Available online at www.prace-ri.eu

Partnership for Advanced Computing in Europe

Aiming to Optimise HPC Resource Utilisation based on Jobs' Energy Consumption Data

U. K. Seth[a], C. Andrieu[a], G. Hautreux[a]*, M. Cloirec[a], G. Gil[a], C. Mérouze[b]

[a]*Centre Informatique National de l'Enseignement Supérieur (CINES), France*
[b]*Big Data & Security Bull ATOS,* France

Abstract

Optimisation of energy consumption and HPC resource utilisation are high-priority issues for any supercomputing centre. Accelerators will drive the floating-point performances of exascale supercomputers; however preparing the scientific communities for these ever-changing architectures is a daunting task. Thus, the work of HPC support teams is crucial in achieving an overall optimised resource utilisation. We highlight a methodology to monitor and optimize resource utilisation at a national supercomputing centre, CINES [8]. We used a tool to collect the total energy consumed (kWh) by the simulation jobs on supercomputer 'Occigen' for several months. Based on benchmark runs and our data, we prepared a classification criterion to distinguish jobs based on their normalised energy consumption; and classified them from low to high-energy consumption classes. Our data analysis shows interesting trends that help us identify suboptimal jobs, along with other insights, to improve resource utilisation as a final goal on CINES's supercomputers.

Keywords— Energy consumption, HPC jobs, exascale, HPC resource optimisation

1 Introduction

With the arrival of "exaflopian" supercomputers [1], machines that are increasingly powerful and efficient but also consume considerably more energy, optimisation of energy consumption is a preoccupying subject that concerns every HPC center around the world. The place of these power hungry supercomputers, in the overall energy consumption pyramid, has become central and a priority, which requires dynamic optimisation approaches on the resource utilisation side. The rankings of such machines are no longer based solely on their floating-point performances (flops), but, since the first Green500 list was published in June 2013, they are also based on the gigaflops/watt ratio [2], which adds a more ecological vision. This criterion even comes into play during the TCO (Total Cost of Ownership) exercise by HPC centers while purchasing new supercomputers.

Various studies, discussing different aspects related to energy consumption on HPC systems, have been published. Articles show that it is possible to predict the energy consumption by jobs in the short term with a small

*Corresponding author email address: hautreux@cines.fr

margin of error to improve the power capping in HPC systems [3], [4]. Several studies have proven that correlations exist between the way of coding and the energy consumed by the code. For example, the CSX format for the execution of the conjugate gradient method is more efficient in terms of performance and consumption than the CSR format [5]. Studies have been carried out to dynamically cap the power draw by tackling execution time imbalances of HPC jobs by power and frequency management [6]. The efficiency of interconnection network between the nodes also impacts the power consumption. Authors in reference [7] discuss an energy efficient high performance hierarchical interconnection network for exascale supercomputers.

Different approaches to use the power budget efficiently are relevant for centres like CINES to keep running their current and future supercomputers. CINES is a French tier-1 supercomputing centre for higher education and research. Since 2015, more than 1500 users utilize its supercomputer per year, and the users consume millions of core-hours every year. The supercomputer 'Occigen' consists of 2106 nodes of Intel Haswell (12 cores) and 1260 nodes of Intel Broadwell processors (14 cores); and its peak theoretical floating-point performance is 3.5 pflops [8]. The research community, using the computational resources at CINES, includes several scientific domains ranging from fluid mechanics to plasma physics and mathematics. The average power drawn by the supercomputing machines at CINES, in recent years, is shown in Figure 1, for which the electricity bills reach millions of euros.

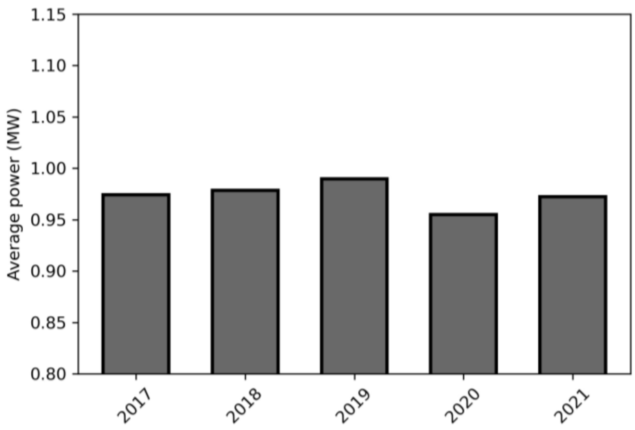

Figure 1: Average yearly power drawn by the supercomputers of CINES in recent years

The HPC team at CINES encounters different instances of suboptimal resource usage that impact the overall efficiency of the system. We have noticed that most of the jobs use very few nodes on 'Occigen'. In 2021, more than half of the total jobs (53.02%) ran on a single node, 75.85% of the jobs used less than 5 nodes and almost 85% of the jobs were executed on less than 10 nodes. Moreover, sometimes some serialised programs are run on full nodes, i.e., an entire node is allocated (e.g. 28 cores for a 2 socket Broadwell node) while only one core is used for the simulation. Such situations arise when the codes or the slurm scripts are not tuned according to the specific cases and the resources demanded. All this results in a suboptimal energy efficiency of the system.

Dynamically monitoring the system resource utilisation is an inevitable part of the HPC support teams' work. A number of job profiling tools (XALT, selfie, etc.) and parameters (frequency, energy, execution time in data communication libraries etc.) are available which can be used to monitor jobs resource utilisation on supercomputers. In this study, we present one such approach to understand and monitor the resource usage by using the total energy consumed (kWh) by the jobs on computing nodes. The main idea is to classify jobs into several groups based on their electrical energy consumption that goes from very low to high depending on the efficiency of the simulation.

In 2022, the supercomputer 'Occigen' will be replaced by 'Adastra', a supercomputer offering a peak performance of about 75 PFlops [9]. With this study, our aim is to identify jobs, which are using the system resources in suboptimal range of energy consumption, and then contact the corresponding users to assist them optimize their resource utilisation. With the experience gained on 'Occigen', the methodology we have developed will be used for the upcoming supercomputer 'Adastra'. This would allow us to optimize the use of computing resources on the supercomputers of CINES, all in compliance with the General Data Protection Rules (GDPR), based on data privacy laws across Europe [10]. In the following sections, we present the results of our study that are rather encouraging and open perspectives for future machine learning studies.

2 Data Management Methodology

Enhancing the utilisation of system resources is one of the most important priorities for an HPC centre. Supercomputing centres like CINES generate a variety of data in several hundred gigabytes every day. Often, some initial data analysis leads to some characteristics of system utilisation that are not known by the system administrators beforehand. With the advancements reached in machine learning technologies, it is inevitable to use these system-generated data to enhance the overall utilisation of the whole system. For a supercomputer, different types of system data come from varied sources within and outside of the system which can be used to develop relevant machine learning models to solve existing problems or for providing insights into the functioning of the system.

2.1 Project Data at CINES

With the availability of several open source tools, exploitation of data for seeking performance enhancement is now inescapable in any setting. However, there is always a whole pipeline from data generation to final exploitation of data which is not trivial to manage. Most of the time, this process of collecting and pre-processing data is longer than the individual data analysis studies themselves. At CINES, we have a dedicated project to manage our data pipeline with the continuous-integration (CI) mechanism available with gitlab. In general, our data pipeline consists of various data sources, a data lake, a data warehouse and need based data marts (Figure 2). The sources of data consist of Slurm's jobs' data, user data, data on the use of storage spaces, energy consumption data etc.

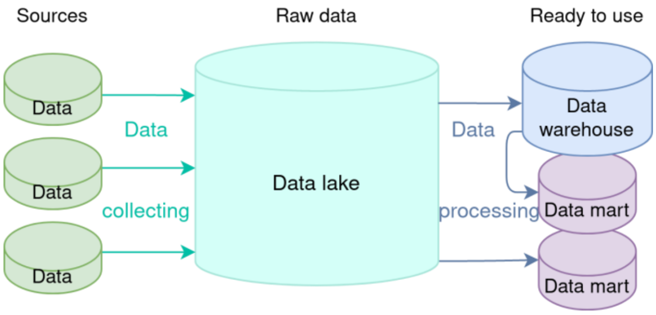

Figure 2: Sketch of a general data pipeline

Slurm's jobs' database is the most important data source which provides a global view of resource utilisation by all the jobs running on the supercomputer. Every simulation job running on the system has one entry in the jobs' data lake. For every job, Slurm provides the number of nodes, cores, threads, elapsed time, memory usage, modules used by the job and other relevant information. To complete the jobs' data warehouse, we aggregate additional relevant job data from Slurm's jobs' data, namely, the user account related information and the energy consumption data by every job.

2.2 Collection of Energy Data

We collect the jobs' energy consumption data from Atos's proprietary tool BEO (Bull Energy Optimizer) [11]. BEO was installed on Occigen's Broadwell partition in 2020. This tool provides power, energy and temperature as three main metrics to observe the energy consumption on cluster nodes and interconnect switches, along with other cluster components. For our data warehouse, we collect the power (Watt) and time (sec) data of a job running on the Broadwell nodes and convert it to total energy (kJ) consumed by the individual job. The total energy consumed by individual jobs is used for the study presented in the following section of this paper. The use of BEO is non-intrusive with zero overhead for the simulation jobs, as BEO runs on a different server and collects the data directly from the blade monitoring controllers located on compute nodes. In addition, collection and management of this energy database provided by BEO is also simplified with its interface in comparison withour other databases.

3 Energy Data Analysis

As explained in the previous section, we started collecting the energy consumption data for jobs in our data warehouse from June 2021 and used for this study a dataset till February 2022, to which we will refer as as 'our dataset' for the rest of this paper. Mainly, we have one column each for energy consumed on nodes and energy consumed on switches by individual jobs; and we add the entries of these two columns to get the total energy consumed by individual jobs. During the nine months, we could collect energy consumption data for a total of 149612 jobs running on the Broadwell partition of 'Occigen', which is roughly 20% of all the jobs running on the full machine (Broadwell and Haswell partitions) during this time.

The users of 'Occigen' could ask for up to 700 nodes for running their individual jobs. The total energy consumed by a job corresponds directly to the number of nodes it was running on and, elapsed time of the job. Normally, the time limit for individual runs is fixed at 24h unless the users request for longer jobs for their particular cases. This limit of 24h pushes users to develop a restart feature in their codes, so that the loss of simulation data can be minimised in case of a failure and the resource allocation can be balanced among all users. The mean of total energy consumed in our dataset was 6.7 kWh. Often plotting all the data, consisting of long ranges, together does not really give us much insight into the data, because the anomaly data points hide the trends at finer levels. Therefore, we have to select some smaller ranges of data to check if some patterns emerge into such smaller zones of the dataset.

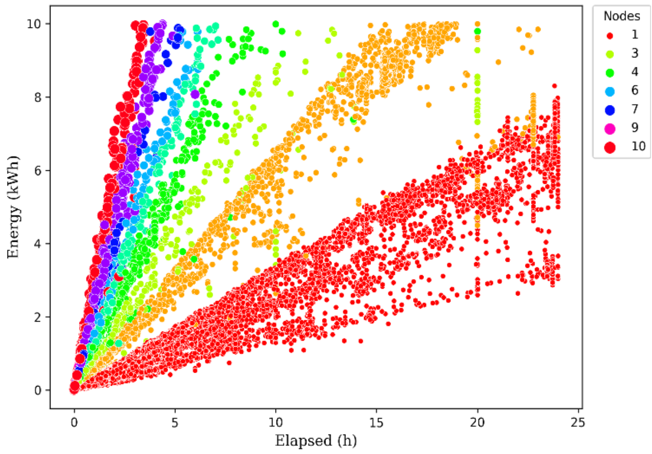

Figure 3: Total energy consumed by jobs in elapsed time. Data points coloured with number of nodes the corresponding jobs were running

While doing such experimental data analysis with our energy consumption data, we plotted the jobs within the elapsed time of 24h and total energy value until 10kWh. Figure 3 shows one such plot, where the individual data points are coloured based on the number of nodes the corresponding job was running. We observe some linear trends emerging in this plot. It appears that for a particular value of number of nodes the plot of total energy (kWh) has a clear linear relation with the elapsed time (h). We observe that with the increasing value of number of nodes the slope of the linear trend is increasing which supports the logical idea that the jobs running on a higher number of nodes will consume a higher amount of total energy in less hours.

Considering the linear trend observed in Figure 3, we also plotted per node energy consumed by jobs in their corresponding elapsed time. Figure 4 shows this per node energy consumption by jobs, and the data points for individual jobs are sized and coloured based on the number of nodes the simulation was carried out. Interestingly, this plot shows an overall linear behaviour of jobs' energy consumption with elapsed time. In this plot, we have only taken the jobs running on the standard job queue of 24h. We observe that most of the jobs (over 95 % of total jobs) are represented by small blue points, which means these were the jobs running on a very small number of nodes as shown with the scale of number of nodes. Most of these data points are concentrated along the global linear trend line which is clearly visible in this plot.

There are also a number of data points which are slightly shifted on both sides from the main linear trend line. Mainly, the larger data points are shifted on the higher side of the energy axis (y), most of which appear to be bigger than 100 node jobs. The shift on the higher energy side of the trend line is more pronounced for the jobs running for more than 20h of elapsed time on a higher number of nodes. Although, there are certain large data points visible within the 5h elapsed time range also, which simply shows some large jobs that finished quickly and thus consumed less per node energy. The highest value of per node energy consumption is slightly above 12 kWh in the data selected for this plot.

There seems to be a separate zone of data points on the lower energy side of the trend line. No large sized data points are visible in this zone, so these are the jobs running on a very small number of nodes (N<10). A line can be drawn on the boundary of this zone, at the lower end of the per node energy value, out of which there are practically no data points, which suggests that there is surely a minimum value of per node energy consumption which is the least value consumed by the jobs in any case.

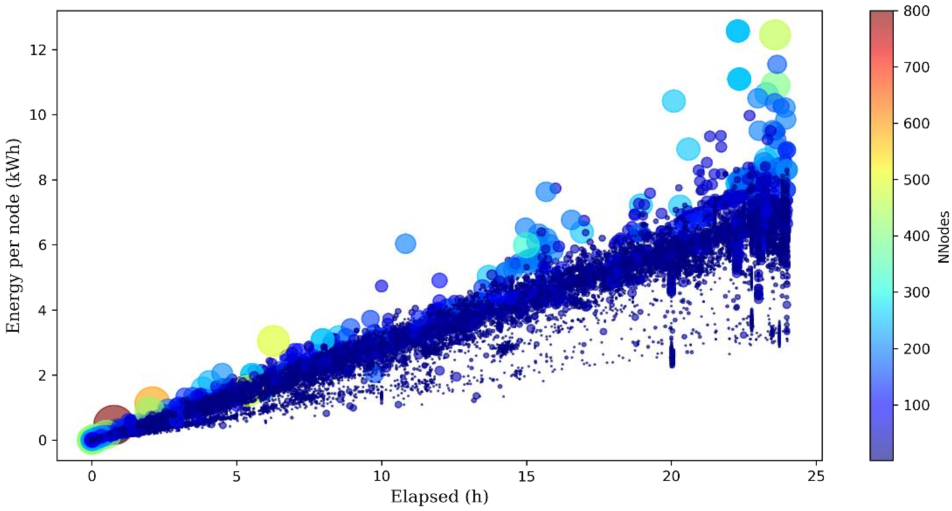

Figure 4: Jobs' per node energy consumption (kWh), data points sized and coloured with number of nodes the job was run

To extend our data analysis, we computed a new variable in our dataset which is the energy consumed per node per hour by individual jobs. We call it the *normalized energy* for jobs which is represented in kW per node. A distribution plot of the jobs having this normalised energy on the x axis is shown in Figure 5. The distribution plot shows that there are two distinguished peaks in the dataset which makes it a bimodal distribution. The two peaks in this distribution appear at 0.14 kW and 0.27 kW values of the normalised energy.

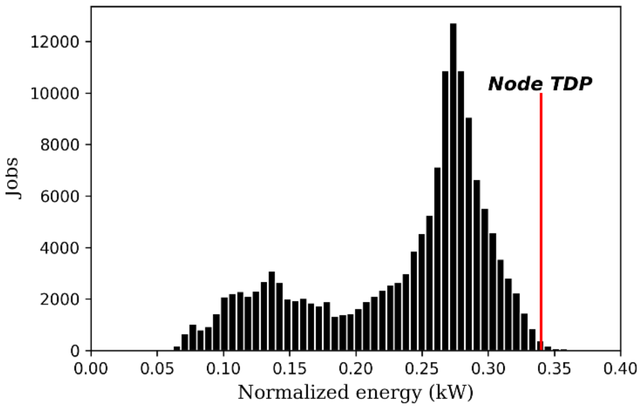

Figure 5: Distribution plot of jobs with normalised energy

We can easily state that there are at least two groups of jobs based on this normalised energy variable, as the jobs' frequencies are significant around both peaks. In Figure 5, we also put a marking for an approximate value of Thermal Design Power (TDP) for the Broadwell nodes of 'Occigen' which is about 0.34 kW. The TDP for a compute node consists of the TDP of all components (processor sockets, memory slots, disks, motherboard etc.) of the node. For example, the combined TDP of two Broadwell processors is 0.27 kW. The thermal design power represents the maximum heat that the cooling mechanism can dissipate from the system.

To better understand these two groups of jobs, we carried out test runs with some benchmark codes and noted their energy consumption. We ran three times a test case of HPL and Stream benchmark on one node for one hour. HPL is a High-Performance Linpack benchmark implementation. It solves a (random) dense linear system in double precision (64 bits) arithmetic on distributed-memory computers [12]. The Stream benchmark is a simple, synthetic benchmark program that measures sustainable main memory bandwidth in Mb/s and the corresponding computation rate for simple vector kernels [13].

We also ran three times an idle job on one node for one hour. For the idle job we simply used the 'sleep' command for one hour duration within the Slurm script. The average values of normalised energy obtained for idle job, Stream and HPL cases were 0.135 kWh, 0.264 kWh and 0.386 kWh respectively. From these benchmark case values, we can notice that the jobs around the first peak value in Figure 5 are consuming energy near the idle job consumption value. And the jobs around the second peak value in Figure 5 are the jobs lying between the Stream and HPL energy consumption values, but more inclined towards the energy consumption of Stream benchmark.

Based on all these observations and the normalised energy range of our jobs, we set a criteria to classify the jobs based on their energy consumption. For the boundaries of the classification criteria, we decided to choose less than 0.15 kW values as the lower bound and higher than 0.35 kW values to be the upper bound. The jobs having less than 0.15 kW as the normalised energy values, are said to be in ‘Idlejob’ class. This class mainly contains the jobs consuming around the energy consumption level of our ‘idle job’ test case.

We decided to keep the range between 0.25 kW to 0.35 kW as the optimal range of normalised energy consumption by the jobs. As mentioned earlier, the observed normalised energy for Stream benchmark was 0.264 kW. Stream benchmark provides a case for sustainable memory access from main memory, and it computes some simple vector kernels as well, making it a simple generalised test case that does both memory access and computations. Bandwidth values below those obtained with Stream benchmark shows a suboptimal use of available memory bandwidth with the corresponding code. Similarly, HPL benchmark goes on the higher side of floating-point performance.

Thus, the normalised energy range between the Stream and HPL values, but slightly tilted towards Stream, is a good assumption for our ‘Optimal’ energy consumption class. The detailed classification criteria is mentioned in Figure 6.

$$0 < \text{Idlejob} \leqslant 0.15$$
$$0.15 < \text{Low} \leqslant 0.2$$
$$0.2 < \text{Moderate} < 0.25$$
$$0.25 \leqslant \text{Optimal} \leqslant 0.35$$
$$0.35 < \text{High}$$

Figure 6: Classification criteria based on normalised energy (kW) consumption values, as used in our study

We classified the jobs based on the explained criteria (Figure 6). We observed that there were 26274 jobs in the Idlejob class which was more than 17% of the total jobs in our dataset (Figure 7). Therefore, more than 17% jobs in our dataset were consuming energy around the level of an idle job, which means they were using a very small fraction of resources (cores, memory, etc.) reserved for them by the user. There were about 58% of jobs which fell into the ‘Optimal’ class, and about 15% jobs fell into the ‘Moderate’ energy consumption class. Interestingly, there was a very small number of jobs, about 0.3% of total jobs, in the ‘High’ energy class also, which were consuming energy above the level of thermal design power (TDP) of the node. Figure 7 summarises the number of jobs in each class based on our classification criteria.

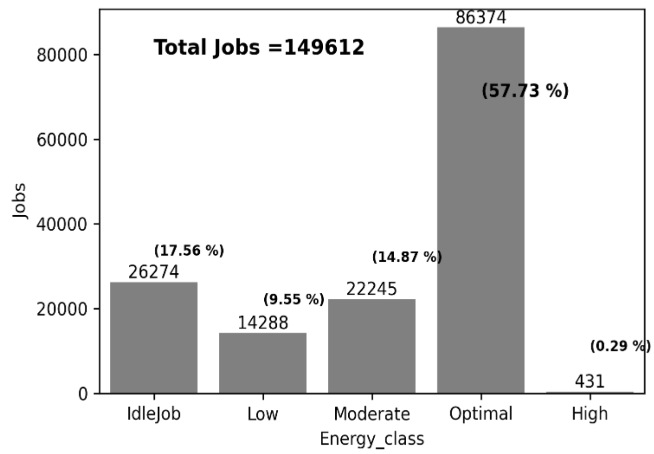

Figure 7: Jobs classified into the energy consumption criteria

We re-plotted the per node energy consumption, and coloured the data points with normalised energy values as shown in Figure 8. With the classification criteria and the scale of normalised energy used in Figure 8, we clearly see that most of the jobs which are concentrated towards the main linear trend line belong to the ‘Optimal’ energy class. The small number of jobs which were running on a large number of nodes (Figure 4), are the ones which consume energy in ‘High’ energy class range. The data point zone of lower energy values, below the linear trend line, corresponds to jobs in ‘Low’ and ‘Idlejob’ classes of our classification criteria. As visible in Figure 7 and

Figure 8, this zone consists of around 27% of total jobs, of our dataset, which combined belong to low and idle levels of energy consumption.

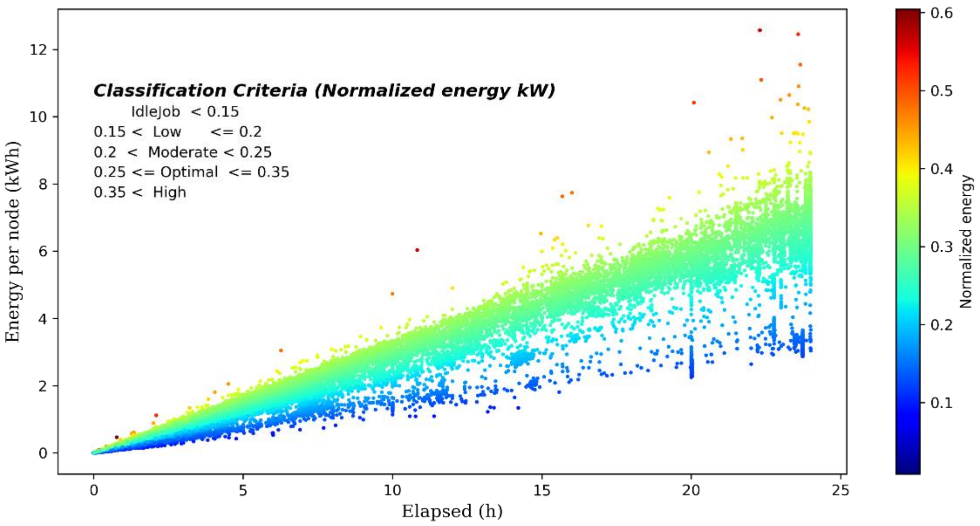

Figure 8: Job's per node energy consumption (kWh), data points colored with normalised energy (kW)

Number of jobs alone does not give us the real idea on actual energy consumption by jobs. We noted that all the jobs in our dataset used about 93.27 million core-hours in total on Occigen during the nine-month period. A core-hour represents running a single core of a processor for one hour. Therefore, if a job ran for one hour on a compute node which had 28 physical cores (e.g. Intel Broadwell), then the job consumed a total of 28 core-hours because all 28 cores were used for one hour. In Figure 9, we show the distribution of total core-hours, for jobs in our dataset, into the energy classes. We observe that about a total of 4.84 million core-hours were consumed by the jobs in Low and Idlejob classes combined, which amounts to about 5% of the total core-hours. This tells us that for these 5% core-hours some optimisation is required to push these jobs towards the optimal use of resources.

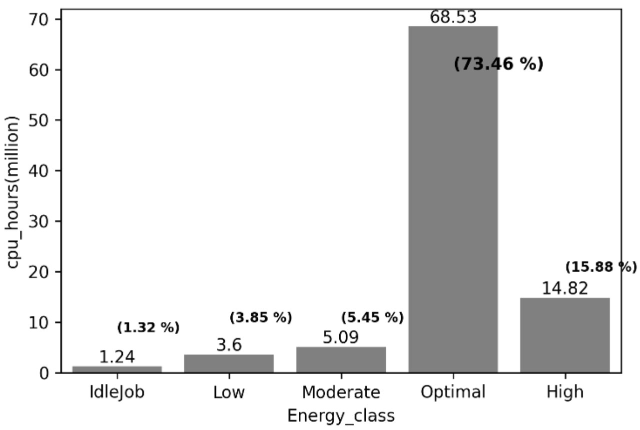

Figure 9: Distribution of core-hours in various energy classes

However, about 73% of core-hours were consumed by the jobs in the 'Optimal' energy class, which represents an optimal use of resources. Interestingly, about 15% of the core-hours were consumed by jobs in 'High' energy class as well. We recall from Figure 7 that the jobs in High-energy class were only ~0.3% of the total jobs in our dataset. Extending the analysis further leads us to discover more insights on the actual use of resources by the users of 'Occigen'.

Figure 10 shows the core-hours used by the jobs in Idlejob class along with the number of nodes they were running on. We had mentioned earlier that about 17% of the jobs were in Idlejob class (Figure 7), and here we see that about 40% of the core-hours of Idlejob class are spent on single node jobs. The set of jobs running on 10 and 20 nodes used around 25% of core-hours each in Idlejob class. To explore further the user profiles of Idlejob class, we plotted the core-hours consumed against the scientific domain list, on x axis, within which the corresponding users applied for resource allocation. For the computational resource allocation, there are 11 thematic committees (CT) which review the user applications in their respective scientific domains (Table 1).

Table 1 Thematic committees and their respective scientific domains

Committee ID	**Scientific domain**
CT1	Environment
CT2A	Non-reactive flows
CT2B	Reactive or/and Multiphase flows
CT3	Biology and Health
CT4	Astrophysics and Geophysics
CT5	Theoretical and Plasma Physics
CT6	Computer science, Algorithms and Mathematics
CT7	Molecular modelling applied to Biology
CT8	Quantum Chemistry and molecular modelling
CT9	Physics, Chemistry and properties of materials
CT10	Artificial Intelligence and cross-disciplinary applications of computing

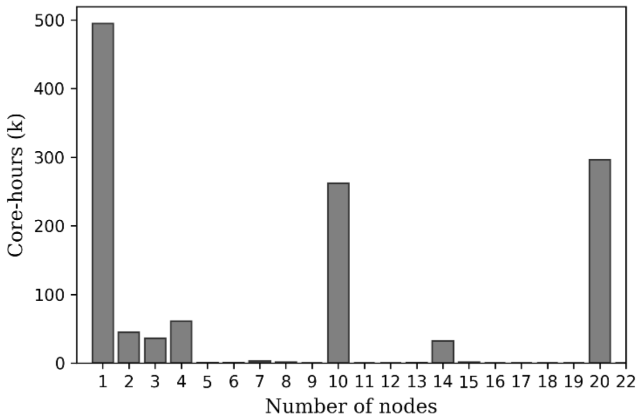

Figure 10: Total core-hours used in ‘Idlejob’ class by jobs running on various number of nodes

Figure 11 shows that the jobs which belong to thematic committees CT7 and CT9, consumed about 30% of the core-hours in each of these two thematic committee groups within the Idlejob class. Committee CT7 represents the scientific domain of “Molecular modelling applied to biology”; and CT9 represents the domain “Physics, chemistry and properties of materials”. Similarly, it was observed that the users in CT5 (Theoretical and plasma Physics) consumed around 54% of total core-hours in High-energy class, and, CT2A (Non-reactive flows) users consumed around 27% of total core-hours in High-energy class. It is interesting to note that in both Idlejob and High-energy classes the energy consumption is concentrated within certain scientific domains.

Experience of user support team at CINES suggests that, sometimes, most of the users of a certain CT use the same software package and the user support team often intervenes to help these users optimize their use of such packages for optimal resource consumption. From our methodology, it will be possible to identify and pro-actively contact users, at an early stage, whose jobs appear in Idlejob or Low energy classes, and, assist them in optimizing resource utilisation from their side.

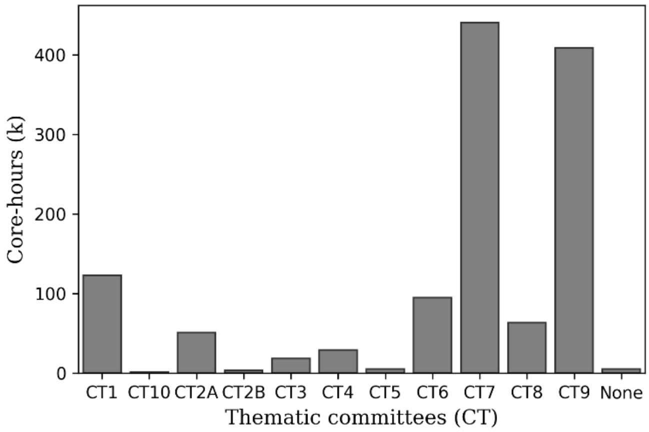

Figure 11: Core-hours consumed in 'Idlejob' class by groups in different Thematic Committees

In Figure 12, we present the percentage of total core-hours consumed in various energy classes by users of each thematic committee (CT). Stacked bar plots are used to separate the percentage consumption among different energy classes. Total core-hours consumed by each CT are also represented with the black '+' sign, the corresponding value of which is read from the total core-hours axis on the right side of the plot. The predominant green colored bars show that most of the core-hours are utilized in 'Optimal' energy class within each CT. However, we also see that for some of the CTs (CT3, CT6, CT7, CT9) the combined value of red (IdleJob) and orange (Low) colored bars are reaching about 15% or more. These red and orange colored bars represent the suboptimal resource utilisation based on the energy consumed by the jobs, which should be targeted for optimisation. Overall, this figure (Figure 12) shows the global classification of total core-hours in the energy classes defined based on our methodology. This is the global energy consumption behavior of all users within all thematic committees.

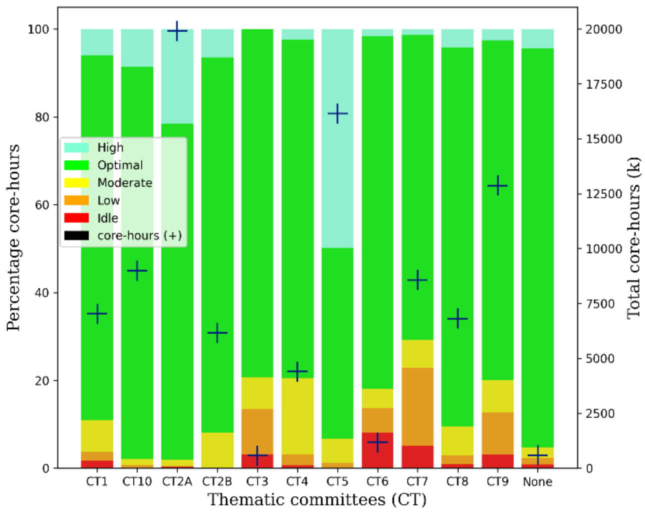

Figure 12: Percentage and total core-hours consumed in various energy classes by users in each Thematic Committees

All the data analysis and the observations made above help us to prepare a kind of energy consumption profile of our users and assist them accordingly. For example, we can easily find out the top energy consumers in classes 'Idlejob' and 'Low'; and contact them to understand what kind of simulations they are running. And consequently, helping them improve resource utilisation by optimizing their simulation cases. We can easily find out how many core-hours an individual user is consuming in all energy classes, because we noticed that most of the users had run jobs which were spread over all the classes we mentioned above. We can also prepare a profile table, which summarises the overall behaviour of a user on the machine to help us make necessary decisions. An example of a user energy-consumption profile table, based on our classification criteria, is shown in Figure 13, which can be improved further by adding more variables as necessary.

Energy_class	energy_total(kWh)	Mean_energy_norm	Mean_Nnodes	cpu_hours(k)
High	19.571940	0.369840	7.000000	1.481760
IdleJob	1437.654993	0.110857	9.554622	375.409961
Low	919.023820	0.171455	6.807018	140.601191
Moderate	1910.126517	0.236855	8.741573	220.138193
Optimal	12793.963850	0.277023	7.502591	1234.495173

Figure 13: A sample 'user energy profile'

The most important observation with this exercise was to note that a significant amount of core-hours on our machine was used for jobs which seemed to stay idle, based on the number of cores they had allocated and the actual energy they were consuming on those allocated cores. And, this informs us that the computing resources were not optimally used by a certain number of jobs (users). An example of such cases can be a job which allocated one full node (28 cores) for the simulation but it ran a serial (1 core) simulation or it used very less number of cores than it allocated.

With this whole exercise, we have another methodology to monitor and optimize the overall resource utilisation on CINES's upcoming supercomputer [9]. Users of supercomputers often run similar kinds of simulations, and if we can find the users whose jobs are consuming well below the optimal level of energy, at an early stage, we can help them optimize their upcoming simulations. We plan to put in place the tools we have prepared for this analysis, in a continuous integration manner, to recover the energy relevant data, and consequently prepare the user profiles on a regular (weekly) basis to dynamically monitor their resource utilisation. We are also working towards developing some machine learning models to help us optimize the resource utilisation based on the overall jobs' data, which we plan to present in our future work.

4 Conclusion

We started this study with the aim of understanding the energy consumption behavior of HPC users on the supercomputer 'Occigen', hosted at CINES, France and consequently, to optimize the resource utilisation as the final goal. We have put in place a mechanism to collect and analyze the HPC jobs' energy consumption data on 'Occigen' with tool BEO. The use of BEO is non-intrusive with zero overhead for the simulation jobs, as BEO runs on a different server and collects the data directly from the blade monitoring controllers located on compute nodes. A clear linear trend for per node energy consumption of jobs was observed when plotted against the elapsed time. Based on benchmark runs (Stream, HPL and idlejob) and on the observed jobs' energy consumption, we prepared a classification criteria and classified jobs in our dataset according to their energy consumption values (kWh). Our classification criteria consists of five levels of energy consumption by jobs: IdleJob, Low, Moderate, Optimal and High, which are discussed in detail in this paper.

We observed that a significant number of total core-hours were utilised by the jobs consuming the electrical energy far below the optimal level of energy consumption. This suggests as if these jobs were running idle and doing practically nothing considering the amount of resources allocated for them. It was observed that these 'Idlejob' class jobs were running on a small number of nodes (e.g. 1, 10 or 20); and they were concentrated into certain scientific domains alone. We noted that about 5% of total core-hours were consumed by jobs in 'Idlejob' and 'Low' energy classes combined. With the methodology presented in this paper, we can identify the corresponding users and assist them in optimizing their resource utilisation, and by pushing these users towards optimised methods, we could probably increase the overall throughput of the machine by about 5%. We plan to continue and advance our methodology on the upcoming supercomputer of CINES, with a clear goal of optimizing the HPC resource utilisation.

References

[1] Top500, November 2021. Accessed on: March 30, 2022. [online]. Available: https://www.top500.org/lists/top500/2021/11/

[2] Green500, November 2021. Accessed on: March 30, 2022. [online]. Available: https://www.top500.org/lists/green500/2021/11/

[3] S. Suzuki, M. Hiraoka, T. Shiraishi, E. Kreshpa, T. Yamamoto, H. Fukuda, S. Matsui, M. Fujisaki, A. Uno, "Power prediction for sustainable HPC", on J-STAGE, vol. 29, pp. 283-294, 2021, doi: https://doi.org/10.2197/ipsjjip.29.283

[4] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, L. Benini, "Predictive modeling for job power consumption in HPC systems", in Lecture notes in computer science, vol. 9697, June 2016, doi: http://dx.doi.org/10.1007/978-3-319-41321-1_10

[5] K. Nikas, D. Siakavaras, V. Karakasis, J. C. Meyer, L Natvig, "An energy-centric study of conjugate gradient method", PRACE Whitepaper. Accessed on: March 30, 2022. [online]. Available: https://prace-ri.eu/wp-content/uploads/wp89.pdf

[6] B. Wang, C. Terboven, M. Müller, "Dynamic runtime and energy optimization for power-capped HPC applications", in IOS Press, 2020, doi: 10.3233/APC200070

[7] F. A. Faisal, M. M. Hafizur Rahman, Y. Inoguchi, "HFBN: an energy efficient high performance hierarchical interconnection network for exascale supercomputer", in IEEE Access, vol. 10, pp. 3088-3104, 2022, doi: 10.1109/ACCESS.2021.3138828.

[8] "The Supercomputer Occigen". Accessed on: March 30, 2022. [online]. Available: https://www.cines.fr/en/supercomputing-2/hardwares/the-supercomputer-occigen/configuration/

[9] "Adastra : la France se dote d'un nouveau supercalculateur pour préparer l'Exascale", Nov. 14, 2021. Accessed on: March 30, 2022. [online]. Available: https://www.genci.fr/fr/node/1150

[10] GDPR, "General Data Protection Regulation", May 2018. Accessed on: March 30, 2022. [online] Available: https://gdpr-info.eu/

[11] X. Vigourous, L. Sauge, "Towards energy consumption application profiling with Bull Energy software", SC17, Nov. 14, 2017. [online]. Available: https://prace-ri.eu/wp-content/uploads/PRACE-at-SC17-Ludovic-Sauge.pdf

[12] A. Petitet, R. C. Whaley, J. Dongarra and A. Cleary, "HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers", *Innovative Computing Laboratory*. Accessed on: March 30, 2022. [online]. Available: http://www.netlib.org/benchmark/hpl/.

[13] McCalpin, John D., 1995: "Memory Bandwidth and Machine Balance in Current High Performance Computers", IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter, December 1995.

Acknowledgements

This work was financially supported by the Members of PRACE aisbl and by the PRACE project funded in part by the EU's Horizon 2020 Research and Innovation programme (2014-2020) under grant agreement 823767.