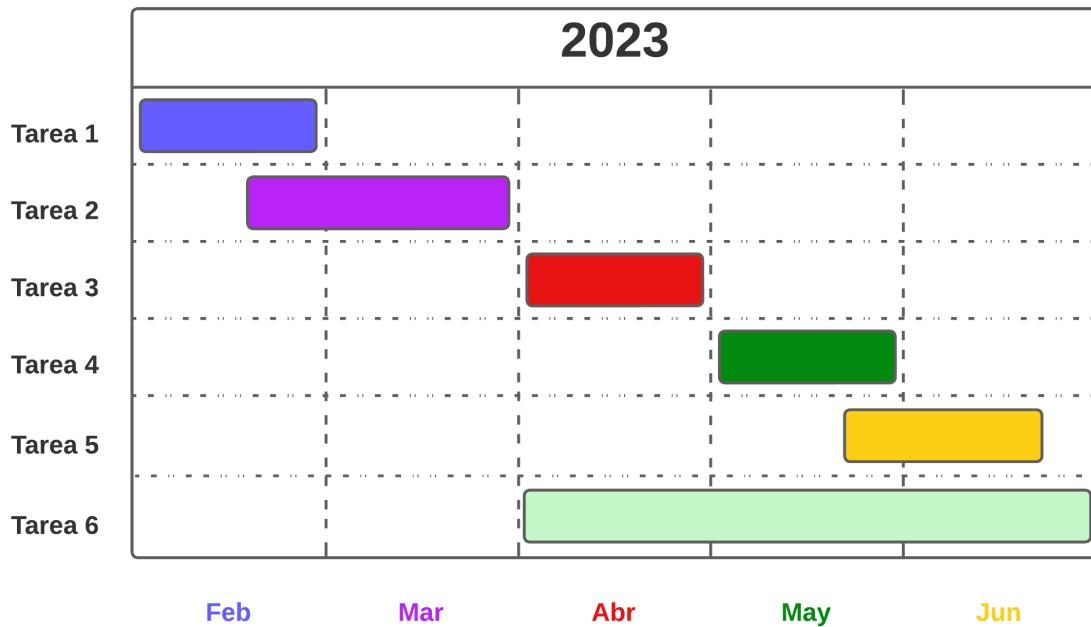


Figura 2.1: Diagrama de Gantt.

2.3. Competencias

Las competencias que se pretenden trabajar con este TFG de la intensificación Ingeniería de Computadores son aquellas relacionadas con la capacidad de analizar, estudiar y evaluar ciertos aspectos de una plataforma hardware como es un clúster HPC.

[IC3] Capacidad de analizar y evaluar arquitecturas de computadores, incluyendo plataformas paralelas y distribuidas, así como desarrollar y optimizar software para las mismas.

[IC7] Capacidad para analizar, evaluar, seleccionar y configurar plataformas hardware para el desarrollo y ejecución de aplicaciones y servicios informáticos.

3. Estado del arte

En este capítulo se hace una breve revisión de la arquitectura de un clúster de altas prestaciones, de sus componentes principales y del uso que se le está dando en el contexto de grandes centros de datos y de supercomputación. Se dedica especial atención a un aspecto muy relevante en la actualidad como es el consumo de energía de este tipo de sistemas. Se muestran algunos datos que ponen de manifiesto dicha relevancia, y se resumen algunas medidas que se han propuesto para reducir su elevado consumo energético.

3.1. Clúster HPC

Los centros de supercomputación especialmente, y los centros de datos cada vez más, manejan aplicaciones que tienen requisitos computacionales muy elevados. Para cumplir con esas exigencias, estos centros disponen de sistemas de computación de alto rendimiento, en particular sistemas basados en clúster, referidos de forma habitual como clúster HPC (del inglés High Performance Computing).

Un clúster HPC es un conjunto de servidores informáticos individuales, conectados entre sí a través de una red de alta velocidad, que trabajan en conjunto para completar tareas complejas que requieren una gran cantidad de procesamiento. Aunque el clúster esté formado por una gran cantidad de componentes, al usuario se le ofrece una visión de un único ordenador, lo cual le facilita el uso de todos los recursos disponibles. Aunque aparezca ante el usuario como un único computador, cada componente tiene la capacidad de funcionar por sí misma y de manera independiente.

Un clúster HPC es actualmente utilizado en campos muy diversos y para una gran variedad de aplicaciones, que van desde la predicción meteorológica hasta el diseño industrial, pasando por la dinámica molecular o la modelización astronómica [28].



Figura 3.1: Ejemplo de Clúster [23].

3.1.1. Arquitectura

La arquitectura HPC es más compleja que simplemente seleccionar un conjunto de componentes y juntarlos, porque todos los componentes interactúan entre sí para dar forma al rendimiento real que obtiene una aplicación. A la hora de diseñar un clúster hay que pensar en la estructura interna del clúster, decidiendo qué funciones desempeñarán cada uno de los componentes que lo forman y cómo será la red de interconexión que los conecte [31].

Se pueden distinguir tres grupos de nodos: los nodos de entrada, los nodos de almacenamiento y los nodos de procesamiento. Los primeros son a los que nos conectamos para hacer uso del clúster; los de almacenamiento guardan los datos de manera permanente; y los de cómputo son los que ejecutan programas dentro del entorno, accediendo a los nodos de almacenamiento y controlados por un planificador o sistema de gestión de procesos.

Normalmente, un clúster pequeño tiene sólo un nodo de entrada, que incluso puede hacer la función de almacenamiento, mientras que un clúster grande pueden tener más de uno para garantizar una mayor disponibilidad. Hay distintas formas de acceder a los nodos de almacenamiento desde los de trabajo, lo que puede dar lugar a distintas arquitecturas.

Dependiendo del sistema HPC, los nodos de computación, incluso individualmente, pueden ser mucho más potentes que un ordenador personal típico. A menudo tienen múltiples procesadores (cada uno con varios núcleos), y pueden tener aceleradores (como unidades de procesamiento gráfico (GPU)) y otras capacidades menos habituales en los ordenadores personales. En la figura 3.2 se muestra la arquitectura de un clúster en producción.

Para funcionar al máximo rendimiento, cada componente debe seguir el ritmo de los demás. Por ejemplo, el componente de almacenamiento debe ser capaz de proporcionar y guardar datos hacia y desde los servidores tan rápido como se procesan. Del mismo modo, los componentes de red deben ser capaces de realizar la transferencia de datos a alta velocidad entre los nodos de cómputo y los de almacenamiento de datos. Si alguno de los componentes del clúster tiene un rendimiento por debajo del ofrecido por el resto, el rendimiento global de toda la infraestructura HPC se puede resentir de forma significativa.

Para destacar la importancia que tiene hoy en día este tipo de arquitectura, indicar que en la última lista de los 500 supercomputadores con mayor potencia computacional, el 98,2% de ellos están basados en una arquitectura tipo clúster. Además, esta tendencia ha ido en aumento con el paso de los años, desplazando a otro tipo de propuestas, en general más heterogéneas [38].

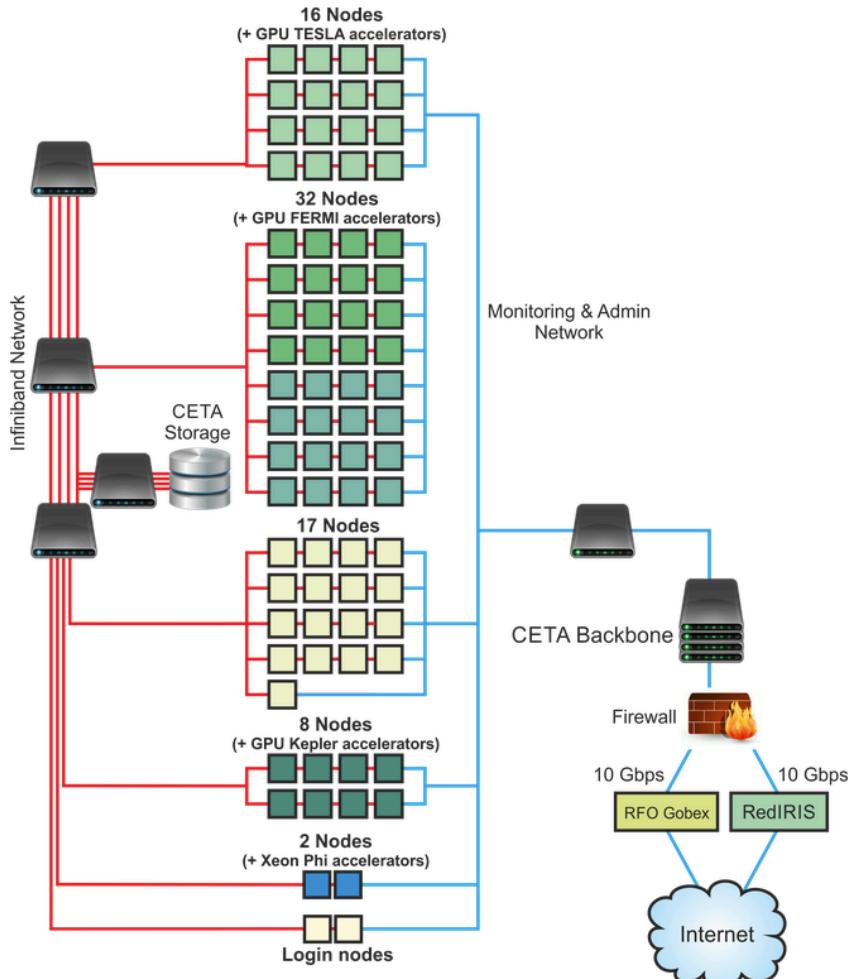


Figura 3.2: Arquitectura clúster HPC (CETA Ciemat [8]).

3.1.2. Componentes

Un clúster HPC está compuesto por diferentes elementos hardware/software, siendo los principales los que se indican a continuación:

- **Nodos de cómputo.** Dedicados a realizar tareas de cálculo fundamentalmente y que actúan de forma independiente o de forma conjunta con todos o parte de los demás nodos. Se colocan en uno o varios armarios/racks, e incluyen los elementos habituales de un servidor [18]:
 - Procesadores. En cada nodo se incluyen varios conectores físicos en los que se insertan las unidades centrales de proceso (CPUs). Se suelen incluir varios, lo que aumenta la capacidad de proceso del nodo. Por lo general, los conectores en un clúster HPC están diseñados para soportar una amplia gama de CPUs, lo que permite que los nodos de cálculo sean flexibles y escalables en función de las necesidades específicas de cada aplicación. También son importantes porque afectan la capacidad de la CPU para acceder a la memoria local y compartida del clúster, lo que puede influir en el rendimiento general del sistema [18].
 - Memoria. Se encuentra dividida en varios canales. Se usan para almacenamiento temporal y en la ejecución de aplicaciones. Cada nodo del clúster tiene su propia memoria local, que está diseñada para almacenar datos que se están utilizando en cada momento. También puede ser usada por otros nodos de forma remota a través de la red de interconexión. La memoria de un clúster HPC es crucial para el rendimiento del sistema, ya que si el rendimiento de la memoria es insuficiente, se pueden producir cuellos de botella en el proceso de cálculo [18].
 - Discos. Cada uno de los nodos puede tener soporte para almacenamiento local permanente, y su uso puede ser distinto para cada uno de ellos, aunque en general se dedica a almacenar datos usados por las aplicaciones o generados por ellas durante su ejecución. Por tanto, no suelen ser de gran capacidad, pues para guardar grandes volúmenes de datos se dedican los nodos de almacenamiento.
 - Aceleradores. Dispositivos diseñados para mejorar el rendimiento de cálculo en los nodos, y por tanto en el clúster, al proporcionar una potencia de cómputo adicional. Los aceleradores pueden ser tarjetas gráficas (GPU, del inglés Graphics Processing Units), FPGAs (Field-Programmable Gate Arrays), coprocesadores especializados u otros dispositivos diseñados para realizar operaciones específicas de forma más rápida y eficiente que las CPUs convencionales. Se puede mejorar significativamente el rendimiento y reducir el tiempo necesario para realizar tareas complejas. Desde hace varios años, varios de los diez primeros supercomputadores que aparecen en el Top500 incorporan GPUs.
 - Interfaces de Red. Componentes físicos y lógicos que permiten la comunicación de datos entre los nodos de cálculo y otros dispositivos en el clúster. Las interfaces de red se conectan a través de un sistema de interconexión, como una red de área local (LAN) de alta velocidad o una red de área amplia (WAN) dedicada, que se utiliza para transmitir datos entre los nodos de cálculo y otros dispositivos conectados a ella [18].

- **Nodos de almacenamiento.** Estos nodos están diseñados específicamente para proporcionar capacidad de almacenamiento masivo y un acceso rápido a los datos para las aplicaciones de computación intensiva. Están equipados con unidades de disco de alta capacidad y velocidad, como discos duros tradicionales o unidades de estado sólido (SSD), que permiten almacenar grandes volúmenes de datos de manera eficiente. Además, estos nodos suelen contar con una arquitectura de red de alto rendimiento para garantizar una transferencia rápida y confiable de datos entre los nodos de cómputo y los nodos de almacenamiento. Esto permite a los usuarios acceder a los datos necesarios de manera eficiente para realizar análisis complejos y ejecutar aplicaciones científicas de gran escala en el clúster HPC.

Los nodos de almacenamiento en un clúster HPC suelen implementarse utilizando diferentes configuraciones, como por ejemplo sistemas de almacenamiento conectado en red (NAS) o sistemas de almacenamiento directamente conectado (DAS).

- **Direct Attached Storage (DAS).** Los nodos de almacenamiento están directamente conectados a los nodos de cómputo, lo que garantiza un acceso de alta velocidad a los datos y una latencia mínima. Esta configuración es especialmente adecuada para aplicaciones que requieren un rendimiento extremadamente rápido y una baja latencia, como el procesamiento en tiempo real y la simulación de alta frecuencia. La conexión se realiza a través de una interfaz de almacenamiento de disco, como una conexión SCSI o SATA.

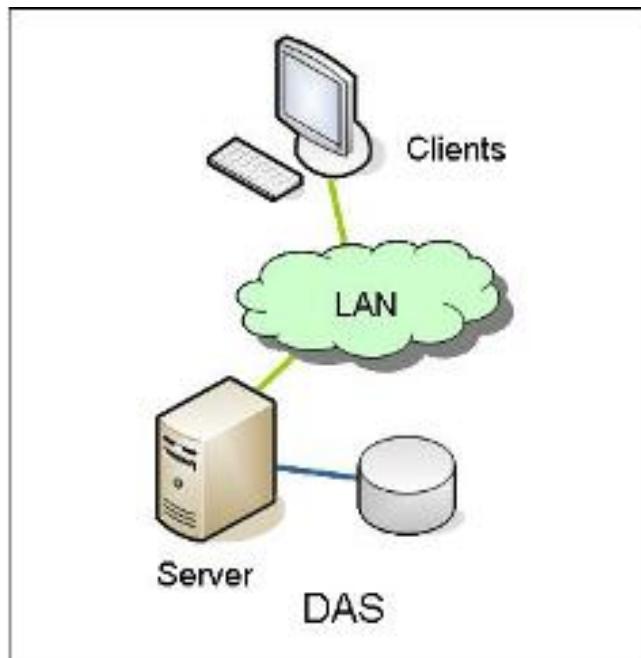


Figura 3.3: Sistema de Almacenamiento DAS [13].

- Network Attached Storage (NAS). Los nodos de almacenamiento están conectados a la red y actúan como servidores de archivos dedicados, proporcionando un acceso centralizado y compartido a los datos a través del protocolo de red, como NFS (Network File System) [33] o SMB (Server Message Block). Puede ofrecer una mayor disponibilidad de datos y capacidad de recuperación ante desastres en comparación con el almacenamiento DAS, ya que los datos están replicados en múltiples dispositivos de almacenamiento en red.

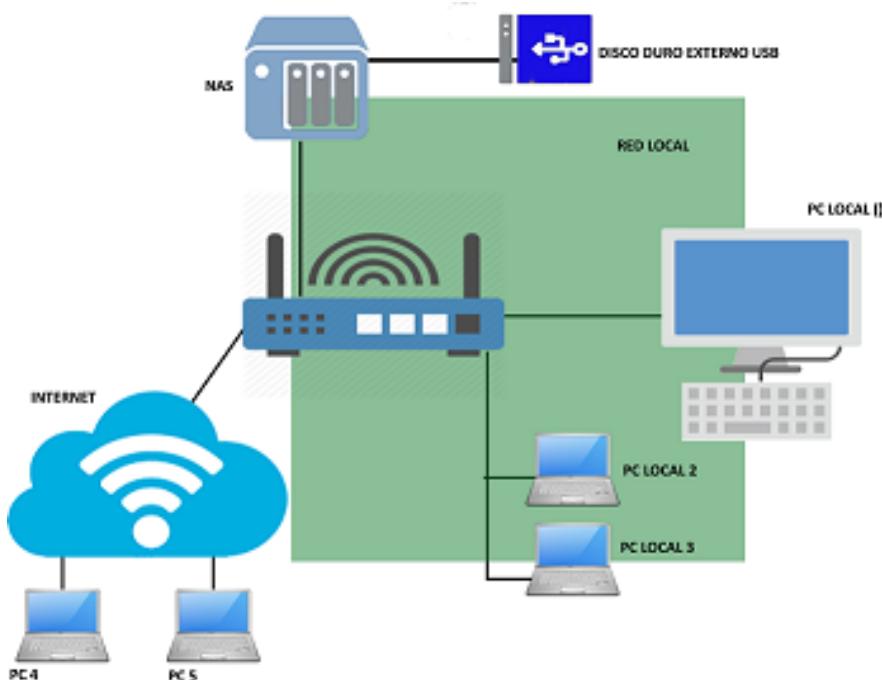


Figura 3.4: Sistema de Almacenamiento NAS [39].

- **Red de interconexión.** Es la infraestructura física y lógica que permite la comunicación entre todos los nodos del clúster. Esta interconexión puede tomar diferentes formas, dependiendo de la arquitectura del clúster, el tamaño del sistema y las necesidades específicas de las aplicaciones que se ejecutan en el clúster. En general, la interconexión en un clúster HPC se realiza mediante una red de alta velocidad dedicada. Algunos ejemplos de tecnologías ampliamente usadas son InfiniBand [21] o Gibabit Ethernet, que están incluidas en el 40% y el 45,4%, respectivamente, de los supercomputadores de la lista Top500. Los elementos principales de la red de interconexión son:
 - Comunicadores. Son dispositivos con varios puertos de entrada y salida, a través de los cuales fluye la información que se intercambian los nodos. Pueden estar integrados en los nodos de proceso o constituir nodos independientes dentro de la red.

- Cables. Establecen las conexiones físicas punto a punto entre comutadores y nodos. Son de cobre o fibra óptica, según los requisitos, así como apantallado o sin apantallar, según la distancia que se requiera sin interferencias. Pueden ser unidireccionales o bidireccionales, haciendo distinción entre full-duplex (tráfico en ambos sentidos de manera simultánea) o half-duplex (un sentido simultáneamente) en los bidireccionales.
- Interfaces de red. Ver sección 3.1.2.

Hay diferentes aspectos de diseño que caracterizan a las redes de interconexión de altas prestaciones, y que determinan su rendimiento [10]. Algunos de ellos son: la topología, que es la representación de la red mediante un grafo donde los vértices son los nodos y los arcos son los enlaces; técnica de conmutación, que determina cuándo y cómo se establecen las conexiones internas entre entradas y salidas, y además qué mensajes deben utilizar las rutas establecidas; control de flujo, que controla el avance de la información entre nodos, es decir, determina el momento en el que la información se transfiere entre componentes de la red (interfaces, buffers, enlaces); encaminamiento, que determina el camino que debe seguir un mensaje en la red para alcanzar su destino a partir del nodo fuente en el que se ha generado.

- **Alimentación y refrigeración.** Las fuentes de alimentación son dispositivos que suministran energía eléctrica a los componentes del clúster. Es importante contar con fuentes de alimentación redundantes para garantizar la disponibilidad del sistema en caso de fallo en una de ellas. La refrigeración es igualmente importante, ya que los componentes del clúster generan grandes cantidades de calor durante su funcionamiento. Si no se disipa adecuadamente, este calor puede afectar negativamente el rendimiento y la vida útil de los componentes. Los sistemas de refrigeración pueden ser de aire o líquido, y se encargan de mantener una temperatura adecuada en los equipos del clúster. Además, es importante contar con sistemas de control ambiental, como detectores de humedad y temperatura, para prevenir cualquier posible daño a los componentes.
- **Software de gestión.** Es el conjunto de herramientas que permiten la gestión y control del clúster. Cabe citar los sistemas operativos (Linux, Windows), programas de monitorización (Ganglia, Nagios), sistemas de gestión de procesos (Slurm, Torque) [32, 4] o aplicaciones para generar, depurar y compilar las aplicaciones que se ejecutarán en el clúster. Todas estas herramientas permiten a los usuarios administrar y ejecutar tareas y aplicaciones de manera eficiente, así como monitorizar el estado y el rendimiento del sistema. [40]

3.2. Consumo de energía

El consumo energético de los centros de datos y de supercomputación es un aspecto de gran relevancia tanto desde el punto de vista económico, debido a lo elevado de las facturas de electricidad de estos centros, como desde el lado técnico, pues puede imponer restricciones a su expansión y por tanto a su rendimiento. Es por ello que, desde hace ya años, se están diseñando y aplicando soluciones para detener su aumento según se incrementa el tamaño de este tipo de centros. Pero resulta crucial buscar soluciones que permitan mejorar la eficiencia energética sin comprometer el rendimiento del sistema [6].

Algunos de estos centros, los más grandes del mundo, pueden llegar a consumir tanta energía como una pequeña ciudad. Ya solo el gasto anual de energía de los centros de datos estadounidenses en 2021 fue de 13.000 millones de dólares, o lo que es lo mismo lo equivalente a la energía de 34 pequeñas centrales nucleares, de 500 mega vatios cada una [27]. Globalmente, se estima que los centros de datos usan aproximadamente 90 billones de kilovatios hora, lo que supone entre un 1% y un 3% del uso total de la electricidad. No obstante, esto no siempre ha sido así, ya que este porcentaje ha ido en aumento, de manera exponencial en los últimos años.

Los gestores de centros de datos deben garantizar un funcionamiento eficiente no sólo para disminuir costes, sino también para minimizar los efectos nocivos impuestos al medio ambiente. Cuanta más energía utiliza un centro de datos, más emisiones de gases de efecto invernadero se liberan al medio ambiente, lo que aumenta la huella de carbono del centro de datos. La población en general, y los gobiernos en particular, se están concienciando cada vez más de la importancia de cuidar el medio ambiente, y de hacerlo en cualquier ámbito.

La relevancia del coste energético de estos centros también se puede observar en detalles que pueden parecer nimios pero que tendrán efectos en el desarrollo de software más eficiente energéticamente hablando. Así, además de mantener un ranking de los supercomputadores en términos de rendimiento computacional (Top500) [38], se ha creado otra clasificación de estos sistemas que tiene en cuenta también el gasto energético. La lista Green500 [38] clasifica a estos sistemas en función de la relación flops y vatios.

Es una situación generalizada que la posición que ocupa el mismo sistema en cada lista sea diferente. En la última actualización, que se ha realizado en junio de 2023 el primer puesto en la lista Green500 ocupa el lugar número 255 en la lista Top500. Por otro lado, el Frontier que ocupa la primera posición en el Top500 está en sexta posición en este ranking de rendimiento energético. El segundo en la lista del Top500 (Fugaku) ocupa la posición 49 en la lista Green500. Y el tercero en el Top500 (LUMI) ocupa la séptima posición en la lista Green500.

Así pues, en el diseño de nuevos centros de datos y de supercomputación es imprescindible tener en cuenta la eficiencia energética. Se manejan diversas métricas para medir la eficiencia energética y el consumo de este tipo de centros, las cuales se usan para comparar diferentes tipos de instalaciones, y determinar de esta forma cuál es mejor o más conveniente. Algunas de las habituales son las siguientes:

- Eficiencia en el uso de la energía (PUE). Es la relación entre la energía total utilizada por un centro de datos y la energía consumida por los equipos informáticos. La iluminación, los equipos de refrigeración y las ineficiencias en la distribución de electricidad dentro de la instalación se incluyen en el cálculo de la energía total. La energía consumida por los equipos informáticos se calcula sumando las lecturas de potencia de los equipos de distribución eléctrica de los racks. La mayoría de los centros de datos tienen como objetivo un PUE inferior o igual a 1,5 o 1,4 para los nuevos centros de datos.
- Eficiencia en la infraestructura del centro de datos (DCIE). Es la inversa de PUE, que convierte la métrica PUE en un porcentaje. Se calcula dividiendo el consumo total de energía de todos los equipos informáticos por el consumo total de energía de la instalación. Esta métrica se utiliza para identificar la eficiencia de los equipos informáticos en relación con el consumo energético total del centro de datos.
- Eficiencia media de los centros de datos corporativos (CADE). Es una métrica utilizada para evaluar y calificar la eficiencia energética global de una instalación en función de su rendimiento. Tiene en cuenta la eficiencia energética de las instalaciones, sus índices de utilización y el nivel de utilización de los servidores. La eficiencia de las instalaciones (energía suministrada a los equipos informáticos) dividida por la energía extraída de los servicios públicos, se multiplica por la eficiencia de los activos de los sistemas informáticos, que es la utilización media de la unidad central de proceso (CPU) en todos los servidores hasta que se realizan esfuerzos de eficiencia.

3.2.1. Estrategias para reducir el consumo

Los responsables de los centros de datos y supercomputación aplican determinadas estrategias para reducir su consumo energético, entre las cuales cabe señalar las siguientes:

- Cambiar al modo "ECO". Muchas veces hay energía que es técnicamente consumida en los centros de datos pero realmente no es usada por ningún equipo en el centro, a lo que nos referimos como gasto de energía.[22].
- Aumentar los puntos de ajuste de temperatura. Los equipos informáticos modernos pueden funcionar en entornos más cálidos, pero muchos gestores de centros de datos siguen enfriándolos en exceso. El aumento de la temperatura puede suponer un ahorro de energía del 4-5% por cada grado de aumento de la temperatura de entrada del servidor. Implementar sensores ambientales y software de monitorización ambiental es fundamental para saber cuánto se pueden aumentar las temperaturas sin introducir un riesgo de daños en los equipos o tiempos de inactividad.

-
- Adoptar prácticas de refrigeración innovadoras. Dado que los sistemas de refrigeración suponen hasta el 60% de la factura de una instalación, los gestores de centros de datos deben adoptar las mejores prácticas de refrigeración. Algunas de las prácticas de refrigeración más eficientes consisten en aprovechar las mejores propiedades de transferencia térmica de los fluidos para refrigerar bastidores de alta densidad, mantener pasillos calientes/fríos, sellar las salidas de cables para minimizar el flujo de aire de derivación y usar paneles ciegos dentro de los armarios de los equipos.
 - Aplicar virtualización de los servidores. La virtualización de servidores implica el uso de servidores físicos como pools de capacidad informática lógica. Normalmente, las aplicaciones se despliegan de forma ineficiente a través de múltiples sistemas en los que hay un servidor y almacenamiento dedicados para cada aplicación. Cada una de estas plataformas consume casi toda la potencia que necesitaría en picos de carga, pero la plataforma está haciendo muy poco trabajo. La virtualización agrega servidores y almacenamiento en una plataforma compartida al tiempo que separa estrictamente sistemas, aplicaciones, datos y usuarios. Mejora drásticamente la utilización del hardware y permite reducir los servidores y dispositivos de almacenamiento que consumen mucha energía y sus equipos de refrigeración asociados.
 - Mantener actualizados los equipos. Aunque es costoso, los gestores de estos centros deberían plantearse actualizar sus equipos, ya que algunos de los antiguos son muy ineficientes. Los equipos más nuevos suelen estar diseñados para necesitar menos energía para realizar la misma cantidad de trabajo, si no más. La compra de nuevos equipos puede suponer una reducción general de los costes, ya que no sólo puede reducir la factura energética, sino que también disminuirá la cantidad de dinero que se gasta en el mantenimiento y la conservación que suelen tener los equipos más antiguos propensos a averiarse.
 - Apagar los equipos informáticos inactivos. Si los equipos informáticos están constantemente encendidos, cuando pasan a estar inactivos siguen consumiendo una parte significativa de la energía que consumirían si se utilizaran al máximo. Para combatir este gasto se pueden apagar remotamente las PDU de los racks y desconectar los equipos inactivos para reducir la cantidad de energía desperdiciada en las instalaciones.
 - Activar la función de gestión de energía de la CPU. La CPU consume más del 50% de la energía necesaria para el funcionamiento de un servidor. Para reducir este consumo, la función de gestión de energía disponible para la mayoría de las CPUs optimiza el consumo de energía cambiando dinámicamente entre varios estados de rendimiento en función de la utilización de la CPU. Este proceso se realiza sin reiniciar la CPU y puede suponer un importante ahorro anual de energía. [22].

A pesar de esas soluciones, el problema del consumo de energía en la industria informática en general, y especialmente en el de los centros de datos y de supercomputación, sigue siendo un tema relevante que aún no se ha resuelto por completo [22].

3.2.2. Monitorización y control del consumo en sistemas HPC

Los profesionales que administran los centros de datos y de supercomputación suelen utilizar herramientas de gestión de infraestructuras de este tipo de centros para controlar y gestionar el consumo de energía, lo que se traduce en una operatividad más eficiente. Estas herramientas proporcionan la información necesaria para aplicar y mejorar muchas de las medidas enumeradas en la sección anterior.

Con este tipo de medios, los administradores de centros de datos y de supercomputación pueden disponer de datos de consumo energético para tomar decisiones más inteligentes, obtener gráficos e informes en tiempo real sobre métricas como el PUE, crear informes de facturación para facilitar comportamientos más eficientes desde el punto de vista energético, ahorrar energía evitando el sobrefriamiento, identificar equipos que consumen mucha energía, y consolidar y virtualizar recursos de forma inteligente.

Para obtener datos del consumo energético de un clúster HPC en instantes determinados, se pueden seguir diversas alternativas. Algunas usan los contadores hardware que incorporan los propios componentes del clúster, procesadores por ejemplo. Otras utilizan dispositivos externos, situados entre la toma general de corriente y el clúster o sus componentes [26].

Todo este amplio conjunto de herramientas que permiten gestionar la energía en sistemas de computación de altas prestaciones, se pueden agrupar en dos categorías principales en función del tipo de tareas que pueden realizar: supervisión y control.

Hay herramientas que sólo permiten consultar el consumo de potencia o energía, mientras que otras permiten además limitar dicho consumo. Así mismo, hay herramientas cuyo objetivo es solamente limitar el consumo, haciéndolo de forma indirecta, por ejemplo modificando la frecuencia de los dispositivos. Por último, hay varias herramientas derivadas de las anteriores, que realizan las funciones de monitorización y control pero facilitando al usuario esa labor.

A continuación se mencionan algunas de esas herramientas agrupadas en varias clases, atendiendo a esa clasificación [9].

Monitorización de energía

Una vez superada la fase de ejecución de aplicaciones en entornos HPC, los esfuerzos se han centrado también en la eficiencia energética, convirtiéndola en uno de sus principales objetivos. Así, se comenzó a monitorizar el consumo de energía/potencia del sistema utilizando medidores externos. Estos dispositivos miden el consumo real, pero no pueden informar del consumo de los subcomponentes del sistema, como CPU, GPU o memoria. Algunos de estos dispositivos son:

-
- PowerMon2. Medidor hardware de consumo. Se trata de un dispositivo basado en un microcontrolador que puede medir la potencia de hasta ocho canales de una fuente de alimentación mediante monitores de potencia digitales. La API del software simplemente lee las muestras de potencia y las hace accesibles al usuario. La frecuencia de medición es de hasta 3 KHz.
 - PowerInsight. Medidor hardware de consumo. Es una solución que usa tecnología basada en ARM como tarjeta de adquisición de datos y placas de detección de energía conectadas a ésta. Se comunican los datos a través de Ethernet y actualmente puede medir hasta 30 canales. Los datos se agregan en la estación central para su postprocesamiento.
 - PowerPack. Medidor hardware de consumo. Tiene un software back-end que soporta múltiples dispositivos de medida de potencia incluyendo Watt's Up Pro, NI y RadioShack, entre otros. Permite la sincronización de segmentos de código de aplicación con perfiles de potencia. El principal problema de este tipo de prototipos es el coste (debido al costoso hardware de terceros) y la usabilidad/escalabilidad (debido a los voluminosos componentes y a los requisitos de cableado).
 - WattProf. Herramienta software utilizada para medir el consumo, que permite altas frecuencias, mediciones directas de los componentes hardware de un nodo de cómputo, tales como CPU, RAM, GPU... La plataforma incluye soporte para recoger los datos, almacenarlos en el controlador y transmitirlos a través de la red. El software permite sincronizar las mediciones con aplicaciones bajo prueba para permitir un perfil de potencia de las aplicaciones.

Control de potencia

Existen métodos indirectos para realizar ese control de potencia. Disponible para CPU y GPU, el escalado dinámico de voltaje y frecuencia (DVFS) es uno de los enfoques para controlar la energía. Se realiza mediante la reducción de voltaje y/o frecuencia del procesador, a costa de pérdida de rendimiento. La limitación dinámica de la concurrencia (DCT) es otra técnica que reduce el número de recursos disponibles y permiten al usuario controlar el consumo y el rendimiento de la aplicación.

Supervisión y control de la potencia

Cabe encuadrar en esta categoría soluciones estrictamente software, como las proporcionadas por empresas como Intel, AMD, IBM o NVIDIA, y otras que además tienen cierto soporte hardware como por ejemplo las unidades de distribución de potencia.

- RAPL (Running Average Power Limit) de Intel, APM (Application Power Management) de AMD, EnergyScale de IBM y NVML (NVIDIA's Management Library) de NVIDIA. Intel RAPL ofrece funciones de supervisión y consumo, para usuarios con privilegios, a través de registros específicos de modelo (MSR). RAPL estima el consumo de energía desde contadores hardware.

- HPC PowerAPI. El Laboratorio Nacional Sandia ha publicado recientemente la primera versión de una API de monitorización de potencia llamada HPC PowerAPI [25]. Se trata de una interfaz portátil para facilitar la medición y el control del consumo de energía en sistemas HPC. El estándar permite a varios proveedores colaborar o permitir la interoperabilidad de su software y hardware, con el objetivo de ofrecer una gestión de la energía más eficiente, y ofrecer soluciones de gestión de la energía más sistemáticas y ricas a los gestores de HPC, desarrolladores de aplicaciones y usuarios.
- PDU (Power Distributor Unit). Dispositivo físico que constituye una entidad de información utilizada en las comunicaciones de red. Proporciona información relevante para la transmisión y recepción de datos a través de un protocolo de red específico [17]. Recoge datos de sus entradas, a las que se le conectan fuentes de alimentación del clúster. Por ejemplo, pueden estar conectados varios nodos y conmutadores a distintas entradas. Además de consultas, se pueden configurar alarmas para controlar el consumo.

Herramientas derivadas

Algunas de las más conocidas son:

- PAPI (Performance Application Programming Interface) [35] que además de los contadores de rendimiento del procesador, se ha ampliado ofreciendo acceso a RAPL y NVML a través de la interfaz PAPI.
- PCM (Performance Counter Monitor) [15] es una herramienta software desarrollada por Intel que permite monitorizar y analizar el rendimiento de los procesadores Intel. Está diseñada específicamente para ofrecer a los desarrolladores una visión detallada de cómo se está utilizando el hardware de los procesadores Intel en tiempo real. Utiliza los contadores de rendimiento integrados en los procesadores Intel para recoger información sobre diversos aspectos del rendimiento del sistema. De esta manera no sólo mide el consumo, sino que también reporta una gran información sobre el funcionamiento del sistema.
- PUPiL (Performance under Power Limits) [41] es un ejemplo de enfoque híbrido hardware-software que maneja DVFS, la asignación de núcleos, el uso de sockets, el uso de memoria y el hyperthreading.

4. Desarrollo

En este capítulo se describen los aspectos más relevantes relacionados con las tareas de monitorización del consumo de energía de un clúster, excepto los resultados de consumo que se dejan para el siguiente capítulo. Se trata de tareas que actualmente se hacen necesarias para comprobar el nivel de eficiencia energética del clúster, y también la de las aplicaciones que se usan en él.

De las diversas posibilidades mencionadas en la sección 3.2.2 para llevar a cabo esa monitorización, se han elegido dos para este trabajo: una basada en un medidor externo, y otra que utiliza contadores hardware. Son dos formas distintas de conseguir el mismo objetivo, y en este capítulo se van a indicar sus principales características, la forma de usarlas, los datos que muestran y una comparación de estos.

Este trabajo surge de la necesidad de establecer una metodología para realizar medidas de consumo de energía en el clúster del grupo de investigación Redes y Arquitecturas de Altas Prestaciones (RAAP) de la Universidad de Castilla-La Mancha. Son muy numerosos los trabajos que se realizan en este grupo utilizando ese clúster. En muchos de ellos es necesario conocer el gasto energético de las aplicaciones o algoritmos que funcionan en el clúster. Además, tanto en los congresos como en las revistas donde se publican los resultados de los trabajos de investigación del grupo se están exigiendo no sólo datos de rendimiento computacional sino también datos relativos al consumo.

Así pues, este trabajo es una primera aproximación a esa metodología, que muy probablemente será ajustada posteriormente en función de los resultados obtenidos y de las conclusiones que se extraigan a partir de éstos.

La sección 4.1 está dedicada a Intel Performance Counter Monitor, la sección 4.2 a la unidad de distribución de corriente, y la sección 4.3 incluye algunas reflexiones y comentarios sobre ambas, intentando determinar cuál de ellas sería más adecuado utilizar en cada situación.

4.1. Intel Performance Counter Monitor

Intel Performance Counter Monitor (PCM) es una herramienta software desarrollada por Intel que permite monitorizar y analizar el rendimiento de los procesadores Intel. Está diseñada específicamente para ofrecer a los desarrolladores una visión detallada de cómo se está utilizando el hardware de los procesadores Intel en tiempo real [15].

Intel PCM utiliza los contadores de rendimiento integrados en los procesadores Intel para recoger información sobre diversos aspectos del rendimiento del sistema. Estos contadores pueden medir parámetros como el número de instrucciones ejecutadas, los ciclos de reloj consumidos, los accesos a la caché, los fallos de caché, la energía consumida, entre otros. Al recopilar y analizar esta información, esta herramienta permite a los usuarios comprender mejor cómo se están utilizando los recursos del procesador e identificar posibles cuellos de botella y ofrecer así posibilidades de mejora [15].

Proporciona una serie de utilidades usadas en línea de comandos para la supervisión en tiempo real. Algunas de ellas son las siguientes:

- `pcm`, usada para realizar una supervisión básica del procesador. Proporciona instrucciones por ciclo, frecuencia del núcleo (incluida la tecnología Intel(r) Turbo Boost), ancho de banda de la memoria y de Intel(r) Quick Path Interconnect, ancho de banda de la memoria local y remota, fallos de caché, margen térmico del núcleo y del encapsulado de la CPU, utilización de la caché, consumo de energía de la CPU y de la memoria.
- `pcm-sensor-server`, expone métricas a través de http en formato JSON.
- `pcm-memory`, monitoriza el ancho de banda de la memoria (por canal y por DIMM de la DRAM).
- `pcm-latency`, monitoriza los fallos de caché y la latencia de la memoria DDR.
- `pcm-pcie`, monitoriza el ancho de banda del PCIe por socket.
- `pcm-iio`, monitoriza el ancho de banda PCIe por dispositivo PCIe.
- `pcm-numa`, monitoriza accesos locales y remotos a la memoria.
- `pcm-bw-histogram`, muestra un histograma de utilización del ancho de banda de la memoria.
- `pcm-core` y `pmu-query`: monitoriza eventos arbitrarios del núcleo del procesador.
- `pcm-tsx`, monitoriza métricas de rendimiento de extensiones de sincronización transaccional.
- `pcm-power`, monitoriza datos de consumo. Muestra datos de consumo de potencia y energía del procesador, Intel(r) Quick Path Interconnect, la memoria DRAM y otras métricas relacionadas con la energía. Esta utilidad es la que se va a usar en este TFG.

Estas utilidades disponen de varios parámetros de configuración. Así por ejemplo, se puede especificar la frecuencia con la que se realizan las mediciones, el nombre de un archivo para guardar los resultados o si se quiere mostrar o no la información en pantalla.

Ejemplo: pcm-power 1 -i=300 -silent

donde "pcm-power" es utilizado para recabar datos relacionados con el consumo; "1" se refiere al número de segundos que dura la medición; "i = 300" indica el intervalo de tiempo entre mediciones consecutivas; y "silent" evita mostrar información en tiempo real.

La salida, relacionada con el consumo, que ofrece ese comando tiene este aspecto:

```
S0; Consumed energy units: 233088; Consumed Joules: 14.23; Watts: 14.26; ...
S0; Consumed DRAM energy units: 0; Consumed DRAM Joules: 0.00; DRAM Watts: 0.00
```

Las aplicaciones también se pueden monitorizar desde dentro, como se muestra en el código 4.1. El proceso consta de una primera inicialización de los contadores de rendimiento. Luego, el estado del contador puede ser capturado tanto antes como después de la sección de código de interés. Diferentes rutinas capturan los contadores para núcleos, sockets o el sistema completo, y almacenan su estado en las estructuras de datos correspondientes. Otras rutinas ofrecen la posibilidad de calcular la métrica basándose en estos estados.

```
PCM * m = PCM::getInstance();
if (m->program() != PCM::Success) return;
SystemCounterState before_sstate = getSystemCounterState();
    [run your code here]
SystemCounterState after_sstate = getSystemCounterState();
cout << "Instructions per clock:" << getIPC(before_sstate,
    after_sstate)
    << "L3 cache hit ratio:" << getL3CacheHitRatio(before_sstate,
        after_sstate)
    << "Bytes read:" << getBytesReadFromMC(before_sstate,
        after_sstate)
    << [and so on]..
```

Código 4.1: Ejemplo de uso de contadores en un código [15]

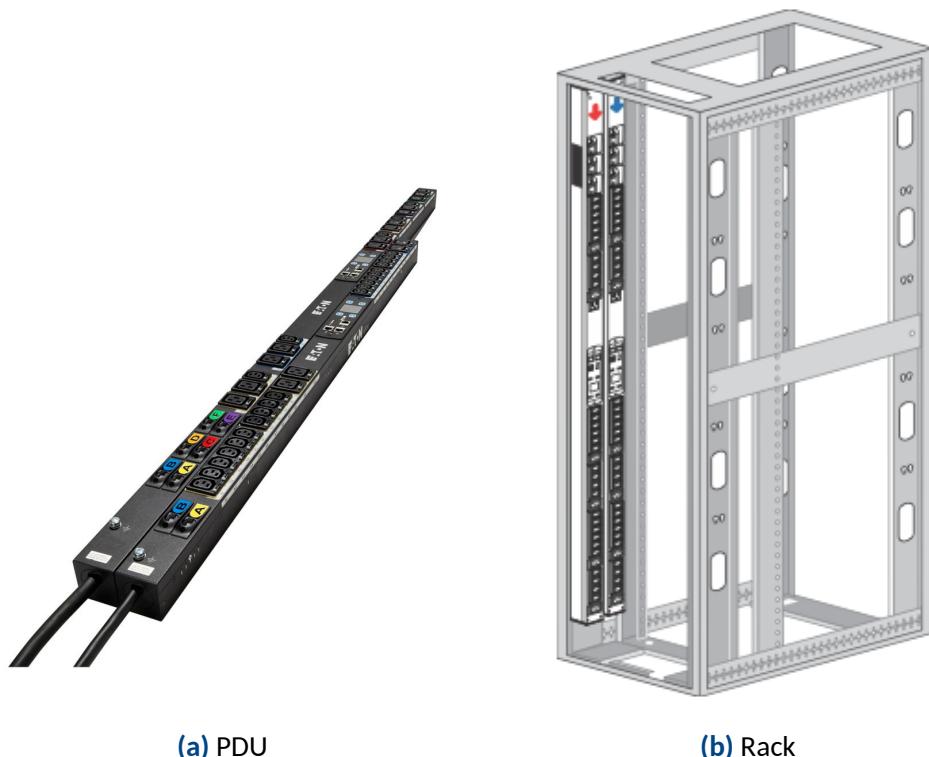
En este Trabajo Fin de Grado, se obtendrán los datos de interés desde fuera de la aplicación. La razón principal es que las aplicaciones utilizadas (sección 5.1.2) son complejas y llevaría tiempo localizar el punto más adecuado donde incluir las llamadas a esas rutinas.

4.2. Unidad de distribución de potencia

Una unidad de distribución de potencia (PDU) (figura 4.1a) se encarga de distribuir la energía a los distintos componentes de una infraestructura informática. Una PDU recibe energía de una fuente de alimentación y la distribuye a los dispositivos que están conectados a ella. Es básicamente una regleta de alta calidad, segura conforme a las normas industriales, que ofrece funciones opcionales para la supervisión, la conmutación y la medición de energía de forma fiable de los componentes instalados en los armarios de instalaciones como las que se pueden encontrar en centros de datos y supercomputación (figura 4.1b).

La fuente de alimentación puede ser una red eléctrica pública o un sistema de alimentación ininterrumpida (SAI). La PDU recibe la energía y luego la distribuye a los dispositivos a través de múltiples tomas o enchufes. Las tomas pueden ser una toma de corriente estándar o una toma especializada diseñada para equipos informáticos.

Las PDU también pueden tener características adicionales como protección contra sobretensiones, protección contra sobrecargas y capacidades de supervisión. La protección contra sobretensiones ayuda a proteger los dispositivos conectados de picos o subidas de tensión. La protección contra sobrecargas ayuda a evitar que la PDU se sobrecargue, lo que puede causar daños a la PDU y a los dispositivos conectados.



(a) PDU

(b) Rack

Figura 4.1: Armario o rack con PDU.

Las funciones de supervisión permiten a los administradores controlar el consumo de energía de los dispositivos conectados. Esta información puede utilizarse para identificar áreas de derroche de energía y aplicar medidas de ahorro energético. Algunas PDU también tienen funciones de gestión remota, que permiten a los administradores supervisar y controlar el suministro de energía a los dispositivos conectados desde una ubicación remota.

Hay diferentes tipos de PDU, con diferentes capacidades. Las habituales son:

- Basic (BA), sin supervisión ni controles inteligentes.
- In-Line Metered (IL), con supervisión en la entrada.
- Metered Input (MI), con supervisión en la entrada y derivación.
- Metered Outlet (MO), con supervisión en la entrada, la derivación y supervisión individual de las salidas, pero sin control de las salidas.
- Switched (SW), con control de las salidas pero sin supervisión individual de las salidas.
- Managed (MA), con supervisión en la entrada, derivación y control y supervisión individual de las salidas.

Las PDU instaladas directamente en los racks desempeñan un papel importante para conseguir que las infraestructuras TI tengan un funcionamiento óptimo. Con las PDU inteligentes, los administradores de TI crean la base para un funcionamiento seguro, eficiente y energéticamente optimizado de los centros de datos.

Una PDU se puede considerar una entidad de información utilizada en las comunicaciones de red. Ofrece datos que contienen información relevante para la transmisión y recepción de datos a través de un protocolo de red específico, como es el caso de SNMP (sección 4.2.1).

La PDU juega un papel fundamental en la comunicación entre los dispositivos de una red, ya que encapsula y transporta los datos a través de los distintos niveles o capas del modelo de referencia OSI (Open Systems Interconnection) o del modelo TCP/IP [17].

Como ya se ha mencionado, la preocupación por la eficiencia energética en los centros de datos no para de aumentar, y elementos como las PDU ganan una importancia altísima, ya que la mejora en este sector siempre debe ir precedido de un registro del consumo. Cuanto mayor sea la calidad de la PDU, más datos se podrán recabar y relacionarlos de manera individual con los diferentes componentes conectados a sus enchufes.

A la hora de seleccionar el modelo específico, se debe de tener en cuenta no sólo las características físicas tales como número de enchufes y funciones de supervisión necesaria, sino que debe poder adaptarse al rack y el rack a la misma. De otra forma sería ineficiente por muy bien que se hayan elegidos sus características [29].

4.2.1. Protocolo SNMP

El protocolo de gestión de redes SNMP (Simple Network Management Protocol) [20] es posiblemente el más usado en la comunicación entre el sistema gestor y los dispositivos gestionados. Este protocolo surge de la necesidad de resolver dos aspectos clave que aparecen cuando una red escala de manera evidente:

- En cuanto a la red, sus recursos y las aplicaciones que funcionan de manera distribuida comienzan a ganar relevancia de manera muy notable.
- Cuanto mayor sea la red a gestionar, mayor será la probabilidad de que los dispositivos fallen, quedando la red o una parte de esta inutilizable, afectando al desempeño, y pudiendo llegar hasta niveles inaceptables.

Se considera un estándar en este ámbito. El protocolo SNMP fue inicialmente definido por la IETF en su primera versión (snmp v1) y ha evolucionado hasta la tercera versión (snmp v3), mejorando tanto en eficiencia de comunicación como en seguridad.

Los elementos principales que se manejan en el protocolo, y con los que es necesario estar familiarizado para entender su funcionamiento son los siguientes [20]:

- Sistema Gestor. Se trata de un servidor centralizado o NMS (Network Management System) que administra todos los dispositivos. Su función principal es realizar solicitudes SNMP a los dispositivos administrados para obtener información sobre su estado o componentes específicos. Dependiendo del tipo de solicitudes, también puede realizar cambios en los dispositivos. Además, el sistema de gestión recibe SNMP Traps o mensajes asíncronos enviados por los dispositivos administrados.
- Dispositivo gestionado. Un router, switch, servidor o cualquier otro dispositivo con una dirección IP y un agente SNMP ejecutándose.
- Agente SNMP. Es una aplicación o proceso que se encuentra en los dispositivos administrados. Su tarea principal es mantener actualizada una base de datos, llamada MIB (Management Information Basement), donde se almacena información sobre varios componentes del dispositivo. Por ejemplo, la utilización de ancho de banda en una interfaz de un router o el uso de CPU en un servidor. El agente SNMP también responde a las solicitudes SNMP enviadas por el sistema de gestión y proporciona la información solicitada. Estas solicitudes son consultas sobre la base de datos MIB.
- MIB (Management Information Basement). Es la base de datos donde se almacena información relevante sobre el dispositivo. La estructura de esta base de datos es similar a un árbol. Algunas ramas del árbol, conocidas como MIB II o MIB pública, son comunes a todos los dispositivos, sin importar su fabricante o tipo. Otras ramas están reservadas y son exclusivas de cada fabricante. La estructura de la base de datos MIB está definida por la SMI (Structure Management Information), siendo la versión inicial (SMIv1) especificada en el RFC 1155 y la versión posterior (SMIv2) en el RFC 2578.

Cada una de las entidades que componen el protocolo se comunican a través de mensajes UDP. Estos mensajes están formados por un identificador de versión, nombre de comunidad SNMP y una unidad de datos del protocolo o PDU (Protocol Data Unit) ¹. Todas las implementaciones SNMP toleran 4 tipos de PDU [19]: GetRequest-PDU, GetNextRequest-PDU, SetRequest-PDU, Trap-PDU.

Una de las partes fundamentales es la Base de Información de Administración, MIB, son los identificadores de Objeto (OID). Los OIDs se utilizan para identificar y acceder a los datos específicos almacenados en la MIB. Están estructurados en una notación jerárquica, similar a un árbol, llamada Notación de Identificador de Objeto (Object Identifier Notation). Siguen la estructura de un árbol invertido, donde cada nodo en el árbol tiene un número único y se divide en subnodos [30].

Las principales ventajas del uso de este protocolo son:

- Es un protocolo maduro, estándar de facto aceptado por la industria.
- Está disponible en gran cantidad de productos.
- Es fácil de implementar y requiere pocos recursos del sistema.
- Facilita la comprensión de las funciones y herramientas de gestión, así como su uso, a los usuarios de estas aplicaciones, principalmente gestores de la red [7].

Entre los inconvenientes cabe señalar los siguientes:

- Falta de seguridad: cualquier estación puede reiniciar variables, no hay control de acceso y la identificación de comunidad viaja sin encriptar.
- Mala utilización del ancho de banda: no existe la posibilidad de transferir información por bloques.
- Limitaciones en el mecanismo de traps: sólo se puede informar de algunos eventos previstos.
- No es apropiado para gestionar redes muy grandes (por el sondeo).

Para recoger datos de diferentes dispositivos utilizando como base el protocolo SNMP, se pueden seguir varias aproximaciones. Así por ejemplo, se pueden utilizar aplicaciones con mayor o menor capacidad en cuanto a presentación de esos datos, o se pueden usar medios más simples como el uso de comandos en línea.

A continuación se dan algunos detalles de las que se han usado en algún momento del desarrollo de este Trabajo Fin de Grado: navegadores MIB, como es el caso de MG-SOFT MIB Browser y Cacti; y comandos desde el terminal.

¹No confundir PDU en el contexto del protocolo SNMP con PDU como unidad distribuidora de potencia.

4.2.2. MG-SOFT MIB Browser

Este navegador de MG-Soft Corporation [2] es una potente herramienta de análisis que permite extraer datos MIB de todo tipo de dispositivos habilitados para SNMP y mostrarlos en un formato comprensible.

MG-SOFT MIB Browser Professional Edition con MIB Compiler es el software específico que se instalaría para la monitorización a través del protocolo SNMP. MIB Browser permite realizar operaciones SNMP Get, SNMP GetNext, SNMP GetBulk y SNMP Set. Además, el software permite capturar y visualizar paquetes SNMP Trap y SNMP Inform enviados desde dispositivos o aplicaciones SNMP arbitrarios de la red.

Algunas de las características a destacar de esta aplicación son [2]:

- Interfaz gráfica.
- Soporte completo para las 3 versiones de SNMP.
- Monitorización de varios dispositivos SNMP simultáneamente.
- Editor avanzado de tablas SNMP.
- Vista de tabla para tablas MIB. (Esto sirve para ver en tiempo real, en forma de tabla, todos los valores de los componentes de una MIB dentro de una carpeta).
- Comparación de agentes SNMP.

Una función de gran interés es la de poder crear una gráfica con los valores de un componente de la MIB cuyo comportamiento se quiera observar. En la figura 4.2 se muestra cómo evoluciona el consumo de potencia durante un determinado periodo de tiempo.

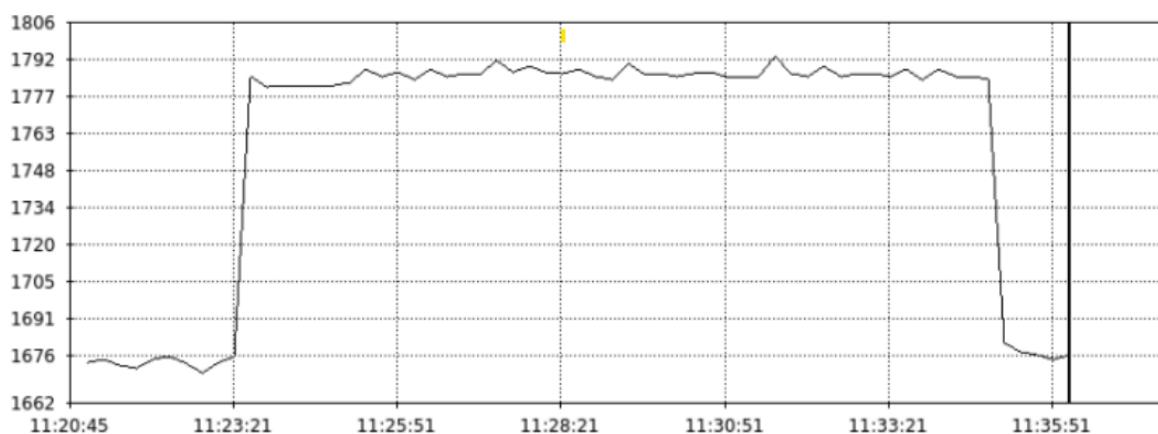


Figura 4.2: Gráfica generada y visualizada en MIB Browser.

En concreto, la gráfica se ha obtenido ejecutando una aplicación HPC durante unos 12 minutos, recogiéndose datos del consumo en vatios desde dos minutos antes y hasta dos minutos después de su finalización.

Se observa de forma clara cómo se incrementa el consumo con el inicio de la ejecución de la aplicación, consumo que se mantiene en los mismos niveles hasta que finaliza, volviendo después a los niveles anteriores a dicha ejecución.

En la figura 4.3 se incluye una captura de la aplicación que muestra cómo crear una gráfica. El primer paso es seleccionar el objeto del que se quiere hacer la gráfica y luego de todas las opciones se elige "graph".

En la misma figura se pueden ver otras opciones que ofrece este navegador, como por ejemplo "walk" o "get" para poder acceder al valor que tiene en ese momento el objeto de la MIB. También se puede usar el comando getNext, que hará un get al siguiente componente de la MIB, y se puede obtener el OID de un componente para realizar alguna consulta de manera externa desde la línea de comandos; además de muchas más herramientas.

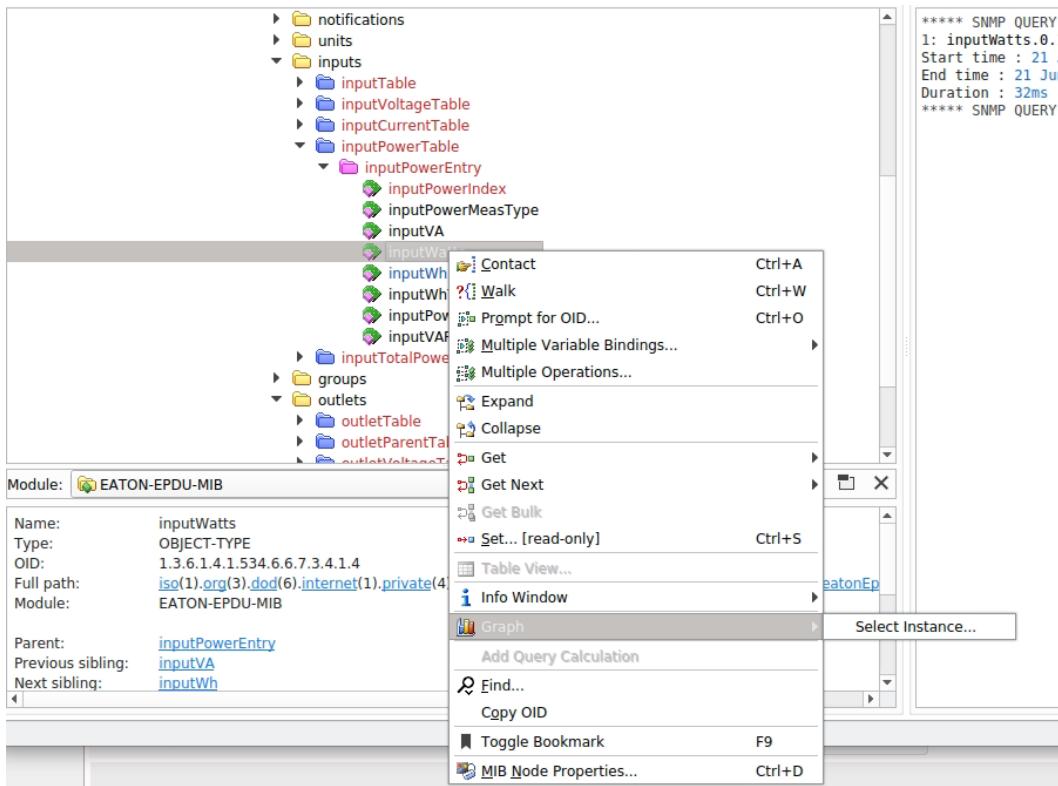


Figura 4.3: Crear una gráfica en MIB Browser.

4.2.3. Cacti

Cacti es una aplicación web disponible para Linux y Windows, de código abierto creada por Tobias Oetiker en el año 2005 [3]. Permite generar gráficas en tiempo real o con datos que han sido guardados previamente en una base de datos.

Cacti fue creada utilizando la base de cuatro aplicaciones o protocolos [3]:

- MySql. Base de datos relacional donde se almacena la información relacionada con la configuración de la aplicación, los usuarios, los equipos registrados y las plantillas asignadas.
- Apache y PHP. Utilizada para crear la aplicación web, además de su configuración y el visionado de las gráficas
- RRdtool. Base de datos, en este caso circular, utilizada para guardar la información referida a los gráficos.
- Protocolo SNMP. Cuyo funcionamiento radica en realizar los polling a los equipos de red.

Puntos a destacar de Cacti que la hacen diferente a las demás:

- Personalización de gráficas. Escasas limitaciones en el origen de los datos mostrados por las gráficas. Pueden estar compuestas por cualquier métrica.
- No hay restricción de licencia. A diferencia, por ejemplo, con MIB Broser donde ciertas funcionalidades vienen solo en versiones de pago.
- Requisitos hardware. Son muy limitados, de manera que puede funcionar en casi cualquier dispositivo.
- Código abierto. Se puede conocer por tanto el funcionamiento interno de la aplicación y estudiarlo.

En el centro de Cacti y su base de datos están el dispositivo y la plantilla de dispositivo. El primer paso es crear dispositivos, que son los que tendrán ciertos atributos asociados como un plantilla de dispositivo, comunidad SNMP y más información, a partir de la cual se crearán gráficos y fuentes de datos apropiados para el dispositivo. Los gráficos se pueden colocar en una estructura árbol que permite un diseño organizado y flexible, permitiendo escalar desde unas decenas de hosts hasta miles de ellos.

En la figura 4.4 se puede ver cómo es la interfaz y todas las opciones principales que hay disponibles. De cada una de ellas se despliega un submenú con un conjunto de herramientas.

Para crear los dispositivos, se deben establecer una serie de opciones que permiten adaptar la aplicación al tipo de dispositivo que se quiera añadir y de la forma que mejor convenga (4.5).



Figura 4.4: Interfaz Cacti.

This is a detailed view of the 'Nuevo dispositivo' configuration form from Figure 4.4. It shows the following settings:

- Opciones General de dispositivo**
 - Nombre de equipo: required
 - Ubicación: Ninguno
 - Asociación de Sonda: Main Poller
 - Asociación de sitio del dispositivo: Edge
 - Plantilla de Dispositivo: Cacti Stats
 - Cantidad de procesos simultáneos: 1 Thread
 - Deshabilitar dispositivo: checked
- Opciones SNMP**
 - Versión SNMP: Versión 2
 - Comunidad SNMP: public
 - Puerto SNMP: 161
 - Tiempo de espera de SNMP: 500
 - Máximo OID por Get Request: 10 OID's
 - Bulk Walk Maximum Repetitions: Auto Detect on Re-Index
- Opciones de Disponibilidad/Accesibilidad**
 - Detección de dispositivos caídos: Uptime SNMP
 - Valor de tiempo de espera de ping: 400
 - Número de reintentos de Ping: 1
- Opciones adicionales**
 - Notas: (empty)
 - ID externo: (empty)

Figura 4.5: Opciones para añadir un dispositivo en Cacti.

4.2.4. Herramientas en línea de comandos

Otra alternativa consiste en usar algunas herramientas desde línea de comandos. Es la opción menos vistosa pero más simple. El primer paso sería instalar el propio SNMP en el dispositivo. Después, se recogen los datos en crudo directamente del dispositivo seleccionado y se tratan posteriormente de la forma que mejor convenga y usando las aplicaciones que se quiera.

Siguiendo esta metodología, lo primero que se necesita para acceder al dispositivo es su IP, la comunidad SNMP que utiliza y el OID del componente de la MIB del que se quiera obtener la información. Con esos datos se obtienen los datos usando herramientas como `snmpget` o `snmpwalk`, que reportan los datos en tiempo real. La sintaxis para `snmpwalk` es

```
snmpwalk [opciones] [comunidad/datos_de_autenticación] [IP del host] [OID]
```

Un ejemplo podría ser así:

```
snmpwalk -v1 -c public 161.67.132.209 1.3.6.1.4.1.534.6.6.7.3.4.1.4
```

donde

- `v1` selecciona la versión 1 de SNMP.
- `c` indica que el siguiente parámetro es la comunidad
- `public` es la comunidad.
- `161.67.132.209`, es la IP del dispositivo.
- `1.3.6.1.4.1.534.6.6.7.3.4.1.4` es el identificador del dispositivo, que es la manera de encontrar el objeto dentro de la MIB del host.

La herramienta `snmpwalk` no solo se utiliza para solicitar un registro de datos específico de un dispositivo SNMP, sino también para obtener registros de datos que le siguen (lo cual es útil en el caso de tablas, por ejemplo). Con el fin de obtener conjuntos completos de información, es decir, una MIB completa, `snmpwalk` utiliza mensajes del tipo GETNEXT que solicitan información a los agentes hasta que se alcanza el final de la MIB en cuestión [16].

La sintaxis del comando `snmpget` es similar:

```
snmpget [opciones] [comunidad/datos_de_autenticación] [IP del host] [OID]
```

La aplicación `snmpget` permite solicitar información de una unidad de red utilizando el protocolo simple de administración de red (SNMP). Para lograrlo, se utiliza el mensaje GET de SNMP, el cual solicita un registro de datos específico en el sistema objetivo [16].

Repetiendo estos comandos, mediante un sencillo script, el número de veces necesario durante el intervalo de tiempo adecuado, se puede recuperar de esta forma los datos que se requieran. Los datos se guardarán en fichero para su posterior tratamiento. Esta forma de recoger los datos tiene la ventaja de que no haría falta ninguna aplicación externa, con la consecuente descarga, configuración y puesta a punto que esto conlleva.

4.3. Discusión sobre Intel PCM y PDU

Como se ha indicado previamente, Intel PCM (Performance Counter Monitor) y PDU (Power Distribution Unit) son dos herramientas diferentes que se pueden utilizar para obtener datos del consumo de energía en un clúster. Por tanto, ambas se utilizan con el mismo propósito, pero hay diferencias significativas en términos de su enfoque, funcionalidad o implementación, entre otros aspectos. Se incluye a continuación una comparación usando algunos de esos aspectos.

- **Funcionalidad.** Intel PCM es un software que permite acceder a los contadores de rendimiento y energía integrados en los procesadores Intel, para obtener datos sobre el rendimiento y consumo de energía de los núcleos de la CPU, las cachés y otros componentes del procesador. Las PDUs son dispositivos físicos que se conectan directamente a la fuente de alimentación y registra datos de consumo de energía, voltaje, corriente y potencia, en tiempo real.
- **Acceso.** Intel PCM proporciona una interfaz de línea de comandos, y también funciones que pueden ser usadas en las aplicaciones que se quieren monitorizar. En general, se accede a las PDUs mediante software centralizado que recopila y presenta los datos de consumo.
- **Granularidad.** Intel PCM proporciona una granularidad muy fina en términos de mediciones de rendimiento y consumo de energía, pues permite obtener datos a nivel de núcleo, caché, instrucciones ejecutadas, latencia de memoria, entre otros. Por contra, la PDU proporciona datos a nivel del clúster o sistema en su conjunto, o también a nivel de componente, como por ejemplo nodo o switch. Es decir, mide el consumo de energía total del clúster, o de sus elementos principales, pero de los componentes de éstos.
- **Objetivo.** Intel PCM es muy adecuado para realizar análisis detallados del rendimiento y la eficiencia energética del procesador, mientras que la PDU es útil para obtener una visión general del consumo de energía del clúster completo.
- **Usuarios.** Intel PCM es habitualmente utilizado por desarrolladores de software y expertos en rendimiento de sistemas para realizar análisis profundos del rendimiento y la eficiencia energética en entornos de computación de alto rendimiento. Por otro lado, la PDU es más utilizada por el administrador del clúster para realizar un seguimiento del consumo de energía general y garantizar una utilización eficiente de los recursos.
- **Implementación.** Para usar Intel PCM se requiere su instalación y configuración en todos los nodos del clúster y es necesario tener acceso a los correspondientes contadores. En el caso de la PDU, hay que disponer de tantas como sean necesarias para conectar todos los elementos del clúster.
- **Coste.** Obviamente, el uso de Intel PCM no tiene coste pues el software es gratuito. En el caso de las PDUs sí existe un coste, el cual dependerá del tamaño del clúster. Hay que indicar que en grandes centros de datos y supercomputación es habitual disponer de este tipo de unidades, pues permiten ejercer un control sobre el gasto energético.

En resumen, Intel PCM está orientado a obtener datos de rendimiento y consumo de energía a nivel de CPU y componentes individuales, mientras que las PDUs se usan para medir el consumo de energía a nivel de clúster o sistema completo, o de sus elementos principales (nodos completos o switches). Así pues, dependiendo del objetivo que se persiga, será más adecuado el uso de un tipo de herramienta u otro.

Si se tuvieran que indicar la situación o situaciones en las que es mejor usar una u otra, podrían tenerse en cuenta las siguientes consideraciones:

Intel PCM

- Cuando se requiere un análisis detallado del rendimiento de los componentes individuales en el clúster, como los núcleos de la CPU o las cachés, que permita identificar cuellos de botella y optimizar el rendimiento a un nivel más fino de granularidad.
- Cuando se quiere mejorar la eficiencia energética de los procesadores y componentes individuales en el clúster, identificando áreas de ineficiencia y realizando ajustes precisos para optimizar el consumo de energía.
- Cuando el enfoque principal está en el rendimiento y consumo de energía de la CPU y sus componentes.

Hay que señalar que aunque Intel PCM es específico para procesadores Intel, lo indicado anteriormente es extensible a otro software similar para otros procesadores.

PDU

- Cuando se requiere una visión general del consumo de energía en todo el clúster, incluyendo todos los elementos principales, como servidores, dispositivos de red y sistemas de almacenamiento.
- Cuando no se requiera instalación ni configuración en cada nodo del clúster, simplificando así el despliegue y evitando la necesidad de acceder a los contadores de rendimiento y energía de los procesadores individuales.
- Cuando se requiera información sobre el consumo total de energía, distribución de la carga, identificación de equipos inefficientes, facturación de energía y otros datos útiles para la gestión y planificación del consumo de energía a nivel de clúster.
- Cuando se requiera total compatibilidad con una variedad de equipos y dispositivos en el clúster, independientemente del fabricante o modelo.

Obviamente, el uso conjunto de ambos tipos de herramientas puede ofrecer una información mucho más completa que permita tanto realizar tareas de monitorización y control por parte de los administradores del clúster, como desarrollar estudios que redunden en la mejora de todos los elementos que componen el clúster, por parte de investigadores y desarrolladores.

4.3.1. Particularidades de este Trabajo Fin de Grado

Con este Trabajo Fin de Grado se pretende mostrar la forma de obtener datos de consumo energético de los elementos principales de un clúster HPC. Es decir, principalmente, de nodos de cómputo y red de interconexión. Se han usado para ello Intel PCM y una PDU. Sin embargo, con ninguna de ellas se puede medir el consumo de todos esos elementos.

Con Intel PCM se puede acceder a los contadores de los procesadores, y con las medidas recogidas con ellos se puede conocer el consumo de los nodos de una forma aproximada. Los elementos de la red de interconexión disponibles en el clúster CELLIA no disponen de contadores que recojan medidas de consumo, y por tanto la única forma de obtener medidas de consumo de los switches será mediante la PDU.

Por otro lado, aunque a la PDU que hay en el clúster CELLIA están conectados tanto nodos de cómputo como switches, no es posible obtener directamente datos de consumo de cada uno individualmente. El modelo de PDU disponible es de tipo Metered Input, y sólo ofrece datos de consumo a nivel global de toda la PDU. En el capítulo 5 se indica cómo se ha procedido para obtener datos de consumo a nivel de nodo de cómputo y de puerto de un switch, pero se comprobará que no serán medidas totalmente precisas.

Otro detalle a señalar en este punto, debido a lo anteriormente indicado, es que las medidas recogidas con ambos tipos de herramientas son diferentes en términos absolutos pues con los contadores examinados por Intel PCM se obtienen valores de unas decenas de vatios para elementos como los núcleos del procesador o para los nodos de cómputo, mientras que los obtenidos de la PDU son un orden superior pues se recoge el consumo de todos los elementos conectados a la PDU. En este segundo caso, se realizan las consideraciones oportunas para obtener los datos de los nodos o del switch.

5. Pruebas y resultados

En este capítulo se incluyen los detalles más importantes relacionados con las diferentes pruebas de monitorización del consumo de energía que se han realizado en este trabajo. En concreto, se ha medido el consumo de potencia en el clúster CELLIA, administrado por el grupo de investigación Redes y Arquitecturas de Altas Prestaciones (RAAP) de la Universidad de Castilla-La Mancha.

En primer lugar se describe el entorno de trabajo en el que se han desarrollado las pruebas. Así, primero se indican las principales características del clúster y del dispositivo externo de medida; a continuación, el software que ha sido necesario utilizar, en concreto se incluye una breve descripción de las aplicaciones HPC que se han usado, y durante cuya ejecución se han realizado las medidas de consumo de energía; finalmente, se comentan las características de las pruebas realizadas, es decir, la configuración de las pruebas, la metodología seguida para realizarlas y el número total de ellas.

En la siguiente sección se incluyen los resultados obtenidos en las pruebas realizadas, para las dos vías de monitorización utilizadas: la basada en contadores internos y la que usa una PDU externa. Los datos se muestran utilizando tanto tablas como gráficas, y se hace tanto de forma separada para cada método como de manera agrupada, facilitando así la comparación de los resultados de ambas opciones.

Además de los datos numéricos, se incluyen comentarios fruto del análisis de aquellos. Se ponen de manifiesto con datos las similitudes y diferencias entre las dos metodologías de medición del consumo de energía en el clúster, y se incluyen unas recomendaciones finales sobre la conveniencia de usar una u otra en las situaciones en las que se requiera ese tipo de datos.

5.1. Entorno de pruebas

Se resumen aquí las características del medio en el que se ha desarrollado el trabajo, y en particular las referidas al proceso de monitorización (equipo, pruebas y resultados).

5.1.1. Hardware

Los principales sistemas o equipos utilizados son el clúster CELLIA y la PDU.

Clúster CELLIA

Después de varias ampliaciones, este clúster actualmente está compuesto por 55 servidores (para un total de 568 cores), 4,3 TB de RAM y 22,4 TB de almacenamiento con tecnología NVMe. Todos interconectados a través de una red InfiniBand y otra GigaEthernet.

Los nodos servidor (nodos de cálculo) se encuentran divididos en tres grupos:

- Un grupo de 38 nodos compuestos por procesadores Intel Xeon E5-2630L v3 de 8 cores. Tienen una memoria RAM de 32 GB con tecnología DDR4 a 1,866 GHz. En la parte del almacenamiento cuentan con un disco duro SATA de 500 GB. Cada nodo tiene una tarjeta de red dual port Infiniband Mellanox MT27500 ConnectX-3.
- Un grupo de 12 nodos con procesadores Intel Xeon E5-2620 v4 de 8 cores. Cuentan con una memoria RAM de 32 GB DDR4 a 1,866 GHz. Para almacenamiento tienen un disco duro SATA de 500 GB. Por el lado de la red, cada nodo tiene una tarjeta dual port Infiniband Mellanox MT27500 ConnectX-3.
- Un grupo de 14 nodos compuestos por dos procesadores Intel XeonSilver 4116 de 12 cores cada uno, memoria RAM de 192 GB DDR6 y una Tarjeta NVMe Ultrastar sn260 PCIE de 1,6 TB de capacidad. Adicionalmente, se tienen dos discos HPE 300 GB SAS 12G Enterprise 10K. Cada nodo tiene una tarjeta de red Mellanox ConnectX® -5 VPI EDR IB (100 Gb/s) dual-port. La mitad de los nodos tiene una tarjeta gráfica Tesla T4.

Para las tareas de administración, el clúster cuenta con un servidor con procesador de 8 cores Intel Xeon E5-2630L v3, memoria RAM de 32 GB DDR4 a 1,866 GHz (4 GB RAM por core); dos discos duros de 512 GB y 2 de 2 TB, y conexión a servidor de disco Fiber Channel con 30 TB y conexión a servidor NAS con 20 TB.

La red InfiniBand del clúster tiene 50 switches Mellanox SB7890 y 5 InfiniScale IV 8-Port QSFP "IS5022". La red GigaEthernet tiene 4 switches de 24 puertos (dos Aruba 2530 y otros dos 3Com).

En este Trabajo Fin de Grado se han usado 8 nodos del grupo de 14 mencionado antes. La razón, como se indica a continuación, es que son estos nodos los que están conectados a la PDU disponible en el clúster.

PDU

El clúster CELLIA dispone de una PDU en uno de sus armarios. Se trata del modelo ePDU G3 Metered Input de la empresa EATON. Tiene las siguientes características principales:

- 24 enchufes.
- Voltaje de 230 V.
- Sistema monofásico (single-phase).
- Interfaz Ethernet.
- Supervisión local mediante LCD.
- Supervisión remota.
- Protocolos de comunicación: HTTP, HTTPS, SSL, Telnet, FTP, SNMP, SMTP, DNS, DHCP, LDAP, RADIUS.

En la figura 5.1 se muestran los nodos y switches que están conectados a la PDU. Los 10 nodos y 4 de los 5 switches forman la topología que se muestra en la figura 5.2.



Figura 5.1: Nodos y switches conectados a la PDU.

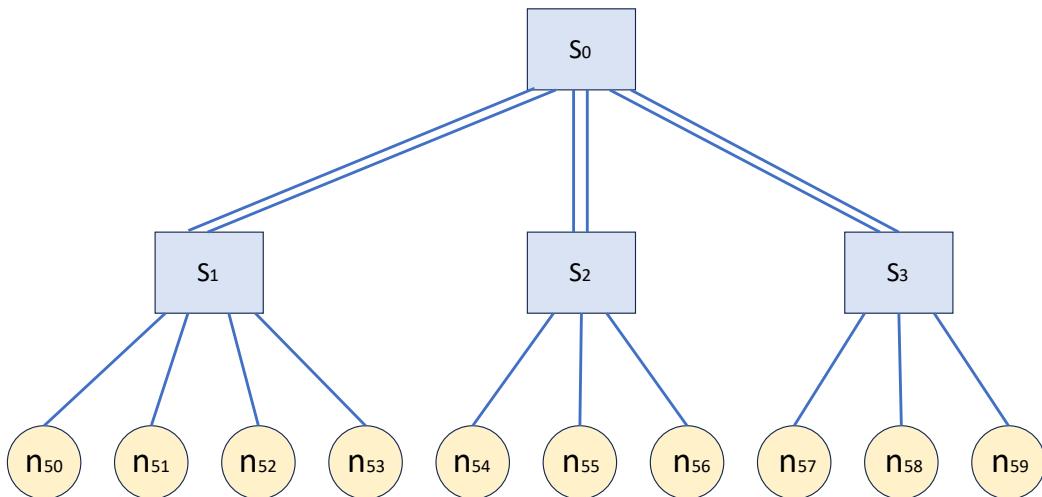


Figura 5.2: Topología de una subred del clúster CELLIA.

5.1.2. Software

Durante el desarrollo del trabajo se ha utilizado diverso software, en algún caso ya disponible en los equipos usados y en su mayoría requiriendo su instalación en dichos equipos.

Aplicaciones HPC

- **GADGET 4.** Es una aplicación utilizada para simular y modelar la formación y evolución de galaxias y estructuras cosmológicas [36]. Es ampliamente utilizado en la investigación en astronomía y cosmología y se puede ejecutar en sistemas con un solo procesador o en un clúster HPC. Es un código flexible que se puede aplicar a distintos tipos de simulaciones, ofreciendo un largo número de complejos algoritmos. Excepto en problemas de prueba muy simples, normalmente se utilizan una o más opciones de código especializadas, denominados módulos, que se activan mediante opciones de compilación. Cada módulo puede tener varias opciones de compilación opcionales y/o valores de archivo de parámetros.

La simulación comienza con la definición de un conjunto de condiciones iniciales, que incluyen la posición, velocidad, masa y otros parámetros de las partículas que se simulan. A partir de ahí, el programa simula la evolución dinámica de las partículas a lo largo del tiempo, teniendo en cuenta las interacciones gravitatorias y otros procesos físicos que pueden influir sobre la evolución de las partículas [34].

- **GRAPH500.** Esta aplicación realiza tareas de procesamiento de grafos, una tarea muy común en una amplia gama de aplicaciones, incluyendo análisis de datos, inteligencia artificial y ciencias de la computación [5]. La aplicación tiene múltiples técnicas de análisis que acceden a una única estructura de datos que representa un grafo ponderado no dirigido. Además de un núcleo para construir el grafo a partir de la lista de tuplas de entrada, hay dos núcleos computacionales adicionales para operar en el grafo. El primero realiza una búsqueda amplia del grafo, y el segundo realiza múltiples cálculos del camino más corto de una sola fuente en el grafo. Los tres núcleos están cronometrados. Hay varias clases de problemas definidos por su tamaño de entrada: toy (17 GB) o alrededor de 10^{10} bytes, que también se llama nivel 10; mini (140 GB) 10^{11} bytes, level 11; small (1 TB) 10^{12} bytes, level 12; medium (17 TB) 10^{13} bytes, level 13; large (140 TB) 10^{14} bytes, level 14); y huge (1.1 PB) 10^{15} bytes, level 15 [1].
- **GROMACS.** Es una aplicación de simulación de materiales diseñada para modelar sistemas biológicos y químicos a gran escala. Se utiliza para simular proteínas, ácidos nucleicos, lípidos y otras moléculas, así como sus interacciones y dinámica. Utiliza técnicas de simulación molecular, incluyendo dinámica molecular, dinámica estática, termodinámica Monte Carlo y optimización de estructuras. Es ampliamente utilizada en la investigación en biología molecular, química computacional y ciencias de materiales. Los núcleos de cálculo están escritos utilizando intrínsecos SIMD para CPU, y CUDA, OpenCL y SYCL para GPU. Se puede ejecutar en paralelo, utilizando el protocolo de comunicación MPI estándar, o a través de una biblioteca propia Thread MPI para estaciones de trabajo de un solo nodo. Existen opciones para equilibrar estática y dinámicamente la carga entre los distintos recursos. [12].

- **HPCG** (High Performance Conjugate Gradients). Es un software de benchmarking y medición de rendimiento para sistemas HPC. Es una alternativa a HPL que se utiliza para evaluar el rendimiento de sistemas de supercomputación y establecer la lista top500. HPCG realiza una serie de cálculos de optimización lineal en una matriz de tamaño específico y mide el tiempo que tarda en completarlos. Las operaciones que realiza son del tipo: multiplicación matriz dispersa por vector, actualización de vectores, multiplicación escalar, suavizador local simétrico de Gauss-Seidel, algoritmo del gradiente conjugado. La implementación de referencia está escrita en C++ con soporte MPI y OpenMP. HPCG es considerado un indicador más preciso del rendimiento real de un sistema HPC para tareas científicas y técnicas específicas [14].
- **LAMMPS** (Large-scale Atomic/Molecular Massively Parallel Simulator). Es un software de simulación de materiales, diseñado para modelar sistemas atómicos y moleculares a gran escala. Simula una amplia gama de materiales, incluyendo sólidos, líquidos, polímeros, biopolímeros, metales, materiales compuestos y materiales cerámicos, entre otros. LAMMPS permite la modelización de sistemas termodinámicos y mecánicos complejos, utilizando una variedad de enfoques de simulación, incluyendo dinámica molecular, dinámica estática, termodinámica Monte Carlo y optimización de estructuras. En el sentido más general, LAMMPS integra las ecuaciones de movimiento de Newton para una colección de partículas que interactúan. Una partícula puede ser un átomo, una molécula o un electrón, un grupo de átomos de grano grueso, o un grupo mesoscópico o macroscópico de material. Los modelos de interacción que incluye LAMMPS son en su mayoría de corto alcance, aunque también se incluyen algunos de largo alcance. Aunque se puede ejecutar en ordenadores portátiles o de sobremesa, está diseñado para computadores paralelos, con soporte de paso de mensajes MPI. Algunas partes también soportan OpenMP multithreading, vectorización y aceleración con GPU [24]

Otro software

- Sistema operativo Ubuntu Server 20.04.4 LTS. Es la distribución Linux instalada en el clúster CELLIA.
- OpenMPI. Distribución MPI instalada en el clúster. Se han ejecutado las versiones paralelas de las aplicaciones HPC, que usan esta biblioteca de comunicaciones.
- Intel PCM. Para obtener datos de consumo energético de los nodos del clúster desde los contadores en los procesadores Intel del clúster.
- Protocolo SNMP. Utilizado para obtener los datos de consumo energético del clúster a través de la PDU.
- Cacti, MIB Browser. Aplicaciones para conectar con la PDU y obtener los datos de consumo.
- GnuPlot. Aplicación para elaborar las gráficas con los resultados obtenidos en el trabajo.

5.2. Métricas

Por las características del trabajo, los datos que se van a manejar son los referidos al consumo energético en un clúster. Tanto con la PDU como con la utilidad Intel PCM se pueden obtener datos de consumo de potencia y de energía. Puesto que existe una relación entre ambas, sólo se va a usar una de ellas.

Hay que señalar que el tiempo de ejecución de cada aplicación depende de la configuración que se establezca a través de un conjunto de parámetros específico de cada una, y aunque se podría forzar a que todas las aplicaciones tuvieran el mismo tiempo total de la ejecución, no parece lo más adecuado. Pero además, el tiempo no es esencial en este trabajo, puesto que el estudio tiene carácter comparativo de dos métodos de medida, y no tanto de rendimiento. Por tanto, en este trabajo se usará el consumo de potencia, medido en vatios.

Como se ha indicado en la sección 4.3, los datos de potencia recogidos por las dos vías consideradas en el trabajo (PDU y pcm-power) no son del mismo orden pues la PDU ofrece los datos de consumo de todos los elementos conectados a ella. Lógicamente, ésto se tendrá en cuenta, y los datos serán tratados en la forma adecuada.

5.3. Configuración de las pruebas

Se han realizado las siguientes pruebas:

- Considerando un único nodo, se ha variado el número de tareas, y como consecuencia el número de cores usados. Se busca determinar la contribución de cada core de la CPU al consumo total.
- Considerando un número fijo de tareas por nodo, se ha variado el número de nodos. La intención es determinar la contribución de cada nodo al consumo total.
- Variando el número de nodos se ha variado el número de puertos con actividad de un switch. El objetivo es determinar el consumo de cada puerto del switch.

En todos los casos se han utilizado las cinco aplicaciones, y la duración de cada ejecución ha sido aproximadamente la misma en todos los casos. En todas las pruebas, excepto en las que se mide el consumo de los switches, se han usado los dos métodos de medida. En el caso de los switches sólo con la PDU. Para cada prueba individual, se han realizado 10 ejecuciones, obteniendo la media.

5.4. Resultados y análisis

Para obtener los datos de consumo, el procedimiento seguido consiste básicamente en ejecutar simultáneamente la aplicación que recoge los datos de consumo y la aplicación HPC en los nodos que se deseé. En realidad, la aplicación que mide el consumo comienza su ejecución unos segundos antes y termina unos segundos después del comienzo y finalización de la ejecución de la aplicación HPC, respectivamente. Para ello se usan los scripts que se incluyen en la sección A.2. La evolución del consumo a lo largo del tiempo es el que se muestra en la figura 5.4. El consumo sube cuando comienza la ejecución de la aplicación HPC, se mantiene más o menos constante durante dicha ejecución y vuelve a bajar al finalizar.

5.4.1. Consumo de los cores de la CPU

En la figura 5.4 se muestra, para cada aplicación HPC, el consumo de potencia de un nodo del clúster en función del número de tareas que se ejecutan, es decir, del número de núcleos que se usan. Se observa que un mayor número de tareas se traduce en un mayor consumo, a excepción de un par de situaciones en las que esto no ocurre. También se puede ver, con los números (tabla 5.1), que el incremento del consumo por cada tarea no es constante. Indicar que las barras desaparecidas en dos gráficas es debido a casos no contemplados por la aplicación o errores en la ejecución.

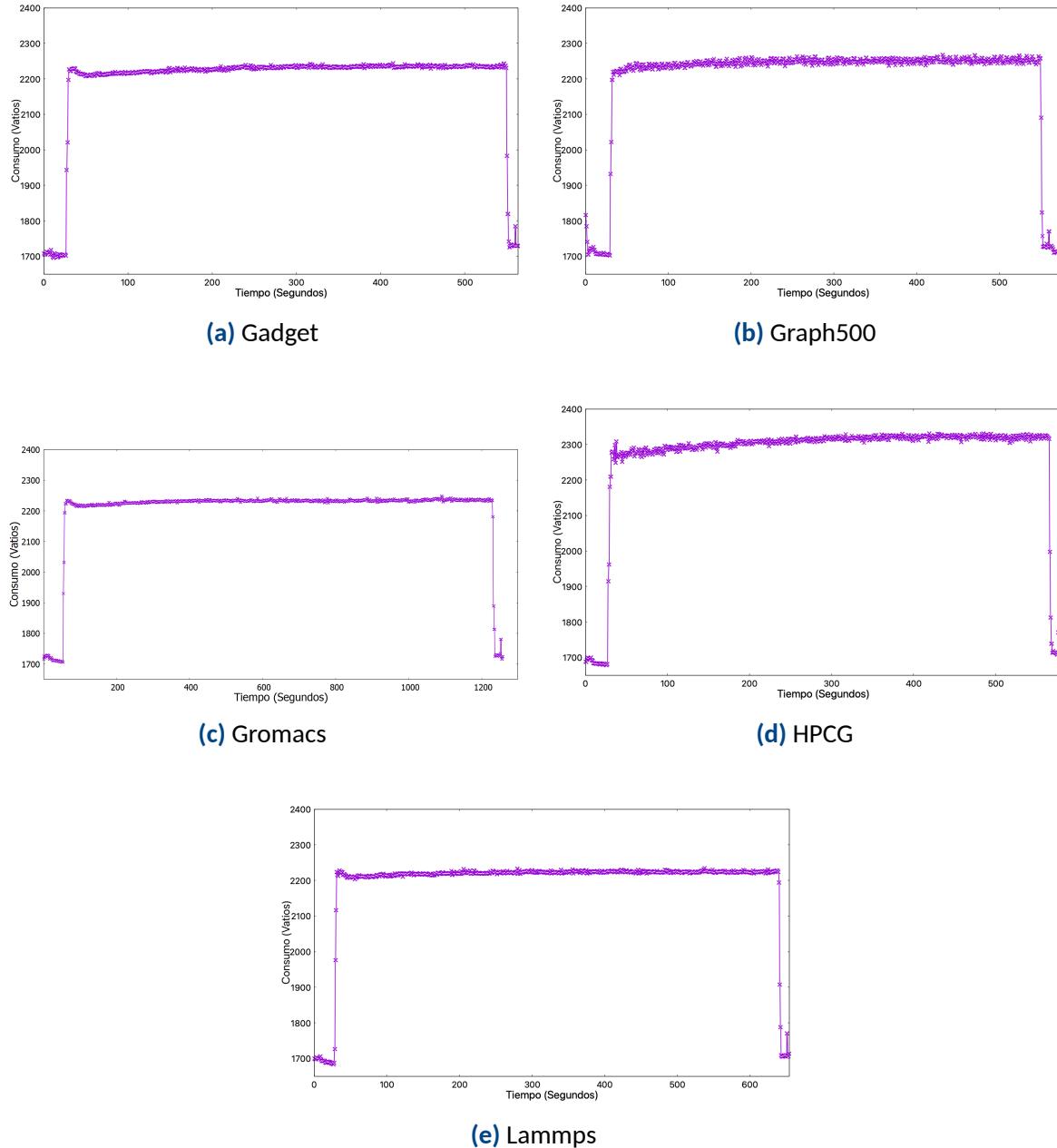


Figura 5.3: Evolución del consumo a lo largo del tiempo. Obtenido con la PDU para 16 tareas y 8 nodos. Obteniendo las medidas cada 0.5s segundos.

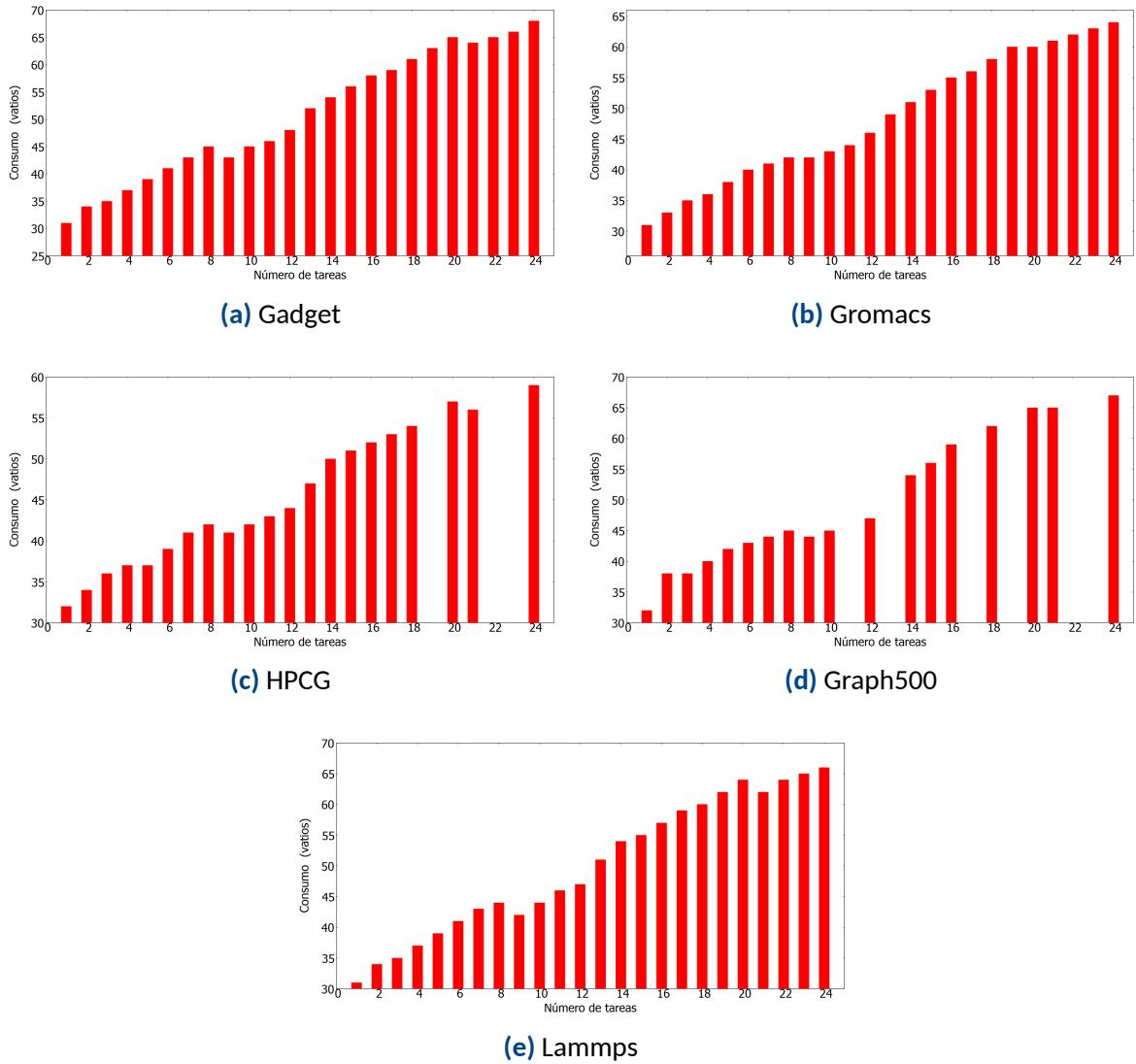


Figura 5.4: Consumo de un nodo en función del número de tareas. Medido con pcm-power.

Por otro lado, en la figura 5.5, se muestran datos en la misma línea que los anteriores, pero en este caso esos datos han sido recogidos con la PDU. Como ya se apuntó en la sección 4.3.1, la PDU registra el consumo global, de todos los elementos conectados a ella, y por tanto los valores que se muestran en las gráficas son un orden de magnitud superior. Sin embargo, lo importante son los incrementos que, al igual que con pcm-power, no se mantienen totalmente constantes. Se observa, igual que con pcm-power, que el consumo aumenta con el número de tareas, pues aumenta en la misma cantidad el número de núcleos de las dos CPUs del nodo que se usan para ejecutar esas tareas.

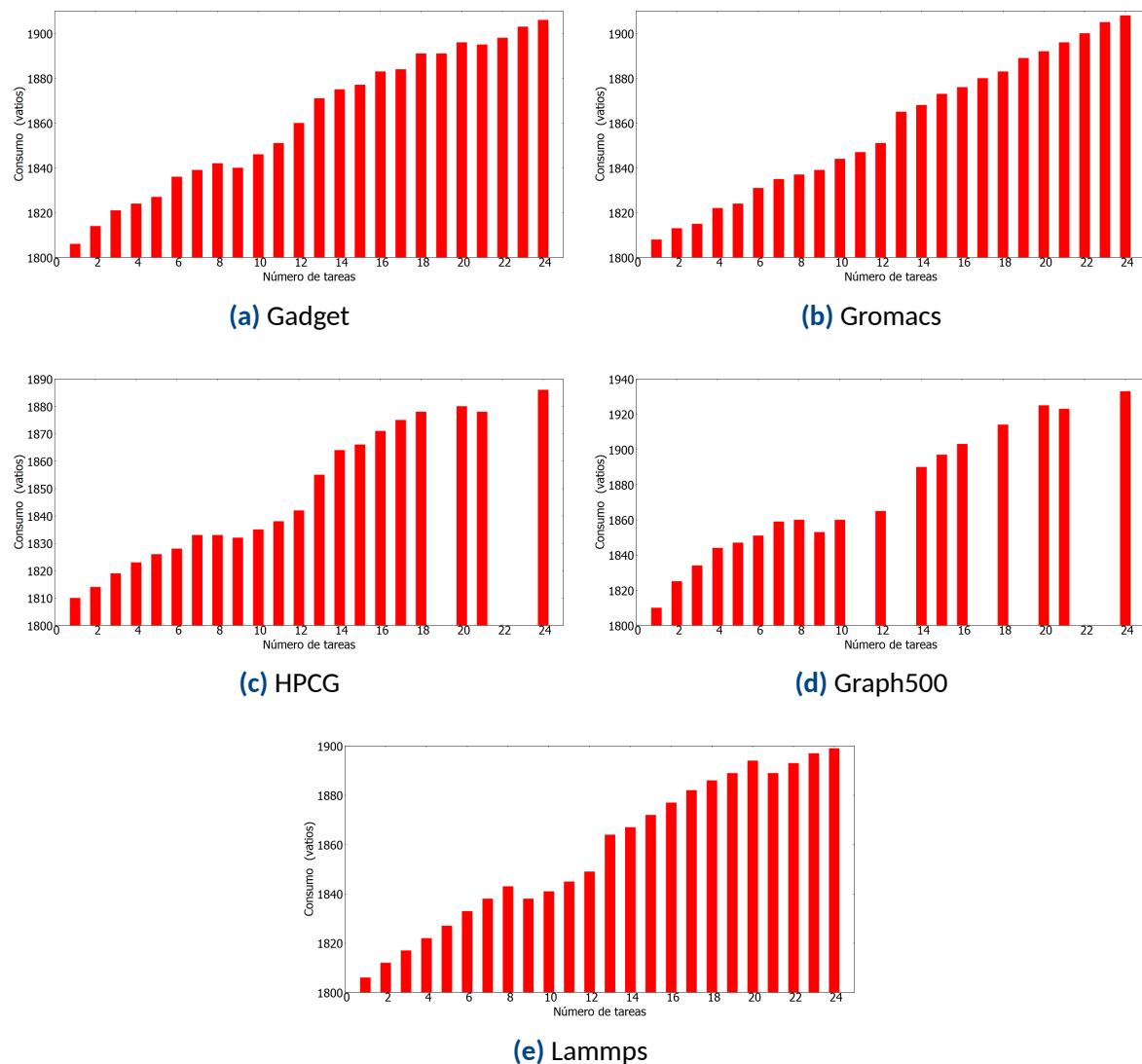


Figura 5.5: Consumo de un nodo en función del número de tareas. Medido con la PDU.

Los datos numéricos con los que se han creado las gráficas de la figura 5.4 se han recogido en la tabla 5.1. Con esos datos se aprecia mucho mejor lo indicado anteriormente respecto al incremento del consumo del nodo cuando se aumenta en uno el número de tareas.

Tabla 5.1: Número medio de vatios de un nodo en función del número de tareas cuando las medidas se toman usando Intel PCM.

Tareas	Aplicación				
	Gadget4	Gromacs	HPCG	Graph500	Lammps
0	13,236	14,124	14,231	12,932	13,023
1	31,698	32,359	32,409	31,238	31,682
2	34,452	34,413	38,047	33,423	34,123
3	35,957	36,129	38,067	35,140	35,770
4	37,911	37,534	40,531	36,649	37,860
5	39,603	37,778	42,140	38,265	39,202
6	41,654	39,796	43,777	40,159	41,224
7	43,801	41,255	44,780	41,827	43,041
8	45,220	42,472	45,853	42,813	44,834
9	43,702	41,646	44,888	42,334	42,947
10	45,163	42,548	45,804	43,326	44,606
11	46,856	43,632	-	44,758	46,330
12	48,408	44,703	47,855	46,082	47,719
13	52,251	44,953	-	49,862	51,797
14	54,820	-	54,9797	51,981	54,087
15	56,302	-	56,821	53,643	55,624
16	58,366	52,526	59,186	55,266	57,468
17	59,703	53,494	-	56,725	59,451
18	61,752	-	62,846	58,507	60,220
19	63,819	-	-	60,209	62,719
20	65,223	57,461	65,944	60,911	64,001
21	64,072	56,4878	65,216	61,238	62,366
22	65,604	-	-	62,022	64,162
23	66,973	-	-	63,479	65,676
24	68,724	59,945	67,654	64,735	66,258

Cabría esperar un incremento constante, pero no es así. Los incrementos oscilan, por ejemplo, entre 0,3 y 3,8 para Graph500 o entre 0,6 y 4 para Lammps, siendo similares para las otras aplicaciones. Hay que indicar que las aplicaciones HPC se han lanzado a ejecución de tal forma que cada tarea es asignada a un core distinto, y por tanto añadir una tarea significa añadir un core más a la ejecución. Por tanto, debe haber otras razones, relacionadas con aspectos de funcionamiento tanto de la propia aplicación como del hardware, que justifiquen esas variaciones.

En esta línea, se puede también observar que cuando el nodo no está ejecutando la aplicación tiene un consumo entre 12 y 14 vatios, y cuando se lanza con una tarea sufre un incremento que es mucho mayor que el que se produce cuando se añade una tarea más. Esto invita a pensar que, aunque la aplicación se ejecute con una única tarea, se activa un conjunto de recursos, al margen del núcleo, que hacen subir el consumo más allá del provocado por dicho núcleo. Es decir, para comprender estos, y los anteriores, resultados es necesario profundizar en el análisis y realizar muchas más pruebas.

Un análisis similar se puede hacer con los datos numéricos usados para generar las gráficas de la figura 5.5, y que se han recogido en la tabla 5.2, correspondientes a las medidas realizadas con la PDU.

Tabla 5.2: Número medio de vatios de un nodo en función del número de tareas cuando las medidas se toman usando la PDU.

Tareas	Aplicación				
	Gadget4	Gromacs	HPCG	Graph500	Lammps
0	1671,231	1673,101	1673,024	1672,234	1671,967
1	1806,164	1810,612	1810,833	1808,613	1806,870
2	1814,969	1814,700	1825,467	1813,449	1812,881
3	1821,307	1819,411	1834,256	1815,735	1817,462
4	1824,490	1823,296	1844,037	1822,831	1822,710
5	1827,442	1826,479	1847,737	1824,706	1827,441
6	1836,843	1828,748	1851,200	1831,740	1833,157
7	1839,186	1833,775	1859,058	1835,938	1838,499
8	1842,148	1833,756	1860,426	1837,074	1843,069
9	1840,147	1832,463	1853,594	1839,237	1838,657
10	1846,832	1835,730	1860,618	1844,160	1841,118
11	1851,959	1838,648	-	1847,982	1845,817
12	1860,596	1842,982	1865,346	1851,127	1849,941
13	1871,709	1855,323	-	1865,769	1864,203
14	1875,595	1864,230	1890,082	1868,394	1867,997
15	1877,370	1866,729	1897,223	1873,248	1872,392
16	1883,134	1871,151	1903,465	1876,120	1877,253
17	1884,842	1875,975	-	1880,865	1882,517
18	1891,862	1878,804	1914,473	1883,521	1886,013
19	1891,974	-	-	1889,673	1889,918
20	1896,555	1880,338	1925,488	1892,449	1894,178
21	1895,033	1878,636	1923,794	1896,797	1889,016
22	1898,557	-	-	1900,810	1893,196
23	1903,093	-	-	1905,496	1897,181
24	1906,731	1886,504	1933,234	1908,321	1899,259

En este punto, y dado que no se ha obtenido una uniformidad en la contribución al consumo de los cores de la CPU, se ha decidido comprobar si los datos tienen el comportamiento lineal que se intuye a partir de las gráficas. Para ello se ha aplicado regresión lineal con intervalos de confianza del 95%.

Para los datos correspondientes a la aplicación Lammmps (tabla 5.1), se obtiene la gráfica de la figura 5.6, en la que aparece la recta de regresión. Puesto que el coeficiente de determinación R^2 tiene un valor de 0,92 es claro que existe una relación lineal en los datos. Y con los correspondientes coeficientes se obtiene la ecuación de la recta que estima el consumo del nodo de cómputo del clúster en función del número de núcleos de la CPU que se usan:

$$y = 1,83x + 29,54$$

siendo y el consumo y x el número de tareas.

De acuerdo con esa ecuación, el consumo estimado de un nodo de cómputo del clúster CELLIA se incrementa en aproximadamente 2 vatios por cada núcleo de la CPU que se usa en la ejecución de la aplicación Lammmps.

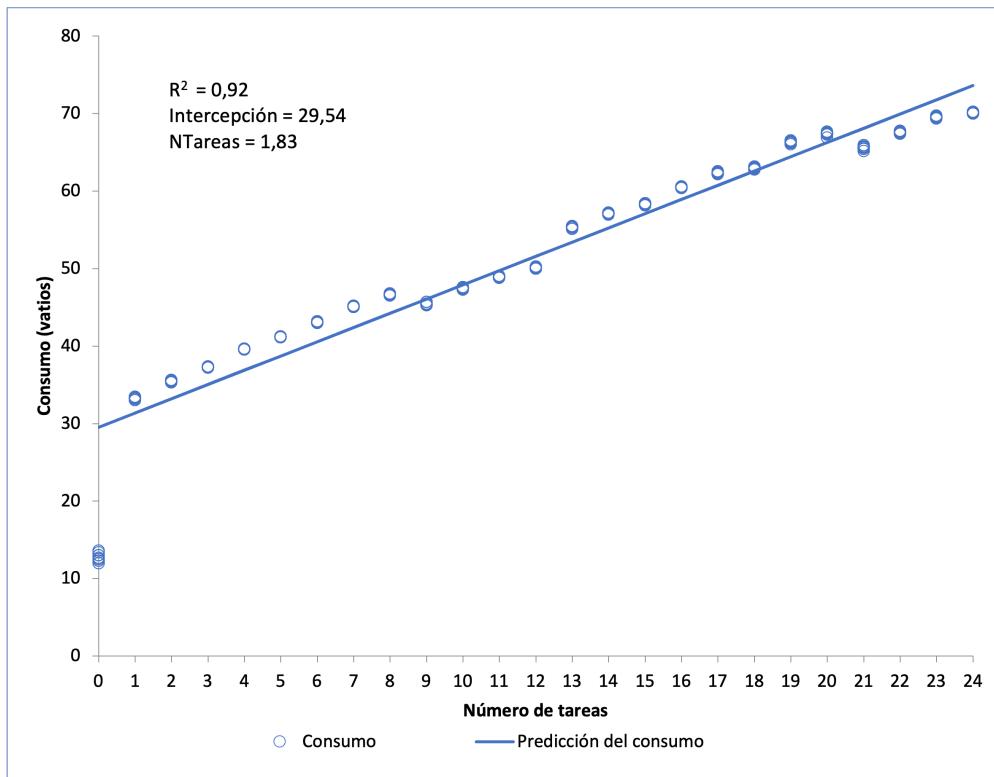


Figura 5.6: Estimación del consumo de un nodo en función del número de cores. Aplicación Lammmps y datos recogidos con Intel PCM.

Con el mismo análisis se obtienen las rectas de regresión para el resto de aplicaciones, obteniéndose resultados muy similares.

Gadget4	$y = 1,79x + 28,19$	con $R^2 = 0,92$
Gromacs	$y = 1,43x + 28,86$	con $R^2 = 0,86$
HPCG	$y = 1,75x + 29,9$	con $R^2 = 0,87$
Graph500	$y = 1,64x + 27,80$	con $R^2 = 0,92$

Se puede realizar un tratamiento idéntico con los datos del consumo obtenidos con la PDU (tabla 5.2). Nuevamente, un valor de 0,85 para el coeficiente R^2 ratifica la relación lineal entre el número de núcleos usados y el consumo producido. En la figura 5.7 se muestra la recta de regresión y los coeficientes, mediante los cuales se obtiene su ecuación. Volver a señalar que los en este caso son menos precisos puesto que la PDU recoge datos de otros elementos conectados a ella que pueden también contribuir al consumo medido.

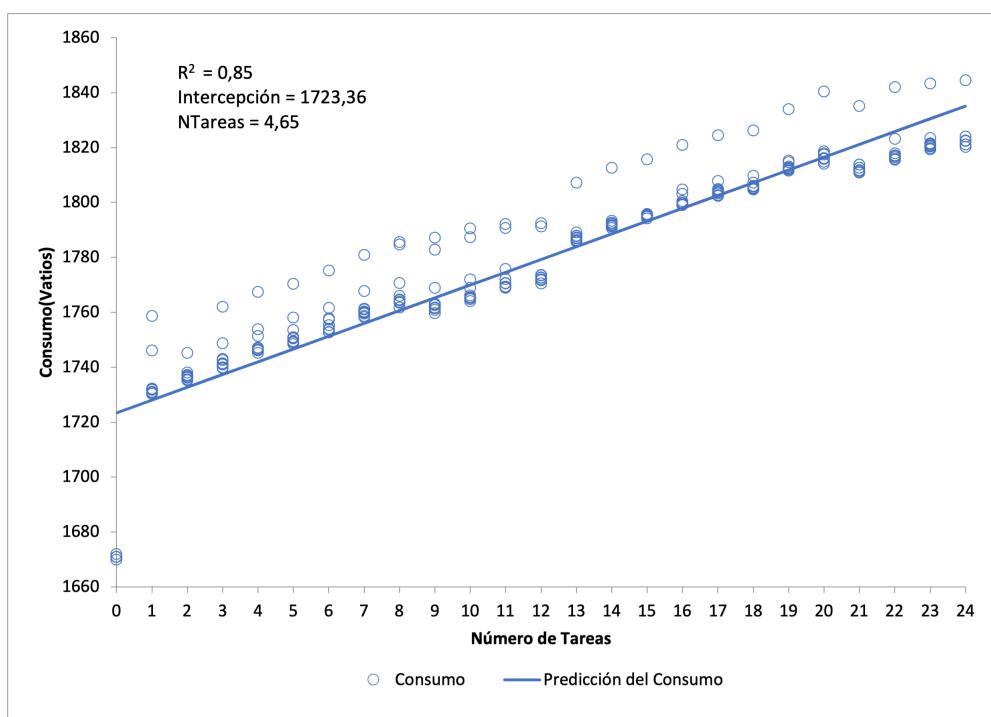


Figura 5.7: Estimación del consumo de un nodo en función del número de cores. Aplicación Lammmps y datos recogidos con la PDU.

Gadget4	$y = 5,6x + 1787,29$	con $R^2 = 0,71$
Gromacs	$y = 5x + 1786,26$	con $R^2 = 0,62$
HPCG	$y = 6,65x + 1793,92$	con $R^2 = 0,7$
Graph500	$y = 5,51x + 1786,97$	con $R^2 = 0,77$
Lammmps	$y = 4,65x + 1723,36$	con $R^2 = 0,85$

5.4.2. Consumo de los nodos del clúster

Se ha realizado otro conjunto de pruebas para determinar el consumo de los nodos de cómputo del clúster CELLIA utilizados en este Trabajo Fin de Grado. Estas pruebas han consistido en ejecutar las aplicaciones HPC utilizando varios nodos, en concreto desde uno hasta 8 nodos. En cada nodo se han considerado dos tareas, una en cada CPU. Los resultados obtenidos con `pcm-power` son mostrados gráficamente en la figura 5.8. Como es lógico, hacer intervenir a más nodos hace aumentar el consumo. Tampoco en este caso los incrementos son constantes, variando entre 28 y 37 vatios, y por tanto no podemos establecer con estos datos un consumo exacto por nodo.

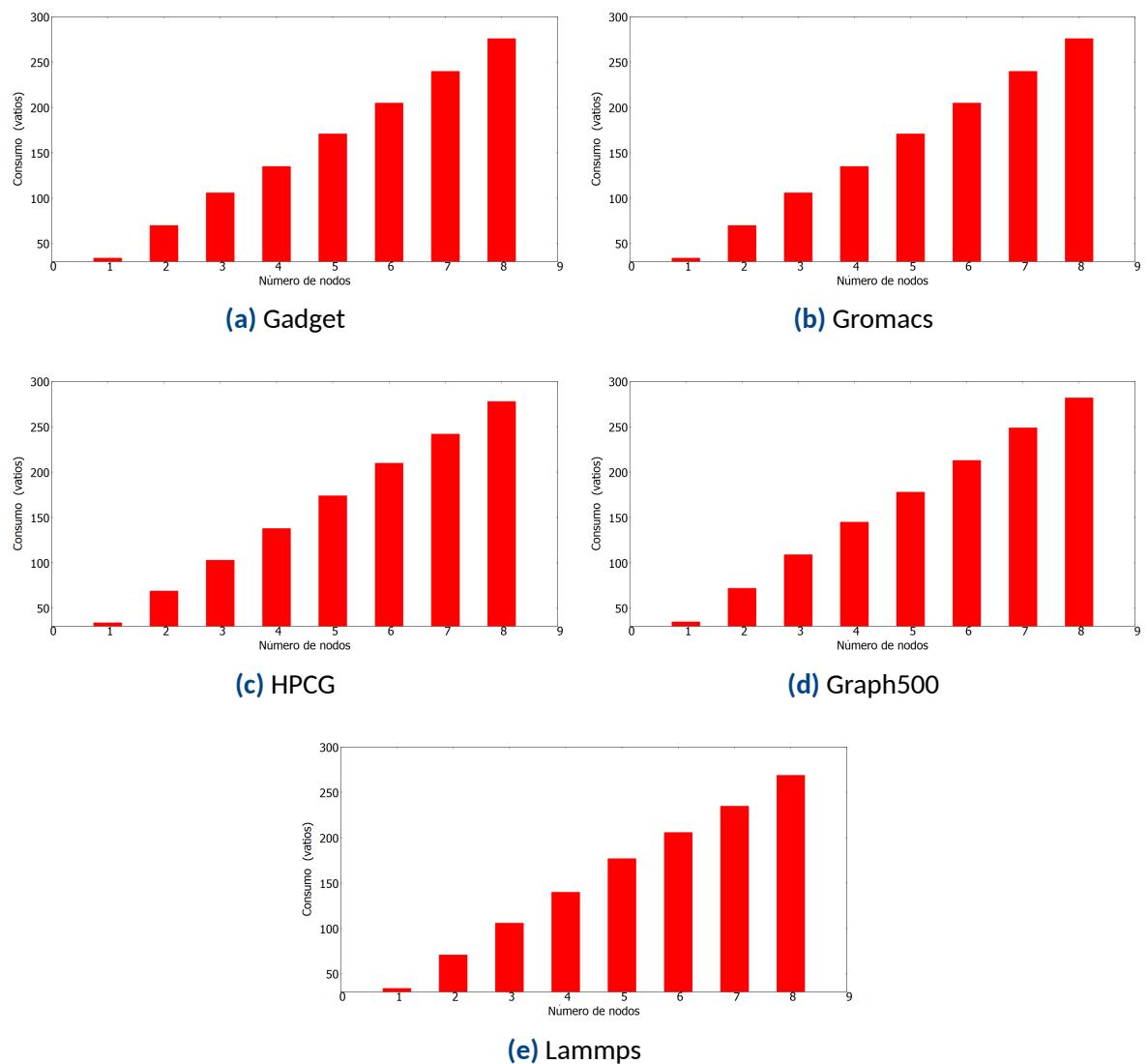


Figura 5.8: Evolución del consumo al variar el número de nodos. Medido con `pcm-power`.

La figura 5.9 incluye las gráficas del consumo variando el número de nodos cuando los datos se han obtenido usando la PDU. En líneas generales, y observando sólo la tendencia y no los valores absolutos, se puede decir que los resultados son similares a los obtenidos con pcm-power. Y también en este caso se produce esa variación en el incremento del consumo al añadir un nuevo nodo.

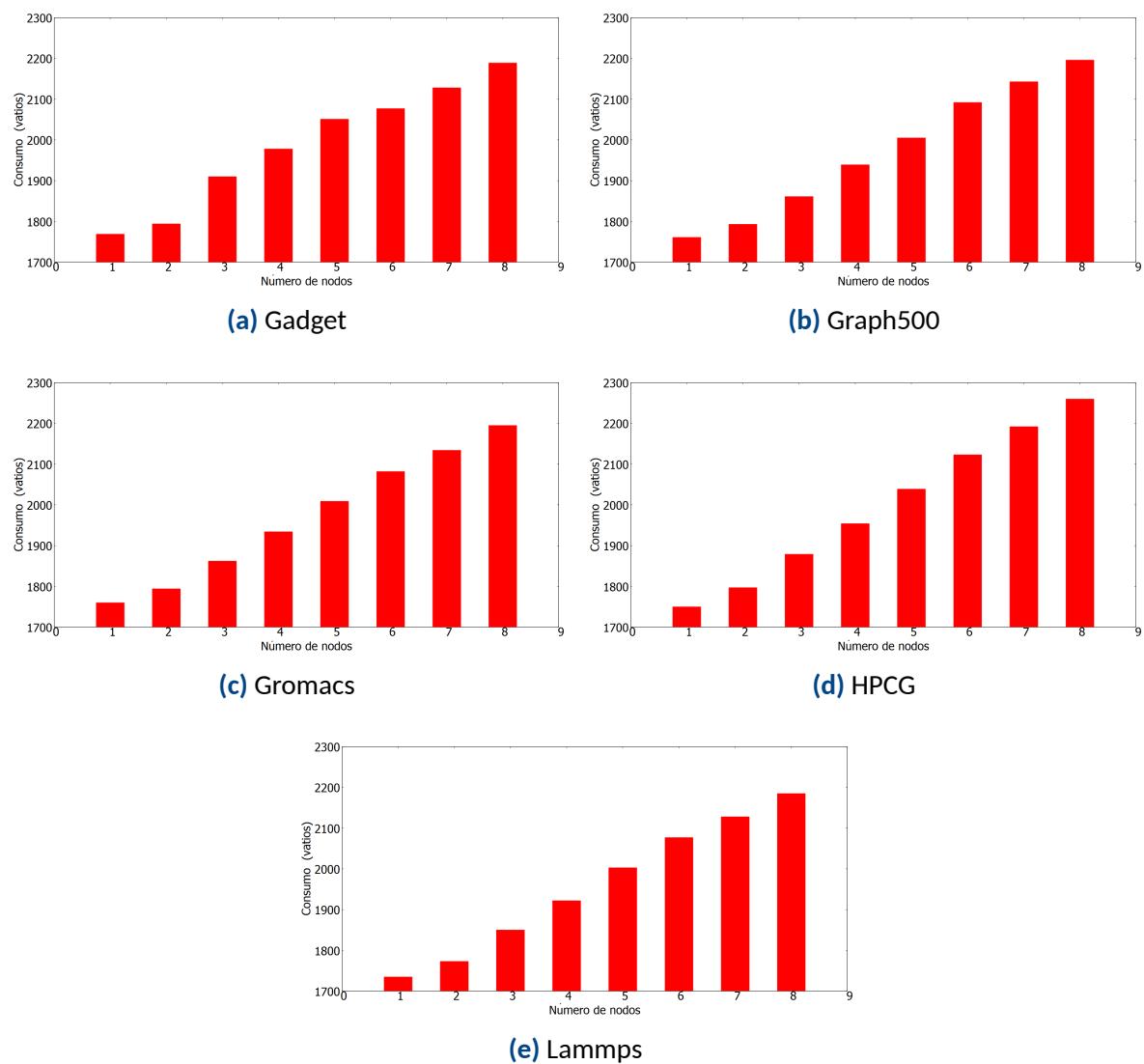


Figura 5.9: Evolución del consumo al variar el número de nodos. Medido con la PDU.

Aunque con menos datos que en el experimento anterior, también para este conjunto de pruebas, se aplica regresión para comprobar si existe una relación lineal entre el número de nodos y el consumo total. Igual que antes, se muestran los detalles de ese análisis para la aplicación Lammmps, y sólo las rectas de regresión para el resto de aplicaciones HPC.

Se aplica regresión lineal con intervalos de confianza del 95%. La gráfica de la figura 5.10 muestra el resultado. Un valor de 0,99 del coeficiente R^2 indica que existe una relación lineal en los datos. La ecuación de la recta mostrada en la figura, obtenida a partir de los coeficientes incluidos en la propia figura, permite estimar el consumo de los nodos del clúster en función del número de éstos usados durante la ejecución de la aplicación.

$$y = 34,34x + 5,32$$

El consumo estimado del clúster debido a los nodos de cómputo se incrementa en aproximadamente 34 vatios por cada nodo más usado en la ejecución de la aplicación Lammmps.

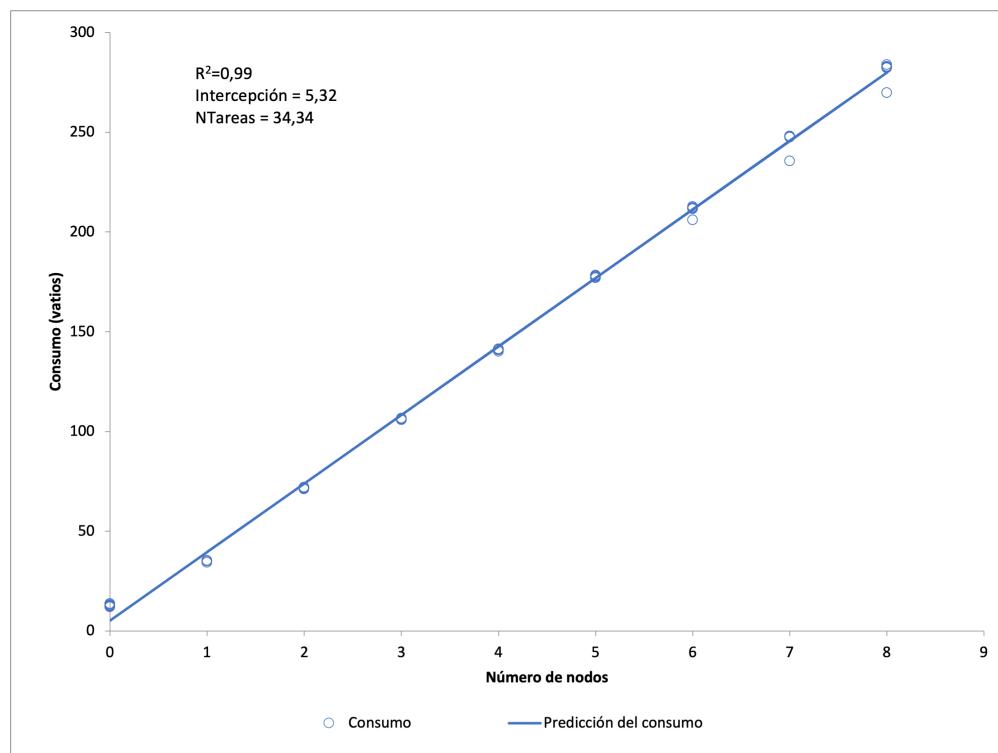


Figura 5.10: Estimación del consumo en función del número de nodos.
Aplicación Lammmps y datos recogidos con Intel PCM.

Para el resto de aplicaciones, las ecuaciones de las rectas tienen resultados muy similares. Estiman valores de consumo para un nodo de entre 33 y 34 vatios.

Gadget4	$y = 33,36x + 5,97$	con $R^2 = 0,99$
Gromacs	$y = 33,86x + 5,14$	con $R^2 = 0,99$
HPCG	$y = 34,45x + 6,99$	con $R^2 = 0,99$
Graph500	$y = 34,44x + 4,30$	con $R^2 = 0,99$

Considerando ahora los datos de consumo recogidos mediante la PDU (gráficas de la figura 5.9), se incluye aquí también la gráfica de la recta de regresión para la aplicación Lammmps (figura 5.11), y sólo las ecuaciones de las rectas para las demás aplicaciones, puesto que el comportamiento es totalmente similar.

Recalcar nuevamente la diferencia en los valores absolutos con aquellos obtenidos con Intel PCM debido a que la PDU registra el consumo de otros elementos al margen de los nodos de computación que intervienen en la ejecución de la aplicación HPC. En cualquier caso, estos resultados sirven para mostrar también esa dependencia lineal de los datos de consumo con el número de nodos.

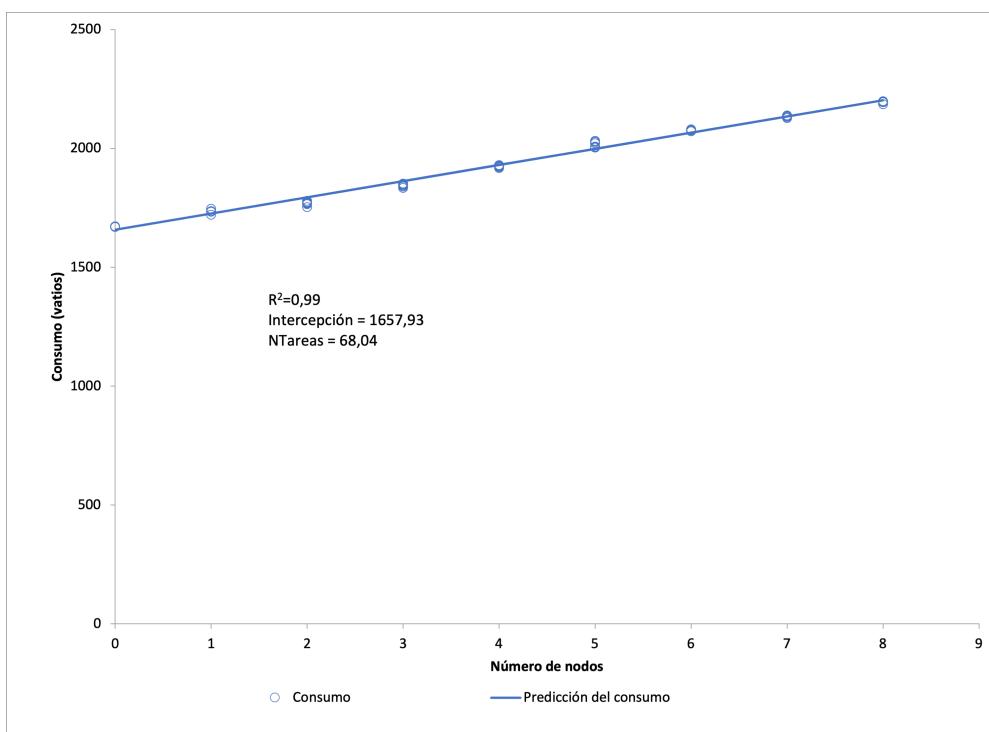


Figura 5.11: Estimación del consumo en función del número de nodos.
 Aplicación Lammmps y datos recogidos con la PDU.

Los resultados para todas las aplicaciones tienen un comportamiento similar, estimando valores de consumo para un nodo de entre 63 y 68 vatios, y de 75 en el caso de HPCG.

Gadget4	$y = 63,33x + 1698,37$	con $R^2 = 0,98$
Gromacs	$y = 65,15x + 1678,60$	con $R^2 = 0,99$
HPCG	$y = 74,34x + 1666,90$	con $R^2 = 0,99$
Graph500	$y = 65,90x + 1677,98$	con $R^2 = 0,99$
Lammps.	$y = 68,04x + 1657,93$	con $R^2 = 0,99$

5.4.3. Consumo de un switch de la subred del clúster

En esta sección se van a mostrar los datos del consumo relativos a un switch de la red que conecta los nodos, obtenidos con la PDU puesto que, como se indicó en la sección 4.3.1, no se puede con Intel PCM. Puesto que no se usan todos los puertos de los switches de la subred que se ha usado en este trabajo, lo que se ha hecho es medir el consumo del switch a nivel de puerto. Concretamente, se ha intentado medir el consumo de un puerto, para lo cual, se ha variado el número de puertos con actividad debida a la ejecución de las aplicación HPC, y se han observado las diferencias en el consumo medido. Al no poder medir con la PDU disponible el consumo de los switches de manera directa, se ha tenido que hacer de esa otra forma.

Así, por ejemplo, ejecutando la aplicación con dos nodos (50 y 51) se usan dos puertos del mismo switch S_1 (figura 5.12a). Ejecutando la aplicación con los nodos 50, 51 y 52 se estarán usando 3 puertos del switch (figura 5.12b). Para conseguir que se usen 4 puertos del switch S_1 , la ejecución se realiza con los tres nodos anteriores y el nodo 53 (figura 5.12c).

Ahora bien, las medidas obtenidas en cada una de esas situaciones incluyen también el consumo de los nodos utilizados. Por tanto, lo que se ha hecho es restar a los valores de consumo obtenidos con la PDU el consumo de los nodos (considerando el resultado obtenido en el primer conjunto de pruebas, sección 5.4.1). De esta forma se puede tener una estimación más aproximada del consumo de los puertos del switch.

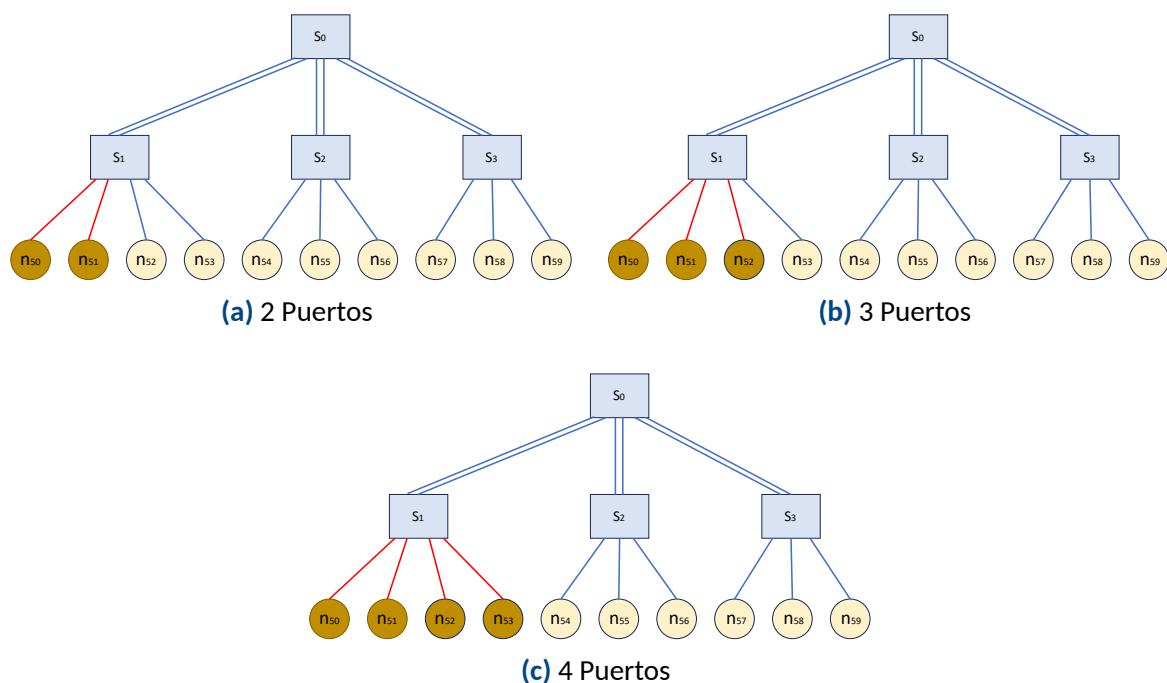


Figura 5.12: Puertos utilizados para medir el consumo de los switches.

Dada la subred utilizada en este trabajo, se ve que el número máximo de puertos de un switch que se podrían usar en las pruebas es sólo 6. Pero teniendo en cuenta el procedimiento indicado para estimar su consumo, se ve también que es complicado obtener resultados incluso para ese número de puertos. Para considerar el switch S_0 , se necesitaría casi todos los nodos, y además se verían implicados todos los switches. Y algo parecido, en menor medida, sucedería si se consideran los puertos de otro de los switches.

Por tanto, las pruebas se han limitado a considerar los casos recogidos en la figura 5.12, implicando así sólo a ese switch y a los nodos conectados directamente a él. Esto simplifica la obtención del consumo de los puertos de un switch, usando el procedimiento antes descrito, si bien es cierto, como ya se ha indicado en varias ocasiones, que no deja de ser un cálculo aproximado por el método de medición utilizado.

Con todas las consideraciones antes mencionadas, se han obtenido los resultados mostrados en la tabla 5.3, y representados gráficamente en la figura 5.13. Puesto que sólo se dispone de tres datos, no se ha aplicado regresión, y la estimación del consumo de un puerto de un switch se obtienen observando la diferencia entre cada pareja de casos consecutivos.

Así, se obtienen estimaciones para el consumo de un puerto de 9 a 11 vatios para Gadget4, 2 a 5 para Gromacs, 3 a 12 para HPCG, 4 a 7 para Graph500, o 13 a 15 para Lammps.

Tabla 5.3: Consumo en función de los puertos activos del switch (PDU).

Tareas	Aplicación				
	Gadget4	Gromacs	HPCG	Graph500	Lammps
2	1693,732	1661,730	1663,897	1648,848	1661,448
3	1704,865	1663,753	1666,886	1656,497	1675,398
4	1714,248	1669,101	1679,604	1661,157	1690,651

5.4.4. Resumen

Aunque los métodos utilizados para medir el consumo de los elementos principales del clúster no ofrecen resultados del todo precisos (especialmente la PDU disponible), se han obtenido estimaciones del consumo de nodos y switches (considerando sus puertos).

Es claro, de los resultados, que en cualquier caso, y a partir de un valor inicial, el consumo aumenta de forma lineal con el número de nodos y puertos activos de los switches.

Aunque lo ideal es recoger datos precisos del consumo de los elementos del clúster, se pueden usar estos resultados para estimar el consumo de sistemas clúster de mayor tamaño.

No se han observado diferencias en el consumo del clúster para las aplicaciones HPC utilizadas. Un conocimiento más profundo de éstas puede permitir plantear otras pruebas, de mayor duración seguramente, que ratifiquen o no este comportamiento.

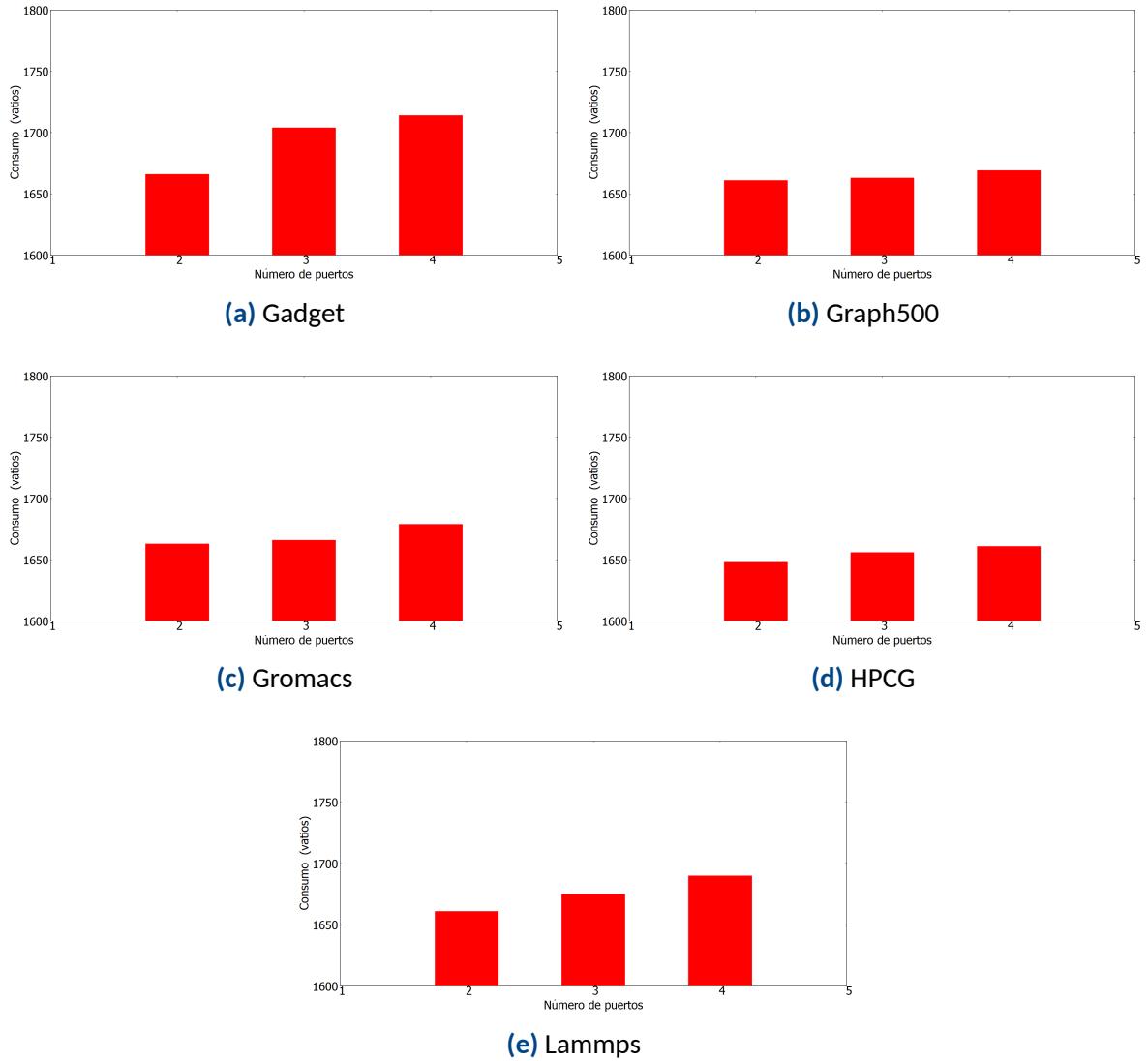


Figura 5.13: Consumo en función del número de puertos activos de un switch. Medido con la PDU.

6. Conclusiones y trabajo futuro

En este capítulo se incluye una lista con las conclusiones más relevantes que se han obtenido a partir de todas las tareas que se han realizado en este Trabajo Fin de Grado. Se proporciona también un conjunto de tareas que se podrían realizar relacionadas con lo mostrado en esta memoria. En la parte final del capítulo se indica cómo se han trabajado las competencias del Grado en Ingeniería Informática indicadas en la sección 2.3, correspondientes a la Tecnología de Ingeniería de Computadores.

6.1. Conclusiones

A partir del conocimiento adquirido durante el desarrollo de este trabajo se han podido extraer algunas conclusiones. Ese conocimiento proviene tanto de la revisión del estado del arte de diferentes cuestiones relacionadas con el consumo de energía de la infraestructura de computación de centros de datos y supercomputación, como por las tareas específicas del trabajo realizado, y que se han centrado en la monitorización de ese consumo.

Se indican a continuación las que se han considerado más relevantes.

- Es necesario crear sistemas energéticamente eficientes. Se hace imprescindible reducir el ritmo de crecimiento del consumo energético en los centros de datos y de supercomputación según éstos siguen aumentando su tamaño. Y no sólo por razones económicas, sino también medioambientales.
- Es necesario disponer de soporte para monitorizar y controlar el consumo. Para tener sistemas más eficientes en términos de consumo, es necesario realizar acciones de control, para lo cual se necesitan herramientas que monitoricen el consumo.
- Se debe elegir el método de monitorización más adecuado. Dependiendo de los objetivos puede resultar más apropiado usar software que recoja datos de los contadores disponibles en el hardware, o usar dispositivos externos de medida. En el primer caso se obtiene información a nivel de componentes, mientras que con las PDUs la información suelen ser a un nivel más global.

-
- Se requiere realizar un estudio más detallado. Para hacer una comparación precisa de los datos de consumo obtenidos mediante los dos métodos usados es necesario realizar más pruebas, sobre todo con la PDU pues hay elementos conectados a ella que no intervienen en el estudio, y sin embargo contribuyen al consumo medido.
 - Se ha observado un comportamiento similar de todas las aplicaciones HPC. Los niveles de consumo recogidos durante la ejecución de todas las aplicaciones consideradas ha sido prácticamente el mismo. Esto puede indicar que la actividad computacional es de igual intensidad en todas ellas.
 - La ausencia de soporte en los elementos de la red de interconexión para registrar datos de consumo ha obligado a tomar las medidas de sus elementos sólo con la PDU, lo cual, dadas sus limitaciones, ha resultado en datos que pueden ser poco precisos. Reorganizando los elementos conectados a la PDU se podrían obtener mejores datos, pero esto no suele ser posible pues el clúster tiene mucha demanda, y no puede pararse y reconfigurarse continuamente.

6.2. Trabajo futuro

Se incluye aquí una lista de tareas que podrían extender el trabajo realizado, con la intención de mejorar algunos de los aspectos que se han tratado o bien de abordar otros que no han sido contemplados en este Trabajo Fin de Grado. Esas tareas son las siguientes:

- Repetir un estudio similar con PDUs managed. Con este tipo de PDU se pueden obtener datos de consumo de energía a nivel de elemento conectado a la PDU. Esto permitirá saber el consumo de un nodo o de un switch.
- Comparar varias herramientas software de monitorización del consumo. En este trabajo se ha usado una única herramienta, y el estudio se puede repetir con alguna más, para determinar cuál sería la más adecuada.
- Comparar varios dispositivos de medición del consumo. Al margen de las PDU, también hay otros dispositivos externos que pueden obtener medidas de consumo. Se trataría de usar varios y buscar el que mejores prestaciones proporcione.
- Realizar un estudio más completo del consumo de la red del clúster. La red ha sido la componente del clúster que menos protagonismo ha tenido en este estudio por dos razones: los elementos disponibles no tienen soporte para recoger datos de consumo, y la PDU disponible no permite obtener datos a nivel de conexión o enchufe individual.
- Buscar justificación a los datos no esperados. Cabía esperar, por ejemplo, que la contribución al consumo de un nodo de cada uno de los núcleos o cores de su CPU fuera el mismo. Esto no se ha observado tal cual en los resultados obtenidos. Se debería profundizar en este aspecto, incluso con un análisis estadístico pormenorizado, para determinar las razones de que eso se haya producido.

6.3. Competencias

Este Trabajo de Fin de Grado nos ha permitido trabajar las competencias mencionadas en el apartado 2.3 de la siguiente manera:

[IC3] Capacidad de analizar y evaluar arquitecturas de computadores, incluyendo plataformas paralelas y distribuidas, así como desarrollar y optimizar software para las mismas. Sobre una plataforma paralela como es un clúster, se ha analizado un aspecto concreto como es su consumo energético.

[IC7] Capacidad para analizar, evaluar, seleccionar y configurar plataformas hardware para el desarrollo y ejecución de aplicaciones y servicios informáticos. Con el estudio que se ha realizado sobre consumo, al haber aprendido a medirlo se está en disposición de poder elegir una determinada plataforma en base a los resultados de este aspecto.

Bibliografía

- [1] Graph500 benchmark. <https://graph500.org/>. Última consulta: 2023-04-13.
- [2] MG-SOFT MIB Browser Personal Edition. <https://www.mg-soft.si/mgMibBrowserPE.html>. Última consulta: 2023/06/26.
- [3] Cacti. Sitio web, Última consulta: 2023/06/21.
- [4] Adaptive Computing Inc. Torque resource manager. <https://adaptivecomputing.com/cherry-services/torque-resource-manager/>. Última consulta: 2022-06-10.
- [5] James Ang, Brian Barrett, Kyle Wheeler, and Richard Murphy. Introducing the graph 500. 2010.
- [6] Javier Balladini, Marina Morán, Dolores Rexachs del Rosario, et al. Metodología para predecir el consumo energético de checkpoints en sistemas de HPC. 2014.
- [7] Jeffrey D Case, Mark Fedor, Martin L Schoffstall, and James Davin. Simple network management protocol (SNMP). Technical report, 1989.
- [8] Centro Extremeño de Tecnologías Avanzadas. CETA Ciemat. <https://www.ceta-ciemat.es/>, Última consulta: 2023/06/15.
- [9] Paweł Czarnul, Jerzy Proficz, Adam Krzywaniak, and Jan Weglarz. Energy-aware high-performance computing: Survey of state-of-the-art tools, techniques, and environments. *Sci. Program.*, jan 2019.
- [10] Jose Duato, Sudhakar Yalamanchili, and Ni Lionel. *Interconnection Networks: An Engineering Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.
- [11] Erik D'Hollander, Jack Dongarra, Ian Foster, Lucio Grandinetti and Gerhard Joubert, editor. *Transition of HPC towards exascale computing*, volume 24 of *Advances in Parallel Computing*. IOS Press, 2013.

-
- [12] GROMACS Development Team. About gromacs, 2021. Última consulta: 2023/04/13.
 - [13] Tatiana Gutiérrez. DAS/NAS/SAN, Última consulta: 2023/06/14.
 - [14] HPCG Benchmark Team. High Performance Conjugate Gradient (HPCG) Benchmark. <https://www.hpcg-benchmark.org/>, 2021. Última consulta: 2023/04/13.
 - [15] Intel. Performance Counter Monitor (PCM). <https://www.intel.com/content/www/us/en/developer/articles/tool/performance-counter-monitor.html>, Última consulta: 2023/06/13.
 - [16] IONOS. Tutorial deSNMP. <https://www.ionos.es/digitalguide/servidores/know-how/tutorial-de-snmp/>, Fecha de acceso: 2023. Última consulta: 2023/06/26.
 - [17] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach*. Pearson, 7th edition, 2016.
 - [18] Tania Marcela Díaz María Mercedes Sinisterra and Erik Giancarlo Ruiz. Clúster de balanceo de carga y alta disponibilidad para servicios web y mail. *Informador técnico*, 76:93–93, 2012.
 - [19] Sandra Luz Martínez and Katerine Taylor. SNMP y RMON. 2005.
 - [20] Douglas Mauro and Kevin Schmidt. *Essential SNMP*. O'Reilly Media, Inc., second edition, 2005.
 - [21] NVIDIA. Introduction to InfiniBand. https://network.nvidia.com/pdf/whitepapers/IB_Intro_WP_190.pdf. Última consulta:: 2022-06-13.
 - [22] Alberto Ortega, Abel Miguel Cano, Juan José Escobar, Miguel Damas, Francisco Montoya, and Jesús González. Sistema de medición de energía para computación de alto rendimiento. 2022.
 - [23] The Astrology Page. Computer cluster, Última consulta: 2023/06/15.
 - [24] Steve Plimpton. Introduction and overview of lammps. https://docs.lammps.org/Intro_overview.html, 2021. Última consulta: 2023/04/13.
 - [25] PowerAPI. PowerAPI. <https://powerapi.org/>, Última consulta: 2023/06/13.
 - [26] Mohammad Rashti, Gerald Sabin, and Boyana Norris. Power and energy analysis and modeling of high performance computing systems using wattpf. In *2015 National Aerospace and Electronics Conference (NAECON)*, pages 367–373, 2015.
 - [27] Mohammad Rashti, Gerald Sabin, and Boyana Norris. Power and energy analysis and modeling of high performance computing systems using wattpf. 2021.
 - [28] Utkarsh Rastogi. A simplified introduction to the architecture of high performance computing. 2019.
-

- [29] Rittal-The System. Power Distribution Unit NG: Para una distribución de corriente segura en racks T. <https://expert.rittal.es/pdu>, Última consulta: 2023/06/21.
- [30] Marshall T. Rose. *The Simple Book: An Introduction to Management of TCP/IP-Based Internets*. Prentice Hall, 1991.
- [31] Oscar Santillán, Huber Gilt, Augusto Ingunza, Kobi Alberto Mosquera Vásquez, and Ivonne Montes Torres. Diseño del sistema hpc-linux-clúster del igp. 2017.
- [32] SchedMD. Slurm workload manager. <https://slurm.schedmd.com/>. Última consulta: 2022-06-10.
- [33] The Internet Society. Rfc 3530 - Network File System (NFS) version 4 Protocol. <https://datatracker.ietf.org/doc/html/rfc3530>. Última consulta: 2022/06/10.
- [34] Volker Springel, Rüdiger Pakmor, Oliver Zier, and Martin Reinecke. Simulating cosmic structure formation with the gadget-4 code. *Monthly Notices of the Royal Astronomical Society*, 506(2):2871–2949, 07 2021.
- [35] Dan Terpstra, Heike Jagode, Haihang You, and Jack Dongarra. Collecting performance data with PAPI-C. In Matthias S. Müller, Michael M. Resch, Alexander Schulz, and Wolfgang E. Nagel, editors, *Tools for High Performance Computing 2009*, pages 157–173, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [36] The Gadget Team. GADGET-4 Documentation. <https://wwwmpa.mpa-garching.mpg.de/gadget4/#documentation>, 2021. Última consulta: 2023/04/13.
- [37] TOP500. GREEN500. <https://www.top500.org/lists/green500/>. Última consulta: 2023/06/01.
- [38] TOP500. TOP500. <https://www.top500.org/lists/top500/2022/06/>. Última consulta: 2023/04/11.
- [39] Ofimática Touza. Esquema de un sistema NAS, Última consulta: 2023/06/13.
- [40] David Trilla. *Disseny i evaluació d'un cluster HPC: Hardware*. PhD thesis, UPC, Facultat d'Informàtica de Barcelona, Jun 2014.
- [41] Huazhe Zhang and Henry Hoffmann. Maximizing performance under a power cap: A comparison of hardware, software, and hybrid techniques. In *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '16, page 545–559, New York, NY, USA, 2016. Association for Computing Machinery.

A. Anexo 1

A.1. Instalación, compilación y ejecución de las aplicaciones HPC.

En este anexo se incluye información necesaria para instalar, compilar y ejecutar las aplicaciones paralelas que se han usado en este Trabajo Fin de Grado. Además de las aplicaciones ha sido necesario instalar en el clúster CELLIA un amplio conjunto de bibliotecas, como por ejemplo openMPI, gsl, fftw3, hdf5, hwloc, vectorclass, entre otras.

A.1.1. Gadget

La aplicación se descarga de la página de gitlab: <https://gitlab.mpcdf.mpg.de/vrs/gadget4>. Una vez descomprimido, se realiza la instalación siguiendo los siguientes pasos:

- El primer paso es situarse en la carpeta src, que contiene los ficheros fuente, distribuidos en distintos subdirectorios.
- Hay dos principales ficheros de configuración, uno de ellos contiene las opciones usadas en tiempo de compilación, y el otro contiene los parámetros usados en tiempo de ejecución. El primero de ellos es Config.sh.
- Para este fichero, hay una plantilla llamada Template-Config.sh a partir de la cual se puede crear el fichero con las opciones que sean convenientes. Una alternativa consiste en emplear uno de los archivos de configuración llamados Config.sh que se proporcionan con los ejemplos de los problemas, y en caso de ser necesario, realizar cambios en él.
- Se utiliza GNU make para controlar el proceso de compilación que se especifica en el archivo Makefile. Con solo ejecutar el comando make, hará que se intente compilar y crear el ejecutable gadget4.

La aplicación trae varios ejemplos, con sus respectivos ficheros de configuración. En este trabajo se ha usado G2-galaxy.

```
#!/bin/bash
#SBATCH --partition=monitoractualizado
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=2
#SBATCH --job-name G2-galaxy
mpirun -np 8 --mca btl tcp,self --mca btl_tcp_if_include eno1
./Gadget4 param.txt
```

Código A.1: Script Slurm para ejecutar la aplicación Gadget4.

Una vez completada la compilación y creado el ejecutable, se pueden lanzar ejecuciones. Para hacerlo y comprobar el correcto funcionamiento, se ha usado Slurm y el subconjunto de nodos del clúster incluidos en una de las colas del sistemas de gestión de procesos. El código A.1 muestra un ejemplo de script.

Como se indica, la partición donde se hacen las pruebas es monitoractualizado, los nodos, 4, y las tareas por nodos, 2. Como un nombre para la ejecución, que tendrá el mismo nombre que el ejemplo G2-galaxy.

Con mpirun se indica el número de procesos, otros parámetros de configuración de la ejecución y la aplicación a ejecutar. Concretamente, con -np 8 se indica el número de procesos (2 por cada uno de los 4 nodos).

La opción:

```
--mca btl_tcp_if_include
```

le indica a OpenMPI qué interfaz de red debe ser utilizada para la comunicación entre nodos. Con eno1, que se refiere a la interfaz de red a utilizar, en este caso será la red ethernet. Es importante elegir la interfaz de red adecuada para la comunicación en un clúster, ya que una mala elección puede afectar negativamente el rendimiento y la escalabilidad de la aplicación MPI.

La aplicación es Gadget4, a la cual se le indican una serie de parámetros de configuración a través del fichero param.txt. Entre otros, se indica formato del archivo, tiempos límites de CPU, características de la ejecución, parámetros cosmológicos, precisión del tiempo de integración, sistema de unidades, densidad inicial estimada, frecuencia de la salida y parámetros de salida, algoritmo de árbol, precisión de la fuerza, frecuencia de actualización del dominio, longitud de reblandecimiento gravitacional.

A.1.2. Graph500

Una vez descargada la aplicación, desde <https://github.com/graph500/graph500> y descomprimida, la compilación es bastante sencilla y sólo requiere incluir las rutas adecuadas en la variable de entorno PATH. Los pasos para la instalación son [1]:

- Situarse en el subdirectorio src, que es el que contiene todos los archivos relacionados con este proceso.
- Modificar el fichero Makefile con los parámetros relacionados al sistema operativo y características del clúster donde se vaya a instalar la aplicación.
- Ejecutar el comando make que se encargará de crear el ejecutable adecuado.

Una vez instalada la aplicación de forma correcta, se ejecuta la aplicación de forma muy similar al resto, utilizando un script Slurm (código A.2), cuyo contenido es más o menos el siguiente:

```
#!/bin/bash
#SBATCH --partition=monitoractualizado
#SBATCH --job-name=graph500
#SBATCH --error=graph500.err
#SBATCH --output=graph500.out
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=1
mpirun -np 4 --mca btl tcp,self graph500_reference_bfs 20
```

Código A.2: Script Slurm para ejecutar la aplicación Graph500.

La partición donde se hacen las pruebas es monitoractualizado. Se usan 4 nodos con un proceso en cada uno. El nombre para la ejecución tiene el mismo nombre que la aplicación, graph500. También se pueden indicar ficheros de error y salida, en este caso con graph500.err y graph500.out respectivamente.

Al ejecutar mpirun se está indicando que se crearán 4 procesos (-np 4), que se usará la red Ethernet, y que la aplicación a ejecutar es "graph500_reference_bfs", al cual se le indica un parámetro, que en este caso es "20". Se refiere al parámetro SCALE que especifica el tamaño de los grafos que se van a procesar, y determina el número de nodos y aristas del grafo. Concretamente, el tamaño del grafo es 2^{SCALE} . Por lo tanto, si se establece SCALE en 20, el grafo tendrá un tamaño de 2^{20} , que es igual a 1.048.576 nodos y 16.777.216 aristas.

A.1.3. Gromacs

Desde el sitio oficial: <https://manual.gromacs.org/current/download.html> se descarga la aplicación y se descomprime, para a continuación compilar siguiendo los siguientes pasos:

- Situarse en la carpeta gromacs.
- Crear el directorio build y cambiarse a éste.
- Ejecutar cmake .. con las siguientes opciones:

```
-DGMX_BUILD_OWN_FFTW=ON -DREGRESSIONTEST_DOWNLOAD=ON  
-DGMX_MPI=on (soporte mpi y ejecutable gmx_mpi).
```

- Luego ya sólo quedará ejecutar make, make check y sudo make install.

Para ejecutar la aplicación se sigue el mismo procedimiento que con las demás aplicaciones, creando un script Slurm, indicando las opciones necesarias. Antes hay que generar un fichero que necesita la aplicación gmx_mpi. Se trata de un archivo tpr que se crea a partir de otros tres, un archivo.mdp, un archivo.gro y un archivo.top. Se usa el siguiente comando:

```
gmx grompp -f archivo.mdp -c archivo.gro -p archivo.top -o archivo.tpr
```

Para el test que se ha ejecutado, se debe acceder a la siguiente carpeta: /gromacs-2022.4/tests/physicalvalidation/systems/ens_argon_md_verlet_pme_vr/input. Ahí se tiene acceso a esos archivos. Para que sea más simple la ejecución se ha dejado el archivo .tpr en la carpeta /build/bin.

Con todo, el aspecto que tiene el script Slurm es el indicado en el código A.3.

```
#!/bin/bash  
#SBATCH --partition=monitoractualizado  
#SBATCH --job-name=gromacs  
#SBATCH --error=gromacs.err  
#SBATCH --output=gromacs.out  
#SBATCH --nodes=4  
#SBATCH --ntasks-per-node=1  
export OMPI_MCA_btl_openib_allow_ib=1  
export OMPI_MCA_btl_openib_if_include="mlx4_0:1"  
mpirun -np 4 --mca btl self,vader,openib ./gmx_mpi mdrun  
ens_argon_md_verlet_pme_vr/system2.tpr -ntomp 1
```

Código A.3: Script Slurm para ejecutar la aplicación Gromacs.

Se indica la partición, el número de nodos, los ficheros de error y salida, y el número de tareas o procesos por nodo. Se introduce algún cambio en este script con respecto a los anteriores para mostrar cómo se usaría la red InfiniBand. Así, la variable de entorno OMPI_MCA_btl_openib_allow_ib se utiliza para permitir el uso de InfiniBand (IB) para la comunicación entre nodos. El valor 1 significa que se permite el uso de InfiniBand.

La variable de entorno OMPI_MCA_btl_openib_if_include se utiliza para especificar la interfaz de red que se utilizará para la comunicación. En este caso, se establece en "mlx4_0:1", lo que significa que se utilizará la primera interfaz Mellanox (mlx4_0) y el puerto 1 (el puerto 0 se identifica como "1" en Open MPI).

El ejecutable a usar en este caso es "gm_mpi". El comando mdrun -s es una opción de gmx_mpi que especifica la entrada de archivo de topología para la simulación. En este caso, la opción -s está seguida de la ruta de acceso al archivo system2.tpr, en el directorio ens_argon_md_verlet_pme_vr/. Este archivo de topología describe los detalles como la estructura molecular, los parámetros de fuerza y las condiciones de simulación. La opción -ntomp 1 indica que no se utilizará multithreading.

A.1.4. Hpcg

La aplicación se descarga de: <https://hpcg-benchmark.org/downloads/hpcg-3.1.tar.gz>, y se siguen los siguientes pasos para la instalación y compilación:

- Situarse en la carpeta /build/bin, donde se encuentran varios ficheros Makefile disponibles para instalar el software, que varían dependiendo del sistema operativo y configuración del dispositivo donde se vaya a instalar la aplicación.
- En este trabajo se usa el fichero Make.Linux_MPI.
- Se ejecuta el comando make con el parámetro arch = Linux_MPI.

Para ejecutar la aplicación se usa un script Slurm con el siguiente aspecto (A.4):

```
#!/bin/bash
#SBATCH --partition=monitoractualizado
#SBATCH --job-name=xhpcg
#SBATCH --error=hpcg.err
#SBATCH --output=hpcg.out
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=2
mpirun -np 8 --display-map --mca btl tcp,self --mca
btl_tcp_if_include eno1 ./xhpcg hpcg.dat
```

Código A.4: Script Slurm para ejecutar la aplicación HPCG.

Se selecciona la partición monitoractualizado, 4 nodos y 2 tareas por nodo. El nombre para la ejecución, que tendrá el mismo nombre que el ejecutable, xhpcg. También se indican los ficheros de error y salida con hpcg.err y hpcg.out, respectivamente.

La opción -np 8 de mpirun indica los 8 procesos a crear, y la opción

```
--mca btl_tcp_if_include
```

indica a OpenMPI qué interfaz de red debe ser utilizada para la comunicación entre nodos, con la opción eno1, que se refiere a la interfaz de red a utilizar, en este caso será la red Ethernet. El ejecutable propiamente dicho es el xhpcg.

El archivo hpcg.dat incluye información sobre la geometría y topología de la malla, la cantidad de memoria disponible en el sistema, las tolerancias y otros parámetros específicos del benchmark hpcg. Este archivo hpcg.dat permite a los usuarios ajustar y personalizar el rendimiento de la aplicación en función de las características de su sistema de computación.

A.1.5. Lammmps

Comienza el proceso de instalación clonando el repositorio desde su github con
git clone -b release <https://github.com/lammps/lammps.git> mylammps. Para la instalación se siguen los siguientes pasos:

- En este caso se usa la aplicación cmake, pues el proceso es mucho más sencillo que con make, así que hay que asegurarse de tenerla instalada.
- En el directorio mylammps, directorio inicial de este software, se crea un nuevo directorio con el nombre build.
- En el directorio build se ejecutan dos comandos:
`cmake .../cmake`
`cmake - build`
- Por último, se ejecuta el comando "make install".

Si todo va bien se creará el fichero ejecutable lmp, que se podrá lanzar con un script Slurm con el aspecto del código A.5.

Se selecciona la partición monitoractualizado, 4 nodos y 1 tarea por nodo. La opción -np 8 de mpirun indica los 8 procesos a crear. Se indican los ficheros de error y salida con lammps.err y lammps.out, respectivamente.

Con export LD_LIBRARY_PATH=/usr/lib/x86_64-linux-gnu/:\$LD_LIBRARY_PATH se añade el directorio /usr/lib/x86_64-linux-gnu/ a la variable de entorno LD_LIBRARY_PATH. Esta variable es usada por el sistema operativo Linux para buscar bibliotecas que se requieren para ejecutar programas. En este caso, el directorio agregado es /usr/lib/x86_64-linux-gnu/.

```

#!/bin/bash
#SBATCH --partition=monitoractualizado
#SBATCH --job-name=lammps
#SBATCH --error=lammps.err
#SBATCH --output=lammps.out
#SBATCH --ntasks-per-node=1
#SBATCH --nodes=4
export LD_LIBRARY_PATH=/usr/lib/x86_64-linux-gnu/:
$LD_LIBRARY_PATH
mpirun --mca btl tcp,self $HOME/.local/bin/lmp -in examples/
KAPPA/in.mp

```

Código A.5: Script Slurm para ejecutar la aplicación Lammps.

Se ejecuta la aplicación lmp, en este caso indicando la ruta completa del directorio donde se encuentra (`$HOME/.local/bin/lmp`). Finalmente, la configuración de la ejecución de la aplicación lammps se establece por medio del fichero `in.mp` que se encuentra en el directorio `examples/KAPPA/`.

A.2. Ejecución de las aplicaciones que registran el consumo

En esta sección se muestra cómo se han recogido los datos de consumo. Puesto que los nodos que están conectados al clúster mediante la PDU no están incluidos en ninguna cola del sistema gestor de tareas (Slurm), no se ha usado éste para lanzar las aplicaciones HPC y las que registran el consumo de éstas. Así pues, la ejecución se ha realizado usando directamente mpirun. Eso sí, se han usado scripts para ello.

El proceso consiste básicamente en lanzar en segundo plano la aplicación que recoge los datos de consumo en los nodos que intervienen en la ejecución de la aplicación HPC. A continuación, con un cierto retraso, se lanza dicha aplicación en esos nodos, de tal forma que se registra el consumo durante la ejecución de la aplicación HPC. Una vez que ésta finaliza, y tras un tiempo breve, se fuerza la terminación de la aplicación que mide el consumo.

A continuación se muestra un sencillo ejemplo de un archivo que ejecuta la aplicación Graph500, corriendo en un nodo, el 51 concretamente, con 2 tareas para este nodo, en dos cores distintos (A.6). La salida de pcm-power se guarda en un fichero al que se le da el nombre que se quiera. En este caso, el nombre consiste del número de tareas totales, el nombre de la aplicación y `"$filename"`. Esto aporta información sobre a qué nodo pertenece la información. En este caso no haría falta pero en pruebas con más de un nodo sí. La opción `-i=660` se utiliza para que el software se ejecute todo lo que dure la aplicación, con un margen tanto anterior como posterior para poder observar el aumento y descenso del consumo.

El final del archivo está relacionado con el procesamiento de los ficheros. Primero se espera a que el programa termine, haciendo un sleep de 15 segundos posterior para poder observar el descenso del consumo mencionado previamente. Una vez pasado este tiempo, se para la ejecución del archivo "llamadas.sh", nos volvemos a situar en el directorio principal, que es donde se almacena el fichero mediciones.txt y con este lo que se hace es cambiarlo de nombre, en este caso para mostrar el número de tareas, 02, y el número de nodos involucrados, en este caso 1 (N1).

```
#!/bin/bash
# Lista de nodos del clúster
nodos=("nodo51")
# Iterar sobre cada nodo y ejecutar el programa
for nodo in "${nodos[@]}"; do
    filename="salida$(ssh ${nodo} 'uname -a | grep -oE "nodo[0-9]+")"
    ssh_command="sudo /opt/pcm/build/bin/pcm-power 1 -i=660 -silent > 02Graph500\$filename"
    eval $ssh_command &
done
./llamadas.sh &
PID=$!
sleep 30
cd graph500
cd src
mpirun -n 2 --host nodo51:2 -x LD_LIBRARY_PATH --map-by core --report-bindings --mca btl tcp,self ../../src/graph500_reference_bfs 22 &

PID2=$!
# Esperar a que todas las instancias del programa finalicen
wait $PID2
#echo "Todas las instancias del programa se han completado."
sleep 15
kill -TERM $PID
cd ..
cd ..
mv mediciones.txt "02N1Graph500SNMP"
```

Código A.6: Script para recoger el consumo de la aplicación Graph500.

La llamada para que se ejecute el archivo "llamadas.sh", está relacionada con la recogida de datos de la PDU, teniendo el archivo los siguientes comandos (código A.7):

```
while true; do
    snmpwalk -v1 -c public 161.67.132.209
        1.3.6.1.4.1.534.6.6.7.3.4.1.4 | cut -d " " -f 4 >>
        mediciones.txt &
    sleep 0.5
done
```

Código A.7: Contenido del fichero llamadas.sh.

Con esto lo que se hace es guardar únicamente los valores de la métrica "inputWatts" registrada por la PDU. El comando "cut" sirve para quedarse únicamente con el dato numérico. La frecuencia de medición es de medio segundo.

La forma final que tiene el contenido del fichero mediciones.txt es como esta:

1778
1776
1800
1900
2000
1900
1800
1775

Antes de llegar a ese resultado, los datos que se obtienen ofrecen más información. Así, con el siguiente comando:

snmpwalk -v1 -c public 161.67.132.209 1.3.6.1.4.1.534.6.6.7.3.4.1.4

se obtiene una salida en la que se muestra el OID específico del resultado mostrado, así como el tipo de dato:

iso.3.6.1.4.1.534.6.6.7.3.4.1.4.0.1.1 = INTEGER: 1773

Siguiendo con el script, una vez terminada toda esta ejecución, queda un último paso que es procesar los datos almacenados con pcm-power, que tienen mucha información pero que aquí sólo interesan los datos de consumo expresados en vatios. Para esto se crea un script llamado "procesar.sh" con el contenido indicado en código (A.8).

De esta forma se añade el prefijo watts al archivo para saber que se ha procesado y quedan los datos numéricos, además con una "," separando los decimales, lo que será de gran ayuda para luego procesar todo estos datos. Se quedarán los datos en una columna de igual manera que con el fichero "mediciones.txt", solo que estos ahora presentan decimales separados de las unidades por una "," como se ha dicho anteriormente.

```
#!/bin/bash

# Verificar que se haya proporcionado un nombre de archivo como
# parámetro
if [ $# -eq 0 ]; then
    echo "Error: No se ha proporcionado el nombre del archivo como
    parámetro."
    exit 1
fi

# Asignar el nombre del archivo proporcionado como parámetro
archivo=$1
# Realizar los comandos sobre el archivo
cat "$archivo" | awk '/Consumed Joules/ {print $10}' > watts"
$archivo"
sed -i 's/\.\./,/g' watts"$archivo"
sed -i 's/;//g' watts"$archivo"
```

Código A.8: Contenido del fichero procesar.sh

Este procesamiento de los archivos se puede incluir en la ejecución general, sobretodo cuando se ejecutan las aplicaciones en un único nodo, pues sólo se tiene un archivo que procesar por ejecución.

Así mismo, se debe señalar que el uso de comandos pcm-power como el mostrado a continuación:

```
sudo /opt/pcm/build/bin/pcm-power 1 -i=20
```

genera una salida con mucha información relacionada con el consumo que es la que el script filtra. En concreto, el comando anterior genera la siguiente información:

S0; Consumed energy units: 235438; Consumed Joules: 14.37; Watts: 14.40; Thermal headroom below TjMax: 59

S1; Consumed energy units: 222970; Consumed Joules: 13.61; Watts: 13.65; Thermal headroom below TjMax: 58