Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Developing a digital twin to measure and optimise HPC efficiency

ARTICLE INFO

Keywords:
HPC
Data-driven optimisation
Digital twin
Energy-aware scheduling
Data processing

ABSTRACT

A Digital Twin for a high-performance computer requires the integration of many sensors and data sources for monitoring operation and tuning of the system.

This paper introduces the network concept for the Digital Twin of PTB's HPC cluster with agents, software components to collect, process and send measurement data to the database of the Digital Twin, and how sensors are separated into networks for security of the data centre.

The Digital Twin is used to tune the system for reduced CO_2 emissions. In a first step, the focus lies on two control parameters: Processor frequencies, scaled specifically to maximise energy savings, and a modified scheduling mechanism, with deliberate delays helping to exploit phases of low CO_2 emissions. In the simulation, frequency scaling showed saving of 3–4 percent and the modified scheduling mechanism, depending on the workload, of 1–9 percent. Combining both methods allowed for savings of up to 11 percent.

1. Introduction

Today's society heavily relies on digital services and infrastructure. Simultaneously, the energy demand of data centres in Germany that provide these services doubled over the last decade whilst energy costs also increased. This creates challenges both from an economic standpoint as well as from an operation standpoint regarding energy consumption and cooling. Normal data centres contain servers that host individual tasks such as e-mail, web services and databases. This paper focuses on high-performance computing (HPC). HPC is intended for problems that exceed the capacities of a single server by combining multiple servers into a single cluster. This cluster or parts of it can then solve that problem cooperatively. The advent of artificial intelligence models has further increased the demand for compute resources. Training these models is a computationally and memory intensive task.

HPC is used in many industry applications in the design and testing process. Cutting-edge research often relies on numeric models solved by HPC systems. State-of-the-art AI technologies are not possible without HPC. HPC systems require large amounts of energy and cooling. Providing compute capabilities in a responsible manner is a challenging task and an important contribution to the Sustainable Development Goals [1].

The optimisation dimensions of an HPC cluster are many-fold [2]. Possible optimisations are the energy efficiency of the HPC system, the associated CO_2 emissions and the total cost of the energy used by the HPC system. While optimising for these goals, one major concern is raw compute performance delivered by the system. This performance is typically measured in floating-point operations per second (GFLOPS). An energy efficient system offers more GFLOPS per Watt. Both GFLOPS and GFLOPS/Watt are determined by the hardware in use and can barely be optimised through software settings. Another possible metric is the Energy-To-Solution which tracks the amount of energy required to solve a given problem. While GLOFPS/Watt tracks raw compute power of the hardware, Energy-to-Solution can differ between two algorithms for the same problem. From an operation standpoint, energy prices and energy generation and thus CO_2 emissions can be metrics to optimise for directly.

Digital Twins have become a valuable tool in the Metrology domain. They help understand complex systems such as HPC clusters by integrating measurement and sensor data of the system, models of the real-life data centre and cluster and track metrics and optimisation goals. A data centre is fitted with many sensors and measuring devices that track the conditions inside it, such as temperature sensors in the servers, the server room as well as outside the building, electricity meters and heat flow meters in the cooling system.

The virtual representation of the real-life object in the Digital Twin can be used to test the influence of system parameters without altering the production system, thus avoiding negative effects on the system and its users. Additionally, testing the parameters in a simulation is faster because no actual compute jobs need to run on the system. It is also cheaper compared to testing directly on the production system because the simulation uses far less energy than the real-world HPC system.

This paper makes two contributions. First, the data collection and transformation from different data sources, sensors and meters into the database format using agents is described. Additionally, security considerations concerning network separation are discussed. Second, the models incorporated in the Digital Twin are used to get an estimation on CO_2 saving potential using two control parameters: Dynamic Voltage and Frequency Scaling (DVFS) and a different scheduling policy.

2. Background

Different definitions for the concept of a Digital Twin exist in literature. This paper uses the definition by H. van der Valk, H. Haβe, F.

https://doi.org/10.1016/j.measen.2024.101481

Please cite this article as: Alexander Kammeyer et al., *Measurement: Sensors,* https://doi.org/10.1016/j.measen.2024.101481

Möller and B. Otto [3]. By that definition a Digital Twin is a virtual representation of a real-world object, in this case an HPC cluster. The Digital Twin integrates several data inputs with the aim of data handling, data storing and data processing. Furthermore, it provides a bi-directional data linkage between the virtual world and the physical object. This data linkage needs to be synchronised regularly to ensure consistency with the real world, even though a permanent, immediate synchronisation is not required. The Digital Twin must provide interoperability with other systems. In recent days, Digital Twins have gained such an importance in industry that they were standardised by ISO [4]. Chapter 3 shows data linkage, data handling and data storage in the Digital Twin as well as InfluxDB that provides and interface to the data layer. The focus of Chapter 4 is on the virtual representation and the implemented models that simulate the system behaviour and how it can be used to explore optimisation potentials.

An HPC cluster is a system consisting of many, often heterogeneous compute nodes, that share an interconnect and storage space. They are connected with the purpose of solving complex tasks exceeding the capabilities of individual nodes. Batch schedulers are used to schedule tasks submitted by users. They assign the requested resources in the form of nodes to the jobs. The Digital Twin aims to replicate a real-world HPC system through simulating this scheduling process and the system condition. They combine data from the real-world object with the data from the simulations for a more realistic system simulation.

Digital Twins, as virtual representations, can help create an understanding of HPC systems and the influence of their parameters without altering the production system, thus avoiding negative effects on the system and its users. Additionally, testing the parameters in a simulation is faster because no actual compute jobs need to run on the real system. Simulating the scheduling process uses less energy than running actual jobs on the system to test parameters and thus is cheaper.

The HPC system with its components has many parameters that influence the overall system performance. The scheduler, as the core component, can employ different scheduling strategies to improve the system utilisation. As part of these strategies, it can start jobs at different times, e.g. when the energy price is cheaper. The compute nodes are the biggest energy consumers of an HPC system due to their number and energy consumption. Dynamic Voltage and Frequency Scaling (DVFS) allows to dynamically change the processors frequency and thus reduce the energy consumption of the system. Nodes that are idle for longer periods of time can also be turned off completely and restarted when needed. The Digital Twin of the HPC cluster can be used to test these methods and their effects.

Scheduling simulations are a common tool when it comes to testing algorithms or parameters for HPC systems. A Digital Twin improves on a simulation by incorporating data from the real-world object, thus allowing actual statements about the system compared to purely synthetic simulations. In literature, different simulations have been created, e.g. for the Slurm scheduler [5–7] or based on a Digital Twin [8]. Scheduling based on energy price and job traces has been done by J. M. Kunkel, H. Shoukourian, M. R. Heidari, T. Wilde [9] and X. Yang, Z. Zhou, S. Wallace, Z. Lan, W. Tang (+ another 2 authors) [10].

3. Sensors in the data centre

The Physikalisch-Technische Bundesanstalt (PTB) operates a data centre with an HPC system for research purposes. It is the central provider of compute resources at PTB. The following chapters describe the hardware and network configuration at PTB; however, the presented concepts are applicable to other HPC centres as well.

A data centre is equipped with a multitude of different sensors that monitor the condition of the data centre. Electricity meters measure the energy usage of the data centre, and the produced heat is removed through a cooling system that has its own set of heat flow meters. The room temperature in the server room is monitored by thermometers. The HPC cluster itself also provides information about its status like usage, current jobs, job queue information and more.

To provide a holistic view on the data centre, the Digital Twin needs to integrate all sensor data. The data from the sensors is mostly time-series measurement data, thus the Digital Twin uses an InfluxDB for storage. InfluxDB stores data as measurement points in buckets. Each point can have fields and tags that contain and describe the data, e.g. through numeric measurement data, units or sensor information, and must have a timestamp that defines when the measurement was taken. Fields and tags are key-value pairs. Fields are used for frequently changing numeric values. Tags only store string data and are intended for metadata. Since the Digital Twin collects all sensor data, it also provides a unified view on the data for the user in a single interface.

The data centre has electric energy meters that are connected to a gateway with REST/JSON interface. The heat flow meters in the cooling system use M-Bus and are connected to a M-Bus gateway that uses XML over http. They are part of the building infrastructure network, a special network without internet access where IoT devices like these gateways are located. An agent software was created that collects the data from these meters, transforms it to an InfluxDB compatible format and sends it to the InfluxDB outside of the infrastructure network. The agent software uses a modular, 3-stage design. The first stage is responsible for the different data sources and their respective protocols. The second stage transforms the raw data into a format understood by InfluxDB and can filter data. The third stage handles the communication with InfluxDB. This schema is shown in Fig. 1. Other common protocols, such as OPC-UA and Modbus/TCP, are currently not offered by sensors in the data centre but libraries for these protocols exist and they can be used to implement the protocols in the agent.

The HPC cluster itself also has a separated network where all the compute nodes and services are located. The cluster provides a REST/JSON interface through the ClusterCockpit software [11]. This includes information about the nodes, the running compute jobs and general system state information. Another agent in this network collects the data in similar fashion to the building infrastructure network.

The Digital Twin itself with the database and user interface resides in the client network of PTB, where the users of the HPC system reside with their client machines. The temperature sensors are accessible through a service that is also available in this network. Again, an agent collects the data in csv format and sends it to the database.

External sources for power grid information and energy price from the SMARD [12] service, CO_2 intensity from ElectricityMaps [13] and weather information from DWD [14] through the Brightsky software [15] are collected as well. They also use REST/JSON. This data is collected by the agent in the client network because the Internet is accessible from there.

Fig. 2 shows all data sources of the Digital Twin, the agent services responsible for data collection, the central InfluxDB and the data flow across network boundaries. Another component labelled “Digital Twin Logic” contains the software that models the system behaviour.

4. Optimisation of CO_2 emmisions

In this chapter we explore the potential for overall CO_2 emission savings by simulating optimised scheduling with the Digital Twin. The impact of two parameters is studied in our simulations: DVFS and a new

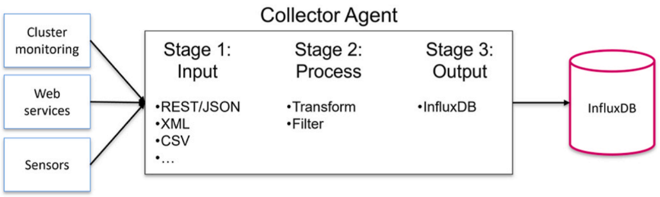

Fig. 1. Collector agent schematic with 3-stages for data collection, transformation and output to the database.

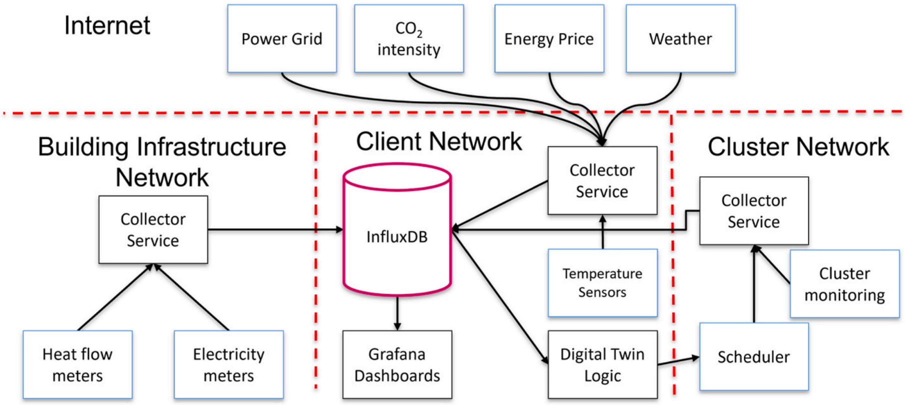

Fig. 2. Network overview of all Digital Twin data sources, agents and network boundaries.

scheduling policy. The currently applied default scheduling policy, FCFS with backfilling, and the processors' stock default base clock frequency are used to define the baseline for our experiments. For all these experiments, job traces with the improved Feitelson job model [16,17] are generated. This model allows to generate job traces with defined characteristics, e.g. to simulate different system utilisation.

The overall generated CO_2 emissions for executing the workload of the generated job traces are determined by the energy available in the energy grid at the time of the jobs' execution. Each source of energy has a different associated CO_2 emission to produce the electric energy. The Digital Twin tracks the current CO_2 intensity from Electricity Maps in the InfluxDB and can determine the emissions for a job. With the known CO_2 intensity of the energy at the time interval of the job's execution, the Digital Twin can determine the associated CO_2 emission of each specific job.

The HPC cluster at PTB uses compute nodes with dual-socket Intel Xeon Gold 6132 CPUs with a base frequency of 2.6 GHz and 192 GB of DDR4 RAM. The CPU can operate at frequencies from 1.0 GHz to 2.6 GHz. The manufacturer specified turbo frequencies of up to 3.7 GHz, but they are not supported by the node. All simulations in this paper use this node.

Modern processors allow for DVFS and may run at different manufacturer defined operating points of processor frequency and voltage. The operating system can select a clock speed based on the current system load level, or a specific frequency can be defined manually. The processor frequency is the main determining factor for the energy consumption of the CPU and in consequence of the whole system and is thus an important tuning parameter. However, not all operating points are equally efficient and energy consumption of operation points varies for different processor models. Moreover, the manufacturer selected stock base frequency for a processor model might not be the best choice in terms of energy to solution.

Another possible optimisation opportunity is related to the applied job scheduling policy. Each HPC system uses a batch scheduler that allocates incoming jobs on the available compute nodes. A common scheduling policy is FCFS, where jobs are scheduled in the order of their submission to the system. Backfilling is a common technique which allows the scheduler to execute smaller jobs out of order in appropriate gaps on otherwise unused nodes and usually improves system utilisation. This paper proposes an addition to this policy: the scheduler can take the current CO_2 intensity into account and delay jobs for a certain amount of time in order to find a future point in time with a lower CO_2 intensity. The search window is limited by a configurable look-ahead parameter. If the job wait time exceeds the look-ahead, e.g. because the system is too busy, then the scheduler will schedule the job without regard to the CO_2 intensity to avoid unreasonable waiting times for the user. For our experiments, a look-ahead window of one, three and five days was tested. A look-ahead of zero is equivalent to the usual FCFS with backfilling strategy without modifications.

The Feitelson Model itself does not generate job workload profiles. The specific job behaviour in terms of energy consumption is not modelled. For our experiments we have used the HPL benchmark [18] as a job profile. This benchmark is used for the TOP500 and Green500 lists and uses highly optimised numerical calculations. It poses an upper bound for possible energy consumption of job profiles we could possibly consider. The experiments were run using two different job arrival factors as parameters to the Feitelson Model. This factor is measured in seconds and defines the mean arrival interval between two jobs, its default being 1500 s. It can be used to tune the load on the system as a lower arrival factor results in a higher system utilisation. For a high utilisation with more than 80 percent, an arrival factor of 750 s was chosen (Fig. 3). A lower system utilisation was simulated using an arrival factor of 3000 s (Fig. 4).

First, the influence of DVFS on the performance was studied. The HPL benchmark was run on all possible operating points and the energy consumption of the node was captured through IPMI. With this data, the Time-to-Solution and Energy-to-Solution can be calculated as seen in Fig. 5. Given the set of operating points, the pareto-optimal set [19] of operation points can be determined as shown in the figure. We have selected the point at 1.8 GHz to run our simulations as it is pareto-optimal (given the HPL workload) in contrast to the default 2.6 GHz base frequency selected by the manufacturer. Simulations at default 2.6 GHz frequency were performed for comparison. In our simulations, we have taken into account the performance and energy consumption characteristics of operation points by adapting a node's energy

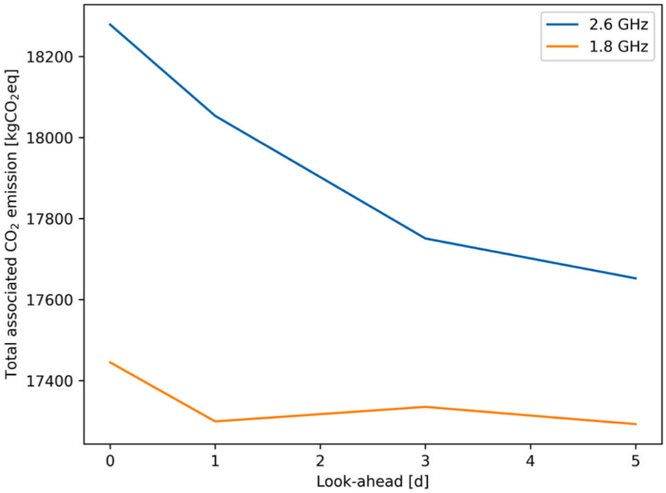

Fig. 3. Total associated CO_2 emission for an arrival factor of 750 s.

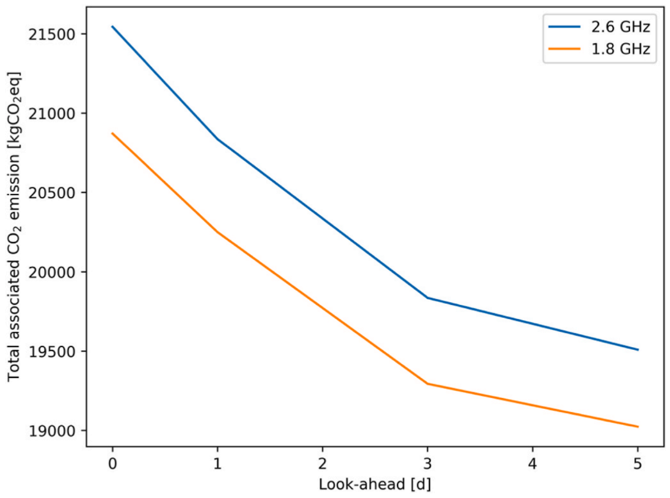

Fig. 4. Total associated CO$_2$ emission for an arrival factor of 3000 s.

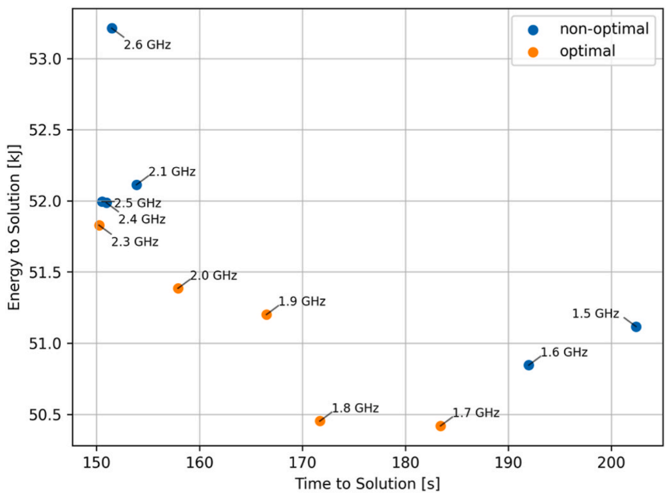

Fig. 5. Time-to-Solution and Energy-to-Solution of the HPL benchmark on a dual-socket Intel Xeon Gold 6132 node with a base frequency of 2.6 GHz. Pareto-optimal operating points coloured in orange.

consumption accordingly and by extending job runtime by the Time-to-Solution difference between the two frequencies.

The findings for improved HPC operation reducing CO$_2$ footprint are summarised in Figs. 3 and 4.

The isolated effect of DVFS can be seen with the look-ahead of 0, where both scheduling methods behave the same. In the high utilisation case with a 750 s arrival factor, the potential savings of DVFS are of the order of 4 percent and for the low utilisation 1500s arrival factor case of 3 percent.

The effect of applying the additional look-ahead at the base frequency can be seen in the blue lines in both figures. In both cases, the improved scheduler can find better starting points with overall lower CO$_2$ emissions. Savings vary depending on the chosen look-ahead and utilisation from 1 percent for one day look-ahead in the high utilisation case up to 9 percent in the 5-day look-ahead low utilisation case. Two additional observations can be made: with increasing look-ahead the idle energy usage of the nodes has a higher impact and thus at some point the idle costs exceeds possible savings. The saving potential thus depends on the load of the system as in the low utilisation case there is more freedom for the scheduler to move jobs.

If we combine DVFS with a look-ahead, in both utilisation cases, this performed better than using DVFS alone. For the 750 s case, savings are limited to below one percent though. This is due to the high overall system utilisation that is further increased by applying DVFS as job execution times increase. This gives the scheduler not enough space to move jobs and thus limits savings. In the low utilisation case, the savings are larger with about 11 percent compared to no optimisations and still 8 percent compared to just DVFS. With increasing look-ahead, the idle cost contributes more to the overall cost and thus also here, with increasing look-ahead idle cost will eventually exceed the savings.

5. Conclusions

A Digital Twin relies on measurement data from sensors placed throughout the data centre. The agents use a three-stage pipeline with an input stage for the sensor protocols, a processing stage for filtering and transforming the measurement data into a format compatible with the database and an output stage that sends the data to the database. Through agents, it is possible to transfer this data to the Digital Twin across network boundaries while maintaining the necessary separation of the individual networks.

HPC requires electric energy for the operation. During the production of this energy, CO$_2$ is emitted. The composition of the energy in the electric grid and thus the associated CO$_2$ emissions vary. The scheduler can take this into account, to reduce the CO$_2$ emissions associated with the operation of the cluster. Through simulations with the Digital Twin, two control variables have been tested for their saving potential: DVFS and a modified scheduling mechanism with deliberate delays. DVFS shows a saving potential of around 3 to 4 percent and the new scheduling mechanism with saving varying between 1 and 9 percent. Both methods proved to be viable options and can be combined for additional savings of up to 11 percent at the cost of longer job wait and execution times. Savings are furthermore subject to overall system utilisation and the trend in the energy mix and thus CO$_2$ intensity. In the future, these methods can be adapted for the real-world system through modification of the scheduler such as Slurm [20], an open-source scheduler, which allows to implement new scheduling algorithms.

This paper focused on one metric, the CO$_2$ emissions. Other metrics such as energy cost or overall energy usage are also possible optimisation dimensions and have not been discussed in this paper.

References

[1] United Nations, Transforming our world: the 2030 agenda for sustainable development, Online, https://undocs.org/en/A/RES/70/1, 2015. (Accessed 20 April 2024).

[2] A. Kammeyer, F. Burger, D. Lübbert, K. Wolter, Towards an HPC cluster digital twin and scheduling framework for improved energy efficiency, in: Proceedings of the 18th Conference on Computer Science and Intelligence Systems, 2023, pp. 265–268, https://doi.org/10.15439/2023F3797. Online. (Accessed 20 April 2024).

[3] H. van der Valk, H. Haße, F. Möller, B. Otto, Archetypes of digital Twins, Bus. Inform. Syst. Eng. 64 (3) (2022) 375–391, https://doi.org/10.1007/s12599-021-00727-7, 2022.

[4] ISO Central Secretary, Digital twin – concepts and terminology, Int. Org. Standard. (2023). Online, https://www.iso.org/standard/81442.html. (Accessed 20 April 2024).

[5] N.A. Simakov, M.D. Innus, M.D. Jones, R.L. DeLeon, J.P. White, S.M. Gallo, A. K. Patra, T.R. Furlani, A Slurm simulator: implementation and parametric analysis, high performance computing systems, Perform. Model. Benchmark. Simul. (2018) 197–217, https://doi.org/10.1007/978-3-319-72971-8_10. Online. (Accessed 20 April 2024).

[6] N.A. Simakov, R.L. Deleon, Y. Lin, P.S. Hoffmann, W.R. Mathias, Developing accurate Slurm simulator, practice and experience, in: Advanced Research Computing, 2022, pp. 197–217, https://doi.org/10.1145/3491418.3535178. Online. (Accessed 20 April 2024).

[7] A. Jokanovic, M. D'Amico, J. Corbalan, Evaluating SLURM simulator with real-machine SLURM and vice versa, IEEE/ACM Perform. Model. Benchmark. Simulat. High Perform. Comput. Syst. (2018) 72–82, https://doi.org/10.1109/PMBS.2018.8641556. Online. (Accessed 20 April 2024).

[8] T. Ohmura, Y. Shimomura, R. Egawa, H. Takizawa, Toward building a digital twin of job scheduling and power management on an HPC system, job scheduling strategies for parallel processing, 47–67. Online, https://doi.org/10.1007/978-3-031-22698-4_3, 2023. (Accessed 20 April 2024).

[9] J.M. Kunkel, H. Shoukourian, M.R. Heidari, T. Wilde, Interference of billing and scheduling strategies for energy and cost savings in modern data centers, Sustain. Comput.: Inform. Syst. 23 (2019) 49–66, https://doi.org/10.1016/j.suscom.2019.04.003.

[10] X. Yang, Z. Zhou, S. Wallace, Z. Lan, W. Tang, S. Coghlan, M. Papka, Integrating dynamic pricing of electricity into energy aware scheduling for HPC systems, in: International Conference for High Performance Computing, Networking, Storage and Analysis, 2013, pp. 1–11, https://doi.org/10.1145/2503210.2503264. Online. (Accessed 20 April 2024).

[11] J. Eitzinger, T. Gruber, A. Afzal, T. Zeiser, G. Wellein, ClusterCockpit — a web application for job-specific performance monitoring, in: 2019 IEEE International Conference on Cluster Computing, 2019, pp. 1–7, https://doi.org/10.1109/CLUSTER.2019.8891017. Online. (Accessed 20 April 2024).

[12] Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen (BNetzA), SMARD - Strommarktdaten, Stromhandel und Stromerzeugung in Deutschland, Online, https://www.smard.de/home/marktdaten, 2023. (Accessed 20 April 2024).

[13] Electricity Maps ApS, Electricity Maps, Online, https://www.electricitymaps.com/, 2023. (Accessed 20 April 2024).

[14] Deutscher Wetterdienst (DWD), Open data server of the German meteorological service, Online, https://opendata.dwd.de/, 2023. (Accessed 20 April 2024).

[15] Bright Sky Developers, Bright Sky JSON API for DWD's open weather data, Online, https://brightsky.dev/, 2023. (Accessed 20 April 2024).

[16] D.G. Feitelson, Packing schemes for gang scheduling, job scheduling strategies for parallel processing, 89–110. Online, https://doi.org/10.1007/BFb0022289, 1996. (Accessed 20 April 2024).

[17] D.G. Feitelson, M.A. Jettee, Improved utilization and responsiveness with gang scheduling, job scheduling strategies for parallel processing, 238–261. Online, https://doi.org/10.1007/3-540-63574-2_24, 1997. (Accessed 20 April 2024).

[18] A. Petitet, R.C. Whaley, J. Dongarra, A. Cleary, HPL - a portable implementation of the high-performance linpack benchmark for distributed, Mem. Comput. (2018). Online, https://www.netlib.org/benchmark/hpl/. (Accessed 20 April 2024).

[19] N. Sudermann-Merx, Fortgeschrittene modellierungstechniken, in: N. Sudermann-Merx (Ed.), Einführung in Optimierungsmodelle, Springer Berlin Heidelberg, Berlin, Heidelberg, 2023, pp. 161–193, https://doi.org/10.1007/978-3-662-67381-2_7.

[20] A.B. Yoo, M.A. Jette, M. Grondona, SLURM: simple linux utility for resource management, job scheduling strategies for parallel processing, 44–60. Online, https://doi.org/10.1007/10968987_3, 2003. (Accessed 20 April 2024).

Alexander Kammeyer[a,b,*], Florian Burger[a], Daniel Lübbert[a], Katinka Wolter[b]

[a] *Physikalisch-Technische Bundesanstalt, Abbestraße 2-12, 10587, Berlin, Germany*

[b] *Freie Universität Berlin, Takustraße 9, 14195, Berlin, Germany*

[*] Corresponding author. Physikalisch-Technische Bundesanstalt, Abbestraße 2-12, 10587, Berlin, Germany.

E-mail addresses: alexander.kammeyer@ptb.de, a.kammeyer@fu-berlin.de (A. Kammeyer), florian.burger@ptb.de (F. Burger), daniel.luebbert@ptb.de (D. Lübbert), katinka.wolter@fu-berlin.de (K. Wolter).