Proceedings of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 385–393 DOI: 10.15439/2024F1066 ISSN 2300-5963 ACSIS, Vol. 39

HPC operation with time-dependent cluster-wide power capping

Alexander Kammeyer*† [0000−0002−7858−0354], Florian Burger *[0000−0003−4745−5515], Daniel Lübbert* [0000−0003−3852−5665] and Katinka Wolter† [0000−0002−8630−0869]
*Physikalisch-Technische Bundesanstalt, Abbestraße 2-12, 10587 Berlin, Germany
Email: {alexander.kammeyer, florian.burger, daniel.luebbert}@ptb.de
†Freie Universität Berlin, Takustraße 9, 14195 Berlin, Germany
Email: {a.kammeyer,katinka.wolter}@fu-berlin.de

***Abstract*—HPC systems have increased in size and power consumption. This has lead to a shift from a pure performance centric standpoint to power and energy aware scheduling and management considerations for HPC. This trend was further accelerated by rising energy prices and the energy crisis that began in 2022.**

Digital Twins have become valuable tools that enable energy and power aware scheduling of HPC clusters. This paper uses an existing Digital Twin and extends it with a node energy model that allows the prediction of the cluster power consumption. The Digital Twin is then used to simulate system-wide power capping for different energy shortages functions of varying degree. Different policies are proposed and tested towards their effectiveness in improving the job wait times and overall throughput under limiting conditions.

Based on a real world HPC cluster, these policies are implemented. Depending on the pattern of the energy limitation and workload, improvements of up to 40 percent are possible compared to scheduling without policies for these conditions.

I. Introduction

HIGH-Performance Computing (HPC) is used, when a single computer is either too slow or too small for a single problem. Multiple computers, so called compute nodes or just nodes, are connected to solve the problem cooperatively. HPC systems have grown in size and capability over the years [1]. Simultaneously, their energy consumption did also grow, with the current top systems using exceeding 20 MW. Energy prices have increased as well, especially since the Russian war against Ukraine and the energy crisis in Europe in 2022 and 2023. The Physikalisch-Technische Bundesanstalt (PTB) operates a HPC cluster with approximately 30 kW installed power for research purposes.

During the 2022 energy crisis [2], Germany implemented "Ordinances on energy saving" [3] that contained measures to reduce energy consumption together with an appeal to public institutions, companies and private households to reduce their overall energy usage. In the case of an immediate energy shortage, power capping and load shedding measures have been discussed. These measures would require immediate power reduction on short notice. As part of PTB's strategy [4] towards energy-efficient HPC, continuous cluster operation under reduced energy availability is one of the goals. This paper presents policies and scheduling strategies for an HPC cluster to continue operation under these conditions.

Digital Twins are virtual representations of real-world objects, such as HPC systems. They collect data about the object and contain models that allow to simulate the behaviour and states of the real object. Over the past decade, they have grown in popularity in many industry applications and begin to see adoption in the HPC domain as well. Scheduling simulations are common in the area of HPC system research and an ideal basis for an HPC Digital Twin as they allow to model the system. In this paper they are used to test and verify the policies.

Figure 1 shows the mean weekly power consumption of the PTB campus where the HPC cluster is located. It shows a base power consumption of the campus at around 270 kW with 5 peaks for the traditional five workdays with very minor peeks on the weekend. With a possible power limitation, there would still be a similar pattern with a day-night-cycle for the workdays. This would allow a cluster operation at night with possible fewer restrictions. While the cluster allows very long jobs, a shift to a day and night cycle requires restrictions on the job length, so that jobs can be completed at night.

The trivial solution to power limitations is to turn off the HPC cluster entirely. The campus energy management system can also forcefully disconnect the HPC cluster or other large consumers from the power supply, if required. This does not guarantee a graceful shutdown and might lead to data loss. However, depending on the limitations of the energy usage, the cluster might remain operational under reduced load or with some of the nodes turned off. The system administrator could turn off nodes manually. This is a very coarse-grained approach. With a more fine grained approach towards power capping of the system and knowledge of the node power consumption, more nodes can remain online. Another common technique is Dynamic Voltage and Frequency Scaling (DVFS). With DVFS, the speed of the processors and thus the energy consumption of the compute nodes can be set dynamically. The Digital Twin of the HPC cluster can be used to estimate the energy consumption of individual jobs and plan accordingly while monitoring the overall system power usage to guarantee the operation within the defined limits. This entire process is also automatic, requiring no manual intervention from the operators.

This paper makes the following contributions:

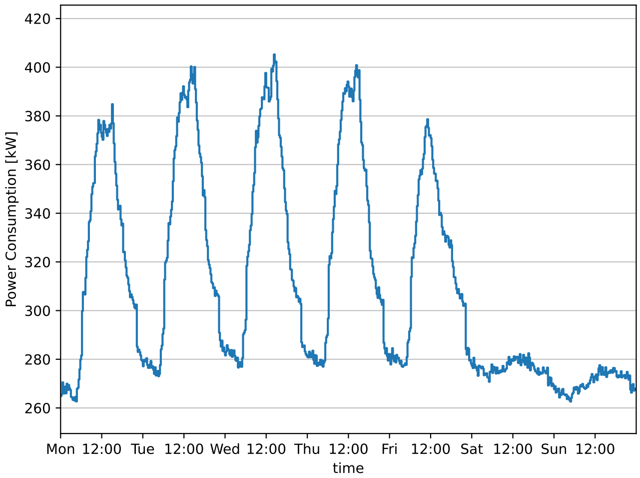


Figure 1: Mean power consumption on the west-side of PTB's Berlin site. The HPC cluster is located in this part of the campus. The graph shows the mean weekly power consumption of 2023 in 15 minute intervals.

- Support for heterogeneous HPC systems with different nodes is implemented for the scheduling simulation in the Digital Twin. Previously, the simulation could only handle one node type. The simulation can also simulate DVFS of the individual nodes with regard to runtime and power consumption. For this purpose, a node energy model is created.
- Different policies, including scheduling fewer jobs to decrease overall system load, shutting down unused compute nodes and DVFS on a per-job level with the node energy model, are implemented in the scheduling simulation.
- The Digital Twin is extended with these policies and through scheduling simulations with different job traces and power limits it is tested that the Digital Twin ensures the system-wide power cap of the cluster. This allows continuous cluster operation under reduced energy availability without manual intervention.

II. Related Work

With increasing energy consumption of HPC systems, research focuses more and more on energy and power related questions. The survey by [5] gives an overview of tools for energy and power management in contemporary HPC systems from a single node all the way to grid systems. It also shows research towards power prediction for different parts of HPC systems. A similar survey by [6] is done about power-aware scheduling.

The energy management framework for supercomputers (EAR) [7] is an accounting, control and optimisation framework for HPC systems. This system requires MPI profiling to create job profiles and uses an algorithm called DynAIS to detect different hotspots within an application. An Energy-aware job scheduling strategy on top of EAR [8] tries to place jobs in heterogeneous clusters based on the job profile.

Similar to the node energy model in chapter III-D, a power profile for different applications or types of applications have been done for WZ factorisation [9] and matrix factorisation [10]. They also run a benchmark on different frequencies but did not determine the pareto-optimal frequencies. The results cannot be used directly here because of different hardware architectures and a different selection of applications. While the node energy model and the papers look at the power cap from the frequency side, the power cap can directly implemented via a true power limit through a driver yielding similar results [11].

A Digital Twin, as defined by [12], is a virtual representation of a real-world object, in the scope of this paper an HPC cluster. Several data inputs are integrated into the Digital Twin, which handles the incoming data, processes it and stores it. With a bi-directional link between the real-world object and the Digital Twin, they can influence each other as changes in the real world are reflected in the virtual world and vice versa. This requires regular synchronisation to ensure consistency, however, by definition, a permanent, immediate synchronisation is not necessary. The Digital Twin must enable interoperability with other systems. The concept of the Digital Twin has gained such importance and has found wide adoption, that it has been standardised by ISO [13].

Digital Twins of HPC systems aid the system operators by allowing to test configuration changes, policies and different scheduling algorithms with altering the actual cluster. Possible negative effects are avoided this way. Simulating the scheduler is common practice in HPC research. Different simulations have been created, e.g. based on the Slurm scheduler [14], [15], [16] or based on Digital Twins [17]. Scheduling in regard to power consumption and pricing has been demonstrated by [18] for different billing strategies by delaying jobs for a static price model. A day and night price model with a 0-1-knapsack strategy is presented by [19]. With the transformation to renewable energies in the energy production, a scheduler can also take CO_2 emissions into account as shown by [20].

III. Preliminary Work

This section presents the Digital Twin for the HPC cluster of PTB and how it has been extended for this paper with support for heterogeneous nodes and a power model. The power model is later used for the power prediction for the power capping.

A. PTB's HPC cluster

The PTB operates a relatively small cluster at around 30 kW installed power. This amounts to approximately 10 percent of the overall power consumption on campus (Figure 1). The cluster is equipped with two different CPU nodes. Nodes of the first node type are equipped with two Intel Xeon E5-2690 v4 [21] each. The second node type is also a dual-socket system with two Intel Xeon Gold 6132 CPUs [22] per node. Both processor types are operating at a base frequency

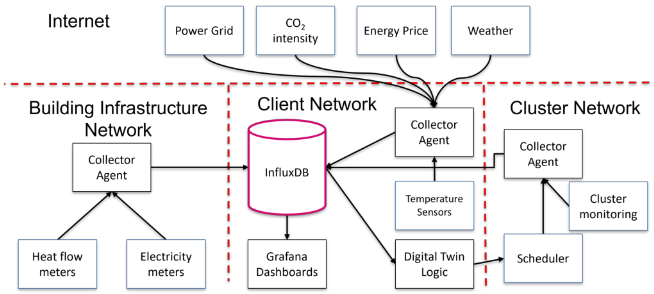


Figure 2: Network overview of all Digital Twin data sources, agents and network boundaries. [20]

of 2.6 GHz. Each node of the first type has 256 GB of DDR4 RAM while each node of the second type has 192 GB. The CPU can operate at frequencies from 1.2 GHz to 3.5 GHz (1.0 GHz to 2.6 GHz), for the first (second) node type, respectively. The manufacturer specified turbo frequencies of up to 3.7 GHz are disabled on the second node type. The cluster provides a total of 60 nodes for applications with 24 nodes of the first type and 36 nodes of the second type.

Both CPU designs are dated. The first was released in 2016 while the second was launched one and a half years later in 2017. Newer CPU generations can be expected to be more energy efficient. In fact, Moore's Law [23] states that the number of transistors doubles every one and a half years. Closely linked to the number of transistors is the performance and often also energy consumption. For the purpose of this study, the performance of the two designs are used as examples. The mechanisms presented in later chapters work independent of specific CPU designs and the underlying hardware is configured through the node energy model, presented in section III-D, in the Digital Twin.

B. Digital Twin Components and Layout

PTB is actively developing a Digital Twin for their HPC cluster [20]. The Digital Twin integrates all sensor data of the data centre as well as the cluster itself. Most of the data is time-series measurement data, thus the central database of the Digital Twin is an InfluxDB. The data centre is equipped with a multitude of sensors, such as electric energy meters, heat flow meters and temperature sensors. Additionally, external data sources, such as energy generation, weather forecast and energy CO_2 intensity, are also integrated into the Digital Twin. For security reasons, the sensors are separated into a so-called building infrastructure network. The HPC cluster also has a separate network. The Digital Twin exchanges data via well-defined interfaces between these networks. The entire network layout with all sensors, meters and data sources is shown in Figure 2. The Digital Twin also contains a scheduling simulation as the digital representation of the cluster behaviour. This simulation uses the data from the InfluxDB to get the system state and can use job traces to test system configurations. This allows the system administrator to test configurations with altering the production system. Simulations are also cheaper, since they do not require actual jobs to run on the system and thus use far less energy. The simulation is event-based and use job traces from the cluster or use the parallel workload format [24].

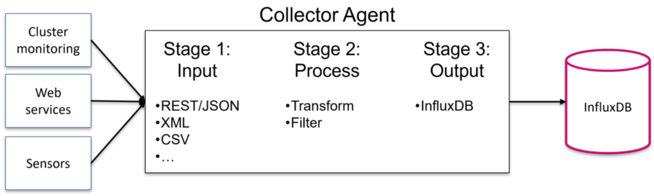


Figure 3: Collector agent schematic with 3-stages for data collection, transformation and output to the database. [20]

C. Data Collector Agent

The collector agents use a three-stage pipeline to collect, transform and send data to the central database. The sensors and data sources each use different protocols such as REST/JSON, XML and M-BUS. The database uses a custom format that can be accessed through a client library. The agent first collects the data through the appropriate protocol, then transforms the data into a format supported by the database and finally sends it to the database. This allows to add new sensors and protocols easily, as only the corresponding stages need to be adapted. The architecture of the agent is shown in Figure 3.

D. Node Energy Model

The cluster has different nodes with different hardware and thus different energy consumption. A node energy model has been created that allows the simulation to trace the energy consumption of different job types running at different frequencies. Other frameworks, such as EAR [7], rely on injecting code into the application to trace job behaviour. For this study, a less intrusive approach has been chosen. Two benchmarks and two applications have been selected and run at different frequencies to create the node energy model. The selected applications represent common applications on the HPC cluster.

The High-Performance Linpack (HPL) [25] is a common HPC benchmark and is also used to create the TOP500 ranking. It is a highly optimised linear equation solver. The benchmark uses vector instructions like AVX. These instructions require large amounts of energy and thus the benchmark is suitable as an upper bound for the node energy model. It is unlikely that a real application uses more energy than the HPL benchmark.

Another common HPC benchmark is the High Performance Conjugate Gradients (HPCG) [26]. The HPL benchmark is not representative of all HPC applications and the HPCG tries to complement this with a broader set of operations with different data access patterns that are harder to optimise than the pattern used by the HPL benchmark.

The first of the real applications is Open Field Operation And Manipulation (OpenFOAM) [27]. It is a computational

Table I: The Node Energy Model

Node Type	Job type	Frequency	Power	Scaling factor
1	HPL	3.2 GHz	415 W	0.966
		2.6 GHz	388 W	1.000
		2.2 GHz	322 W	1.163
	HPCG	3.2 GHz	147 W	0.884
		2.6 GHz	137 W	1.000
		1.5 GHz	108 W	0.932
	OpenFOAM	3.5 GHz	266 W	0.922
		2.6 GHz	240 W	1.000
		2.3 GHz	185 W	1.205
	Geant4	2.8 GHz	191 W	0.894
		2.6 GHz	183 W	1.000
		2.3 GHz	156 W	1.177
	Idle	-	57 W	-
	Offline	-	5 W	-
2	HPL	2.6 GHz	349 W	1.000
		2.3 GHz	349 W	0.992
		1.8 GHz	298 W	1.133
	HPCG	2.6 GHz	170 W	1.000
		2.3 GHz	161 W	0.926
		1.8 GHz	146 W	1.067
	OpenFOAM	2.6 GHz	273 W	1.000
		2.5 GHz	228 W	1.107
		2.3 GHz	212 W	1.188
	Geant4	2.6 GHz	219 W	1.000
		2.2 GHz	181 W	1.159
		2.0 GHz	167 W	1.311
	Idle	-	51 W	-
	Offline	-	5 W	-

fluid dynamics package that is commonly used by PTB researchers for investigating flows through pipes and other geometries.

The second selected application, Geometry and Tracking (Geant4) [28], is a Monte Carlo simulation toolkit for studying particles passing through matter. It is widely used in various fields such as high energy physics, medical physics and others.

Each of the four jobs was run on the two node types on all supported frequencies. The energy consumption of each job was monitored through IPMI. This allows to calculate two metrics for each frequency: Time-to-Solution (TtS) and Energy-to-Solution (EtS). Figures 4a and 4b show two such results for OpenFOAM and the HPL benchmark respectively on the newer node type 2. With regard to the two metrics TtS and EtS, the pareto front [29], [30] for each application and node type is computed, shown as the orange dots. These points are not dominated by any other point. For the purpose of this paper, from each optimal point set, two points have been chosen together with the processor base frequency of 2.6 GHz. This allows the simulation to choose from three points for each job type. Applying DVFS also changes the job execution time. The simulation adjusts for that by multiplying the job length with a factor based on the runtime of the job compared to the base frequency of 2.6 GHz. The node model also contains values for idle and offline power consumption of the nodes. A node consumes energy when offline, because the Wake-on-LAN functionality needs to listen for the magic packet. Table I summarises the frequencies, power and scaling factor.

IV. Problem definition

With the begin of the energy crisis in Europe in 2022, rising energy prices and energy scarcity became a concern for HPC operators. The way a cluster uses energy can be controlled through the scheduler and resource manager. Possible scheduling policies in terms of energy cost, e.g. through delaying jobs, have been discussed in Chapter II and have been shown with a Digital Twin [4]. Regarding energy scarcity, a HPC cluster has a variable energy consumption mostly defined by the compute nodes and the jobs running on them. This paper focuses on policies that can be implemented to continue stable operation under reduced energy availability conditions when the limitations are known in advance.

Energy scarcity can arise through different factors: an insufficient power supply to the entire campus from the energy supplier or an insufficient distribution inside the campus resulting in a scarcity for the cluster. The power capping functionality can also be used to reduce the thermal output of the cluster in case of problems or limitations of the cooling infrastructure.

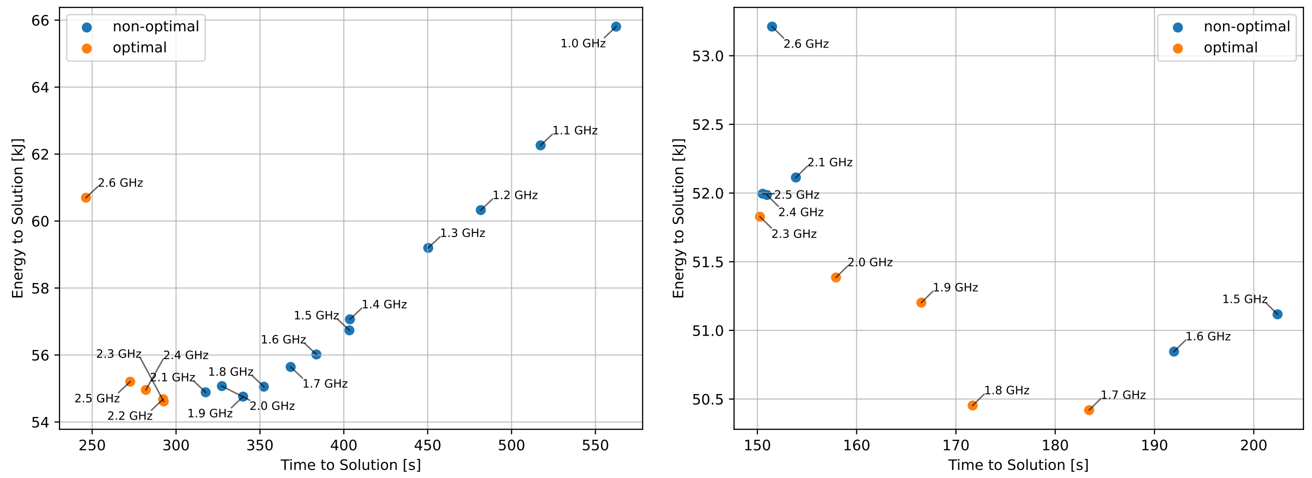
A data centre can be viewed as four pillars [31]: building infrastructure, system hardware, operating software and user applications. Changes to the first and second pillar are possible but often involve installing new hardware or larger construction work. Both is time consuming and not suitable for short term measures against shortages. This paper focuses on changes in the system software, the third pillar, that can be applied by the HPC system administrator immediately.

A. Proposed Policies

The main task of a batch scheduler is to allocate resources to jobs. The current cluster configuration requires users to select one of the two node types. Therefore, the scheduler must put jobs entirely on one of the two node types. The scheduler cannot select a different node type or make mixed allocations. If this criterion was to be relaxed, the scheduler could select nodes first, that have a lower EtS for the given job type.

The nodes contribute the most to the power consumption of the cluster. The energy model in Table I shows, that both types have an idle consumption of 57 W and 51 W respectively. This results in an idle consumption of 3204 W for the entire cluster. As an energy saving measure, nodes that are currently not used, can or must be turned off to stay below the power limit. An interval of 5 minutes has been selected as policy for this case. If a node is idle longer, the Digital Twin will shut it down and restart it, if needed. The boot time of a node is about 2 minutes, after which the node can receive new jobs. If the power limits allows it, the Digital Twin will keep a certain number of nodes online so that new jobs have a chance to start immediately.

DVFS allows to scale the energy consumption of the nodes according to the current limitations. Figure 4a shows all frequencies of a node type. While the node uses less power on these lower frequencies, the EtS and TtS is much higher compared to frequencies from the middle of the frequency range, with the two lowest frequencies, 1.1 GHz and 1.0 GHz



(a) Pareto front for OpenFOAM on node type 2 with all supported frequencies.

(b) Pareto front for HPL on node type 2. Frequencies below 1.5 GHz are not shown.

Figure 4: Two exemplary pareto-fronts. Pareto-optimal frequencies are coloured orange. The remaining frequencies are coloured blue.

exceeding the highest frequency 2.6 GHz in terms of EtS. The decisions has been made, not to use these frequencies as they are too inefficient. For each job type and each node type, a set of frequencies has been select as describes in Chapter III-D.

Changing the frequency, and thus speed of the processors, implies a change in job runtime. If the scheduler forces a certain frequency, it adjusts the job runtime with the scaling factor from Table I. As described above, jobs run exclusively on one of the node types. If they ran on different types simultaneously, the simulation would have to adjust the overall runtime accordingly and respect the different speeds of the nodes. Since this is currently not enabled on the cluster, it has not been integrated in the simulation.

For the purpose of this simulation, all power limitations are known in advance. Currently the cluster allows a very high job length of up to four weeks. This makes it impossible to react to any new limitations on short notice and jobs need to be cancelled. Given the cyclic nature of the power consumption on campus (Figure 1), a shorter maximum job runtime is required. The issue with crisis induced power limits is that they may also be only known a few days or even just hours in advance. Although no concrete time frame has been set, the notice will most likely not come four weeks beforehand.

The simulation currently does not support the cancellation of jobs for power capping. The Digital Twin tries to start a job only if the energy limit allows the jobs and nodes to run. In a real-world scenario, jobs might be cancelled, e.g. if their power profile drastically exceeds previously observed profiles by the Digital Twin. Simply re-scheduling the job might not be feasible because a job is not guaranteed to be free from side-effects. In case of job cancellation, manual intervention by the user is necessary.

B. Algorithm

Algorithm 1 Pseudo-code of the core scheduling routine

```
backfill_power(eligible_jobs) {
  for (Job j : eligible_jobs) {

    // test if nodes need to be booted
    // check if enough nodes are offline
    if (j.nodes > online_nodes(j) &&
        j.nodes <= online_and_offline_nodes(j)) {
      // how many nodes need to be booted?
      toboot = j.nodes - online_nodes(j);

      // check if the nodes would
      // exceed the power limit
      if(!check_power(j, get_offline(j, toboot))){
        continue;
      }
      boot_nodes(j, toboot);
    }

    // test of enough nodes are online
    // and available
    if (j.nodes <= online_nodes(j)) {

      // check if the job and nodes would
      // exceed the power limit
      if (!check_power_dvfs(j)) {
        continue;
      }

      assign_nodes(j);
      j.wait_time = tick - j.submit_time;
      running.add(j);
      eligible_jobs.remove(j);

      // trigger an event in the simulation
      // on job completion
      e = new JobEvent(tick + j.run_time_scaled,
               j, JobState.COMPLETED);
      eventQueue.add(e);
    }
  }
}
```

With the considerations from the previous section, a policy has been developed, that handles the cluster operation under reduced energy availability conditions. The scheduling simulation is event driven. Three different events are defined: a job event is triggered when a job is submitted or finished, a node event is triggered when a compute node gets assigned a job, finishes a job, starts or shuts down and a power event is triggered when the current power limit for the cluster changes. The simulation then handles the event and calls the scheduler. The core scheduling logic is shown as pseudo-code in Algorithm 1.

For each queued job, the algorithm first checks, if nodes need to be turned on (lines 6 and 7). *online_nodes()* returns the idle nodes the job can use and *online_and_offline_nodes()* checks whether enough nodes are idle or online. It does not make sense to boot nodes when the job cannot start. If that is the case, it is checked in line 13 with *check_power* whether the additional nodes, together with the projected energy consumption of the job, would exceed the power limit. The function *get_offline()* returns a list of offline nodes the scheduler would allocate to the job. If the limit is exceeded, the next job is checked. If not, the required nodes are booted. Each nodes takes two minutes to boot.

If a job has enough nodes available (line 21), the algorithm checks if the job fits within the limit with the *check_power_dvfs()* function in line 25. The function tests all DVFS settings from Table I and adjusts the job length if necessary. If the job power requirement exceeds the limit, the algorithm moves to next eligible job. Otherwise, the job gets assigned compute nodes, is started and removed from the eligible job list (lines 29-31). An event is created when the job finishes so that the scheduling simulation can mark the nodes available again. The event queue is a priority queue that sorts the events by their tick.

This algorithm is First Come, First Serve (FCFS) with backfilling. In this implementation, jobs are allowed to push back larger jobs. This design decision has been made because energy availability during a crisis situation is unclear. The cluster could stop operation entirely. Therefore, completing jobs gets precedence over the fairness criterion.

V. Evaluation

The previous chapter introduced the proposed policies. This chapter focuses on their implementation in the Digital Twin. The current cluster allows long job run times. The average power usage on campus peaks at around 400 kW (Figure 1). For this experiment, a limit of 300 kW is assumed. The power limit for the cluster is the value between the average consumption and 300 kW. The limit is capped at 5 kW power usage for the cluster under the assumption that some part of the power budget gets allocated to the cluster. As shown in Figure 5a, a job trace from the real cluster together with this power limit only allows a few jobs to start in between the limits and some more during the weekend. The rest is kept in queue until the limit ends due to their size.

Table II: Results of the scheduling experiment with the two power limit patterns and averages for 5 different job traces each.

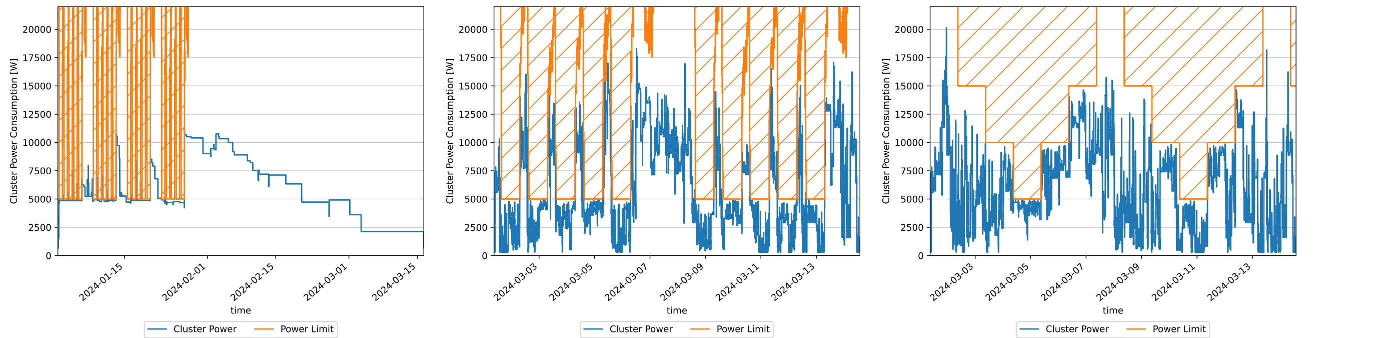
pattern	policy	wait time	sim run time
synthetic	limit	3.766 h	14.082 d
	shutdown	2.472 h	14.059 d
	dvfs	2.575 h	13.417 d
	dvfs + shutdown	2.289 h	13.414 d
real	limit	5.423 h	15.110 d
	shutdown	3.649 h	14.757 d
	dvfs	5.124 h	14.948 d
	dvfs + shutdown	3.415 h	14.663 d

As discussed in Chapter IV-A, shorter jobs can circumvent this issue. To generate such a job trace, the Feitelson job model [32], [33] is used. This model allows to configure the maximum job length together with some other parameters such as arrival rate of the jobs. It does not support different job types. They are generated at random and with equal probability using a discrete uniform distribution. For this experiment, two different limits are used for testing. The first one is the same as in Figure 5a, based on the campus power consumption (*real*). The second is a simpler, pyramid-shaped limit that uses the same lower bound of 5 kW (*synthetic*). Five different job traces with different seeds were generated for the following experiments.

The first step is to validate that the scheduler does not exceed the power cap. All compute nodes remained online and all jobs ran at the highest frequency. The scheduler can, if required, turn off individual nodes if it would otherwise exceed the power limit. They are re-booted when needed by a job. In a second step, the scheduler will turn off nodes pro-actively in order to save energy and thus allow more jobs to run. An idle node is turned off after 5 minutes. Finally, DVFS was added to the simulation. This allows the scheduler to start jobs at a lower frequency but also with less power. The results can be found in Table II.

The table contains two metrics: the mean job wait time and the overall run time of the simulation (Figures 5c, 6a and 6c). For the synthetic limit, the pro-active shutdown of compute nodes brings down the mean job wait time by 34 percent from 3.766 h to 2.472 h. Enabling DVFS gives an improvement of 32 percent. Combining node shutdown and DVFS further improves the wait time down to 2.289 h or 40 percent compared to the run without. The overall simulation run time on the other hand benefits more from DVFS than node shutdown. Here, the scheduler can start jobs at lower frequency and thus fit more jobs below the limit. This policy prefers smaller jobs which can in turn lead to an increase of the wait time for larger jobs.

The second experiment with the power limit based on the campus energy usage showed an improvement of 37 percent. The effect of DVFS on the wait time is smaller because of the longer periods of low power limits in this experiment. Combining both policies also yields a 5 percent improvement as in the first test case. For this simulation, the simulation length benefited more from the node shutdown than DVFS.



(a) Power trace of the PTB job trace showing four weeks of power limitations based on the campus energy usage.

(b) Power trace of a Feitelson job trace showing two weeks of power limitations based on the campus energy usage.

(c) Power trace of a Feitelson job trace showing three weeks of power limitations based on the synthetic power limit function.

Figure 5: Results of the experiments

For all experiments should be noted, that the results are dependent on factors such as job length, the pattern of the limitation and overall utilisation of the cluster. In this simulation, the lowest limit was at $5\,\mathrm{kW}$ while the idle consumption of the cluster is $3.2\,\mathrm{kW}$. This leaves only $1.8\,\mathrm{kW}$ for compute jobs.

VI. Conclusion and Future Work

This paper presented an approach to handling power limitations in the energy supply for an HPC cluster with the aid of a Digital Twin. The Digital Twin is able to trace and predict the power consumption of the HPC cluster. This allows the Digital Twin to control the HPC cluster and keep the overall power consumption below a defined threshold. This power capping capability is required, if the energy consumption of the HPC system needs to be limited.

Support for heterogeneous HPC clusters was added to the Digital Twin. Previously, only homogeneous HPC clusters with a single node type were supported. The Digital Twin can handle clusters with multiple types of nodes, each with an individual power consumption. Based on an example HPC cluster at PTB, a node power model was created (Table I). This model allows the Digital Twin to use DVFS to run jobs at different clock speeds and influence their energy consumption.

This paper proposed three strategies to continue operation under reduced load: trace power consumption and only start jobs that fit in the power budget, shutdown idle nodes when they are idle for a certain amount of time and DVFS to allow jobs to start with a reduced frequency set by the scheduler and also reduced power consumption (Algorithm 1).

These strategies were then compared with two different energy limits: one synthetic energy limit and one energy limit based on the campus energy consumption. The simulation showed, that the cluster can stay below the power cap and the target metrics were improved with the proposed policies, in some cases of up to 40 percent.

This paper presented a strategy and validated it in an experiment with a Digital Twin. The next step is to implement this strategy for the open-source scheduler Slurm [34] and verify the results on real hardware. This requires close coordination with the users of the HPC cluster as it affects the availability of the system. This paper relied on a node energy model with prior knowledge of the power profile of the applications. In a production scenario, the Digital Twin would need to learn new applications and use an estimate, e.g. of the very power intensive HPL benchmark, for unknown applications.

Acknowledgement

We would like to thank Holger Duzinski and Thomas Várdaru for their support of this research project. They provided us with helpful insight into the campus infrastructure and enabled access to sensors and meters without which the Digital Twin development would not be possible.

References

[1] Prometeus GmbH, “Top500 list,” May 2024. [Online]. Available: https://top500.org/

[2] A. Konopelko, L. Kostecka-Tomaszewska, and K. Czerewacz-Filipowicz, “Rethinking eu countries’ energy security policy resulting from the ongoing energy crisis: Polish and german standpoints,” *Energies*, vol. 16, no. 13, 2023. doi: 10.3390/en16135132. [Online]. Available: https://www.mdpi.com/1996-1073/16/13/5132

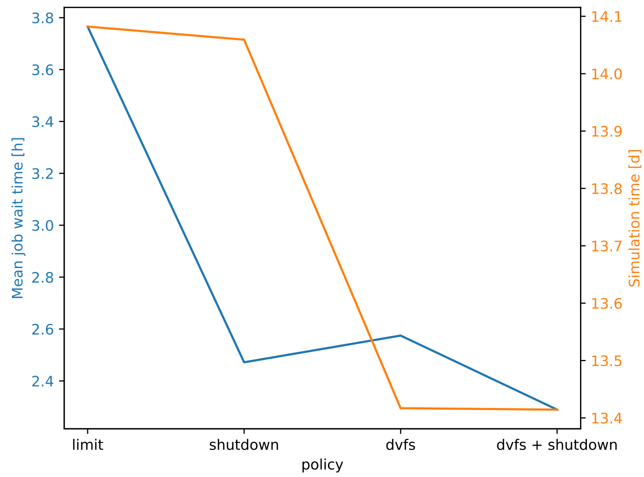
[3] Bundesministerium für Wirtschaft und Klimaschutz, “Ordinances on energy saving ensikumav and ensimimav,” Sep. 2022. [Online]. Available: https://www.bmwk.de/Redaktion/DE/Downloads/E/ensikumav.html

[4] A. Kammeyer, F. Burger, D. Lübbert, and 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*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 35, 2023. doi: 10.15439/2023F3797 p. 265–268.

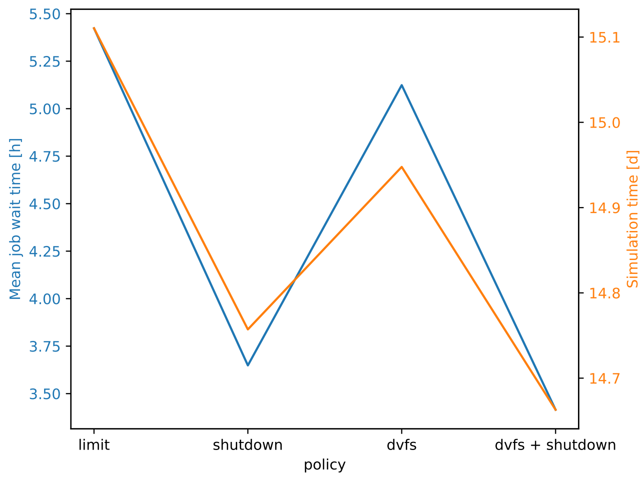
[5] P. Czarnul, J. Proficz, and A. Krzywaniak, “Energy-aware high-performance computing: Survey of state-of-the-art tools, techniques, and environments,” *Scientific Programming*, vol. 2019, p. 8348791, 2019. doi: 10.1155/2019/8348791. [Online]. Available: https://doi.org/10.1155/2019/8348791

[6] B. Kocot, P. Czarnul, and J. Proficz, “Energy-aware scheduling for high-performance computing systems: A survey,” *Energies*, vol. 16, no. 2, 2023. doi: 10.3390/en16020890. [Online]. Available: https://www.mdpi.com/1996-1073/16/2/890

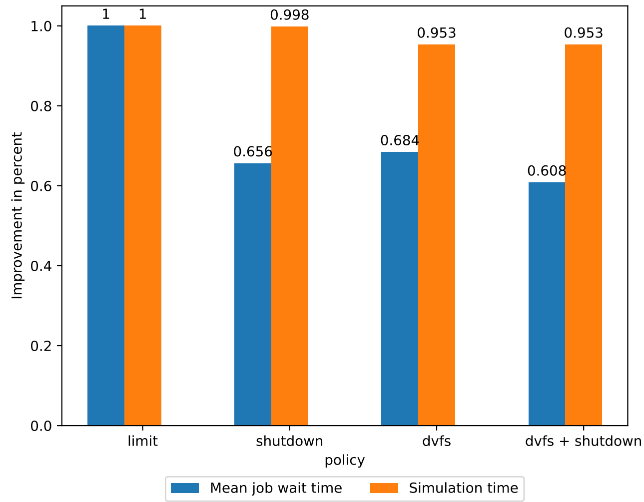
[7] J. Corbalan and L. Brochard, “Ear: Energy management framework for supercomputers,” *Barcelona Supercomputing Center (BSC) Working paper*, 2019.



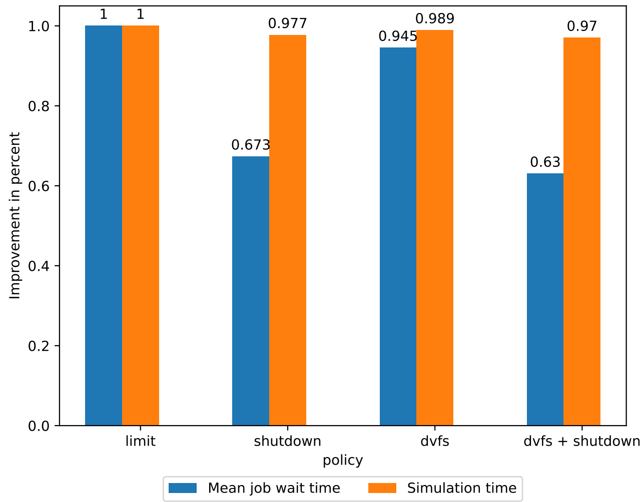
(a) Case of synthetic power limit function based on cyclic rising and falling limit. Absolute values for the two metrics mean job wait time and overall simulation time.



(b) Case of real power limit function based on campus power usage. Absolute values for the two metrics mean job wait time and overall simulation time.



(c) Case of synthetic power limit function. Relative values for the two metrics mean job wait time and overall simulation time normalised to the first policy.



(d) Case of real power limit function based on campus power usage. Relative values for the two metrics mean job wait time and overall simulation time normalised to the first policy.

Figure 6: Results of the experiments

[8] M. D'Amico and J. C. Gonzalez, "Energy hardware and workload aware job scheduling towards interconnected hpc environments," *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2021. doi: 10.1109/TPDS.2021.3090334

[9] B. Bylina, J. Bylina, and M. Piekarz, "Impact of processor frequency scaling on performance and energy consumption for wz factorization on multicore architecture," in *Proceedings of the 18th Conference on Computer Science and Intelligence Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 35, 2023. doi: 10.15439/2023F6213 p. 377–383.

[10] B. Bylina and M. Piekarz, "The scalability in terms of the time and the energy for several matrix factorizations on a multicore machine," in *Proceedings of the 18th Conference on Computer Science and Intelligence Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 35, 2023. doi: 10.15439/2023F3506 p. 895–900.

[11] A. Krzywaniak, J. Proficz, and P. Czarnul, "Analyzing energy/performance trade-offs with power capping for parallel applications on modern multi and many core processors," in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2018. doi: 10.15439/2018F177 p. 339–346.

[12] H. van der Valk, H. Haße, F. Möller, and B. Otto, "Archetypes of digital twins," *Business & Information Systems Engineering*, vol. 64, no. 3, p. 375–391, Jun 2022. doi: 10.1007/s12599-021-00727-7. [Online]. Available: https://doi.org/10.1007/s12599-021-00727-7

[13] ISO Central Secretary, "Digital twin – concepts and terminology," International Organization for Standardization, Geneva, CH, Standard ISO/IEC 30173:2023, Nov. 2023. [Online]. Available: https://www.iso.org/standard/81442.html

[14] N. A. Simakov, M. D. Innus, M. D. Jones, R. L. DeLeon, J. P. White, S. M. Gallo, A. K. Patra, and T. R. Furlani, "A slurm simulator: Implementation and parametric analysis," in *High Performance Computing Systems. Performance Modeling, Benchmarking, and Simulation*, S. Jarvis, S. Wright, and S. Hammond, Eds. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-72971-8_10. ISBN 978-3-319-72971-8 p. 197–217.

[15] N. A. Simakov, R. L. Deleon, Y. Lin, P. S. Hoffmann, and W. R. Mathias, "Developing accurate slurm simulator," in *Practice and Experience in Advanced Research Computing*, ser. PEARC '22. New York, NY, USA: Association for Computing Machinery, 2022. doi: 10.1145/3491418.3535178. ISBN 9781450391610. [Online]. Available: https://doi.org/10.1145/3491418.3535178

[16] A. Jokanovic, M. D'Amico, and J. Corbalan, "Evaluating slurm simulator with real-machine slurm and vice versa," in *2018 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, 2018. doi: 10.1109/PMBS.2018.8641556 p. 72–82.

[17] T. Ohmura, Y. Shimomura, R. Egawa, and H. Takizawa, "Toward building a digital twin of job scheduling and power management on an hpc system," in *Job Scheduling Strategies for Parallel Processing*, D. Klusáček, C. Julita, and G. P. Rodrigo, Eds. Cham: Springer Nature Switzerland, 2023. doi: 10.1007/978-3-031-22698-4_3. ISBN 978-3-031-22698-4 p. 47–67.

[18] J. M. Kunkel, H. Shoukourian, M. R. Heidari, and T. Wilde, "Interference of billing and scheduling strategies for energy and cost savings in modern data centers," *Sustainable Computing: Informatics and Systems*, vol. 23, p. 49–66, 2019. doi: 10.1016/j.suscom.2019.04.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221053791830297X

[19] X. Yang, Z. Zhou, S. Wallace, Z. Lan, W. Tang, S. Coghlan, and M. E. Papka, "Integrating dynamic pricing of electricity into energy aware scheduling for hpc systems," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '13. New York, NY, USA: Association for Computing Machinery, 2013. doi: 10.1145/2503210.2503264. ISBN 9781450323789. [Online]. Available: https://doi.org/10.1145/2503210.2503264

[20] A. Kammeyer, F. Burger, D. Lübbert, and K. Wolter, "Developing a digital twin to measure and optimise hpc efficiency." Submitted to IMEKO World Congress 2024, 2024.

[21] Intel Corporation, "Intel xeon processor e5-2690 v4," Jul. 2024. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/91770/intel-xeon-processor-e5-2690-v4-35m-cache-2-60-ghz.html

[22] ——, "Intel xeon gold 6132 processor," Jul. 2024. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/123541/intel-xeon-gold-6132-processor-19-25m-cache-2-60-ghz.html

[23] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, p. 114 ff., Apr. 1965. doi: 10.1109/N-SSC.2006.4785860

[24] D. G. Feitelson, D. Tsafrir, and D. Krakov, "Experience with using the parallel workloads archive," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, p. 2967–2982, 2014. doi: 10.1016/j.jpdc.2014.06.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731514001154

[25] A. Petitet, R. C. Whaley, J. Dongarra, and A. Cleary, "Hpl - a portable implementation of the high-performance linpack benchmark for distributed-memory computers," Dec. 2018, version 2.3. [Online]. Available: https://www.netlib.org/benchmark/hpl/

[26] J. Dongarra, M. A. Heroux, and P. Luszczek, "High-performance conjugate-gradient benchmark: A new metric for ranking high-performance computing systems," *The International Journal of High Performance Computing Applications*, vol. 30, no. 1, p. 3–10, 2016. doi: 10.1177/1094342015593158. [Online]. Available: https://doi.org/10.1177/1094342015593158

[27] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computer in Physics*, vol. 12, no. 6, p. 620–631, 11 1998. doi: 10.1063/1.168744. [Online]. Available: https://doi.org/10.1063/1.168744

[28] S. Agostinelli, J. Allison, K. Amako *et al.*, "Geant4—a simulation toolkit," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506, no. 3, p. 250–303, 2003. doi: 10.1016/S0168-9002(03)01368-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168900203013688

[29] N. Sudermann-Merx, *Fortgeschrittene Modellierungstechniken*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2023, p. 161–193. ISBN 978-3-662-67381-2. [Online]. Available: https://doi.org/10.1007/978-3-662-67381-2_7

[30] D. Kolossa and G. Grübel, "Evolutionary computation and nonlinear programming in multi-model-robust control design," in *Real-World Applications of Evolutionary Computing*, S. Cagnoni, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. ISBN 978-3-540-45561-5 p. 147–157.

[31] T. Wilde, A. Auweter, and H. Shoukourian, "The 4 pillar framework for energy efficient hpc data centers," *Computer Science - Research and Development*, vol. 29, no. 3, p. 241–251, Aug 2014. doi: 10.1007/s00450-013-0244-6. [Online]. Available: https://doi.org/10.1007/s00450-013-0244-6

[32] D. G. Feitelson, "Packing schemes for gang scheduling," in *Job Scheduling Strategies for Parallel Processing*, D. G. Feitelson and L. Rudolph, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. ISBN 978-3-540-70710-3 p. 89–110.

[33] D. G. Feitelson and M. A. Jettee, "Improved utilization and responsiveness with gang scheduling," in *Job Scheduling Strategies for Parallel Processing*, D. G. Feitelson and L. Rudolph, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997. ISBN 978-3-540-69599-8 p. 238–261.

[34] A. B. Yoo, M. A. Jette, and M. Grondona, "Slurm: Simple linux utility for resource management," in *Job Scheduling Strategies for Parallel Processing*, D. Feitelson, L. Rudolph, and U. Schwiegelshohn, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. doi: 10.1007/10968987_3. ISBN 978-3-540-39727-4 p. 44–60.