Smart Grid-aware Scheduling in Data Centres

Markus Mäsker,
Lars Nagel
and André Brinkmann
Zentrum für Datenverarbeitung
Johannes Gutenberg-Universität Mainz
{maesker, nagell, brinkman}@uni-mainz.de

Foad Lotfifar
and Matthew Johnson
School of Engineering and Computing Sciences
Durham University
{foad.lotfifar, matthew.johnson2}@durham.ac.uk

***Abstract*—In several countries the expansion and establishment of renewable energies result in widely scattered and often weather-dependent energy production, decoupled from energy demand. Large, fossil-fuelled power plants are gradually replaced by many small power stations that transform wind, solar and water power into electrical power. This leads to changes in the historically evolved power grid that favours top-down energy distribution from a backbone of large power plants to widespread consumers. Now, with the increase of energy production in lower layers of the grid, there is also a bottom-up flow of the grid infrastructure compromising its stability. In order to locally adapt the energy demand to the production, some countries have started to establish Smart Grids to incentivise customers to consume energy when it is generated.**

This paper investigates how data centres can benefit from variable energy prices in Smart Grids. In view of their low average utilisation, data centre providers can schedule the workload dependent on the energy price. We consider a scenario for a data centre in Paderborn, Germany, hosting a large share of interruptible and migratable computing jobs. We suggest and compare two scheduling strategies for minimising energy costs. The first one merely uses current values from the Smart Meter to place the jobs, while the other one also estimates the future energy price in the grid based on weather forecasts. In spite of the complexity of the prediction problem and the inaccuracy of the weather data, both strategies perform well and have a strong positive effect on the utilisation of renewable energy and on the reduction of energy costs. Our experiments and cost analysis show that our simple-to-apply low-cost strategies reach utilisations and savings close to the optimum.

I. Introduction

With the "Energiewende" (energy transition) [20], the German government decided to enforce a more sustainable energy development policy improving the overall energy efficiency and the share of renewable energy. Many other countries follow similar policies. The reasons are manifold and include the reduction of greenhouse gas emissions, the risk of nuclear accidents and the costs of and the dependency on fossil fuels. The shift from nuclear and coal-fired power plants towards wind and solar power plants results in a widespread energy generation in subgrids and in the decoupling of energy production and energy consumption. Therefore, the following problems need to be addressed:

1) Energy generation within distribution grids makes the grid more complex and causes problems because many of the transformers to the respective transmission grids are often not capable of transporting the (peak) energy produced by windmills and solar collectors. Since it would be costly to purchase high-performance transformers to adapt the grid to accommodate the mini power plants, it is desirable to consume energy locally when it is produced and adapt demand to production.
2) Energy is not always produced when it is needed and not always needed when it is produced. For this reason, it is desirable to adapt consumption to generation as long as there is no efficient way of storing energy.
3) Energy consumption should be contained since energy that is not used does not have to be generated.

Several countries try to overcome the first two problems by introducing a *Smart Grid* that monitors the state and the load flow of the electrical grid's elements and provides these data in real time such that measures can be taken if necessary. Stimuli for consumers are prices that reflect the situation in the grid. If too much energy is produced in an area, the price will drop for the consumers there.

In anticipation of the Smart Grid and dynamic energy prices, solutions for these problems were developed in the project "GreenPAD"[1] which considered a scenario in Paderborn, Germany. It focused on a local data centre offering computing services to research institutes and small companies. The main issue was to build a scheduler that schedules the incoming workload to time periods of energy surplus and thus lower energy prices.

In this paper we describe the scenario and challenges and evaluate two types of schedulers using different performance metrics. To keep the expenses low, we concentrate on schedulers that use either free or inexpensive data. Aside from a standard, CPU-optimised FIFO scheduler that is run for comparison, the schedulers utilise data from the Smart Grid including information about the current local energy production and consumption.

One of the schedulers, the so-called *green scheduler*, also analyses low-cost weather recordings and forecasts to predict the future energy surpluses and prices. Although this task is in principle more complex than the prediction of on-site solar and wind power plants (treated e.g. in [16], [15], [11], [12]), we show that our low-cost schedulers already suffice to increase the share of renewable energy to a nearly optimal value. The price to pay is an increase in the turnaround time so that one has to make a compromise between the green energy rate and the service quality. In terms of our assumed price model, the

[1] http://www.green-pad.de

978-3-901882-70-8 ©2015 IFIP

energy costs saved amount to about 13.7 % in a scenario where 16.9 % would be optimal. From these results it follows that, at least for small data centres, the purchase of more expensive weather or energy forecasts would not be profitable as they might not save the money they cost.

The paper is structured as follows: After a discussion of related work, we describe the scenario in more detail. In Section II we outline the software consisting of a scheduler and an energy prediction component, where the latter is only used by the Energy-efficient Scheduler. The schedulers are compared and evaluated in Section III before the paper is summarized and concluded in Section IV.

A. Related Work

The deployment of renewable energy has recently gained popularity in the IT industry [4], [9] and inspired projects in both, academia and industry, for example *DC4Cities*[2], *Parasol*[3], *GreenStar Network*[4], *GreenQloud*[5] and *Green Mountain*[6]. The main research challenge is the irregular power output of wind farms and solar collectors. Solutions to these problems usually include one or more of the five key aspects that were defined by Deng et al. [9]: 1) generation models, 2) prediction of renewable energy, 3) capacity planning, 4) scheduling within and 5) in between data centres. In this paper we concentrate mostly on the fourth point, but also consider the second one. Therefore, this survey first discusses publications related to energy prediction and then work about energy-aware scheduling.

Improvements in *numerical weather prediction* (NWP) and in power forecast algorithms have considerably improved the accuracy of the forecast models in the past decades [14]. The taxonomy of forecast models is so diverse that we cannot cover it completely, but only name a few models: direct time series forecasting (e.g. [6], [3]), time series models in combination with neural networks ([18], [13]), direct power forecast models with statistical improvements (e.g. [8], [13]), models dealing with non-linear power curves and the accuracy of NWP input (e.g. [14], [10]). For an extensive survey the reader is referred to Giebel et al. [10].

In [7] Brown and Renau introduce *ReRack*, a simulation environment for analysing the costs associated with the employment of renewable energy. The software includes an *optimizer* that uses a genetic algorithm to improve the system subject to a user-defined cost function. The paper suggests input and algorithms in the form of models and parameters, but only provides a very brief section about the actual application of the tool. Ren et al. present a framework in [21] that helps to reduce a data centre's energy costs and possibly its carbon footprint. The analysis uses linear programming and is based on the energy prices for on-site and off-site green energy as well as energy from other sources. Provided that the carbon footprint target is not too high, they show that the on-site generation of green energy can also reduce the costs.

The following papers discuss energy-aware schedulers and present software solutions for different scenarios: *SolarCore* [16] considers a system that relies on solar power as the main energy source, but automatically switches to grid power when solar energy drops below a threshold. By controlling the power state of servers, a green energy utilisation of 82% is achieved with little impact on performance. Solely relying on renewable sources, *Blink* [22] puts servers in active or inactive mode depending on the energy situation. Its major drawback is that using only renewable sources is unrealistic and causes unbounded performance degradation. *iSwich* [15] explores a design that puts servers into two groups: the first half is supplied with energy from the grid, the other half with on-site wind energy. Based on the availability of wind energy, *iSwitch* migrates load between the groups. The system introduced in [2] is a real-time scheduler for batch and service jobs based on off-site solar and wind energy production and they use short-term weather forecasts to get more precise energy predictions. The project *Parasol* at Rutgers University proposes the software systems *GreenSlot* [11] and *GreenHadoop* [12]. *GreenSlot* is a batch job scheduler for data centres which are powered by an on-site photovoltaic array and that use the electrical grid only as a backup. The scheduler predicts the available solar energy and places the jobs in such a way that their deadlines are met and that the utilisation of green energy is maximised. *GreenHadoop* is a similar system designed for Hadoop jobs. By deferring the map and reduce jobs, it tries to match the variable green energy supply.

Besides placing and migrating jobs within a single data centre, many papers consider the case of migrating jobs between geographically dispersed data centres. *GreenWare*, proposed by Zhang et al. [23], is a middleware that dispatches jobs to data centres based on local energy prices. The authors found that, if energy is dynamically priced based on the proportion of fossil energy, the usage of fossil energy can be significantly reduced. *Free Lunch* [1] co-locates data centres with renewable energy generation sites and migrates workload between data centres according to available power. *GreenNebula* [5], developed by the *Parasol* project, follows a similar approach. It extends the *OpenNebula* cloud manager and maximises the use of green energy by migrating VMs across data centres. In [17] Li et al. assume a dynamic pricing market and propose a collaboration framework for energy cost optimisation that couples data centres with the electricity market. They claim that this collaboration can reduce the costs by up to 75%. Niehörster et al. [19] propose a scheduling mechanism for a dynamic pricing model based on the spot market of the *European Energy Exchange*[7] (EEX). A multi-agent system, which is aware of the price, is placed on top of a cloud's infrastructure layer. Scheduler agents collaborate with worker agents that monitor the jobs during their execution and control the system such that it fulfils the service-level agreements while minimising the electricity costs.

B. Scenario

This paper describes schedulers that increase the usage of locally produced energy and thereby reduce the energy costs, but in contrast to most related work (i.e. [2]), a more complex scenario is considered. The energy price depends on the surplus in the local grid so that energy predictions are only useful if they are made for the whole local grid involving suppliers

[2]http://www.dc4cities.eu
[3]http://parasol.cs.rutgers.edu
[4]http://www.greenstarnetwork.com
[5]https://www.greenqloud.com
[6]http://www.greenmountain.no
[7]http://www.eex.com

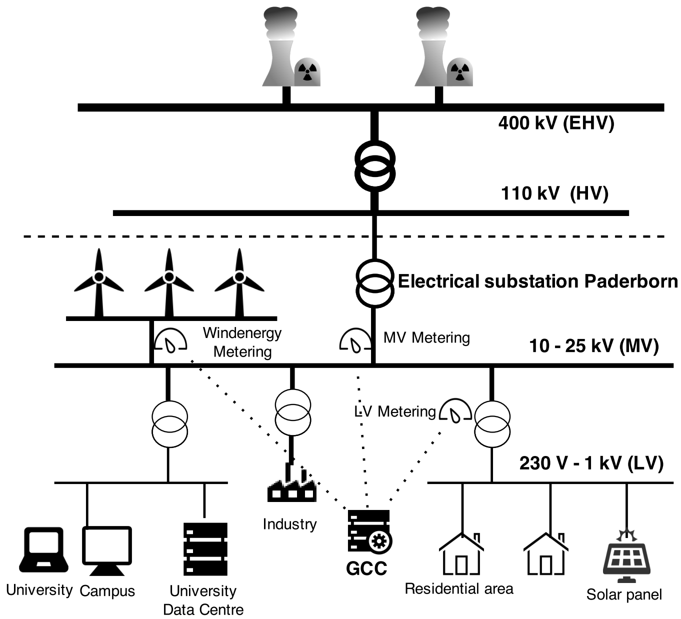


Fig. 1: Proposed smart grid architecture in Paderborn

and consumers, and not only for an on-site power plant. This section describes the scenario in more detail.

1) Smart Grid: The hypothetical Smart Grid that we consider in our experiments is located in Paderborn, Germany. Aside from the data centre, the local electrical grid supplies companies of different size and residential areas. As depicted in Figure 1, the suppliers in this medium voltage (MV) grid are wind farms and photovoltaic collectors on rooftops. The fossil-fuelled power plants are located in the extra high voltage (EHV) grid outside of the MV grid. Although the Smart Grid is not yet in place, the necessary values are made available by the local grid provider *Westfalen Weser Energy*[8] (WWE). Besides the data centre's usage, data from additional metering points are supplied which are used to train the linear models for energy prediction (cf. Section II-A). These meterings are the energy flow between MV grid and high voltage (HV) grid, the energy generated by the wind farms and the contribution of the solar panels. However, since the panels are located in residential areas, these last values are actually the difference between production and consumption in the respective low voltage (LV) grids.

2) Weather data for energy prediction: The green schedulers include planning algorithms that predict the local energy production in the near future. Apart from the energy values, these schedulers require current weather readings and forecasts. The former are taken from the closest weather station of the *German Weather Service*[9] (DWD) in Bad Lippspringe, the latter from the *European Weather Consult*[10] (EWC). Instead of predicting the energy oneself, one could also purchase energy forecasts, but these forecasts are usually only offered to energy companies, cover wider areas and can be very expensive. The energy prediction models will be described in Section II-A.

[8] http://ww-energie.com
[9] http://www.dwd.de
[10] http://www.weather-consult.com

3) Workload: The share of energy consumed from the HV grid can be reduced because of two reasons: First, computing clusters or clouds are usually not fully utilised so there is the possibility of running jobs at more favourable times. Second, we are interested in data centres with a large share of interruptible and migratable computing jobs, usually so-called batch jobs that can be stopped and restarted as well as replaced essentially at any time. In our experiments, a large data centre is simulated using freely available traces[11].

4) Objective target: The goal is to increase the share of locally produced green energy while keeping the performance degradation low and without compromising the average throughput of the data centre. By delaying and interrupting jobs, however, it is obvious that the quality of service will degrade. We use a natural quality measure for batch jobs, namely the *average turnaround time (TAT)*. The turnaround time of a job is the time it stays in the system, i.e. the time from its arrival at the queue to its completion.

Besides the share of renewable energy we will also assess the schedules by calculating the energy costs. Since there is no suitable price model in Germany yet, we use a hypothetical one. Assuming in future the energy price will be dynamic and highly dependent on where the energy is produced, this model sets the price according to the current surplus. In Section III-A we will describe it in more detail.

II. Green Control Centre

The *Green Control Centre (GCC)* is our implementation of the energy-efficient cloud environment. As depicted in Figure 2, it consists of the *Energy Prediction Component* and the *Scheduler Component* that will be described in the following subsections.

The Green Control Centre was embedded in an *OpenStack*[12] cloud environment, for which reason Figure 2 displays a few OpenStack components. Nevertheless, the concept is generally valid for any data centre running computationally intensive jobs.

A. Energy Prediction Component

The *Energy Prediction Component* predicts the future availability of renewable energy. Its inputs are the current energy and weather readings as well as the weather forecast. The subcomponents *Wind Model* and *Photovoltaic Model* use these inputs to compute the energy forecast for the wind farms and the low voltage grids (with their solar collectors), respectively. The *Grid Model* combines the output of these two subcomponents with a consumption estimate by the *Consumer Model* to predict the energy surplus or shortage in the Paderborn grid. This difference between production and consumption is the value that determines the energy price in our scenario. It must be evened out by either receiving energy from or providing energy to the high voltage grid. In the worst case, the energy suppliers have to be turned off while the grid provider still has to pay for them.

[11] http://www.cs.huji.ac.il/labs/parallel/workload
[12] http://www.openstack.org/

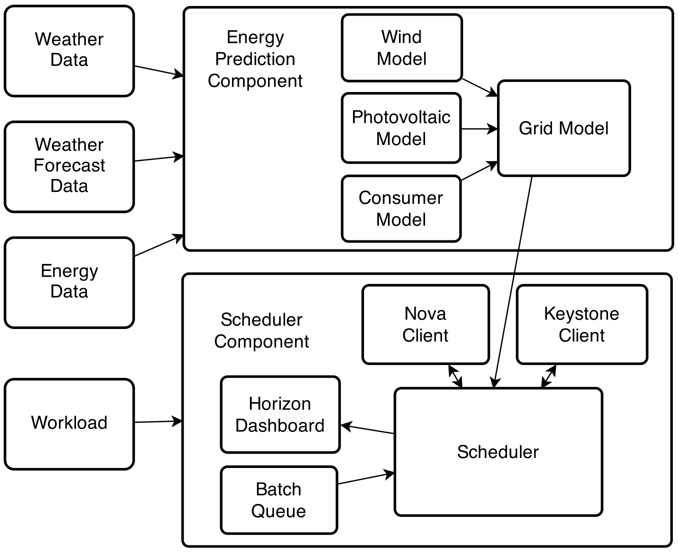


Fig. 2: Green Control Centre architecture diagram

TABLE I: Pearson correlation of weather measurements and wind energy production

Weather attribute	Correlation		
	Wind farm 1	Wind farm 2	Wind farm 3
Wind speed	0.678	0.622	0.591
Wind direction X-Axis	0.093	0.137	0.097
Wind direction Y-Axis	0.305	0.151	0.056
Sunshine	-0.172	-0.133	-0.108
Temperature	-0.181	-0.164	-0.140
Atmospheric pressure	-0.229	-0.249	-0.234
Rain	0.118	0.083	0.093

TABLE II: Pearson correlation of the weather attributes with the low voltage grid energy exchange

Weather attribute	Correlation
Wind speed	0.18967
Wind direction X-Axis	0.02072
Wind direction Y-Axis	0.05713
Cloud coverage	0.67093
Temperature	0.44410
Atmospheric pressure	0.01274
Rain	-0.07279

1) Wind Energy Prediction: To build a model that predicts the wind energy based on weather forecasts, it is necessary to determine the relevant weather attributes. This is done by computing the Pearson correlation of individual attributes with the generated wind energy. The results are displayed in Table I. The selected attributes are *wind speed*, *wind direction*, *temperature* and *atmospheric pressure*. Unsurprisingly, the wind speed shows the strongest correlation. Yet, although the value of 0.678 indicates a *high correlation*, it should be even higher. The reason for the relatively low values is that the quality of the weather data is affected by the distance between the weather station and the wind park (17 km) and by the fact that European weather services measure the wind speed at a height of 10 metres while the hubs of the wind turbines are between 36 and 62 metres high. We can quantify the error because we also have the wind speed readings the turbines take at hub height for operational purposes. Their correlation coefficients are around 0.82. The discrepancy in the coefficients suggests a significant difference between the weather readings and the actual weather at the wind farms.

Our analysis of the wind speed forecast data shows that the expected error grows with the wind speed. Although the overall expected error of 1.0 ms^{-1} is quite acceptable, it becomes large for high wind speeds, for instance 5.3 ms^{-1} for a wind speed of 10.0 ms^{-1}. Translated into wind energy, the values measured range between 0 and 31 MW. The average error of our wind energy prediction remains high due to its immediate dependence on the weather forecasts. It amounts to 3.0 MW.

For the prediction of wind energy we applied linear regression. The linear model was trained with the chosen weather attributes taking the readings of exactly one year. We investigated whether more advanced machine learning techniques would improve the forecast quality, but did not see a significant change. One approach was the extension of the linear model by clustering the weather data before applying linear regression. Another approach tested was the popular power curve model (e.g. [14], [10]) which directly maps the wind speed to the generated energy. Based on the weather readings, we derived power curves for the wind farms and used them for the prediction, but we could not see any improvement. We suppose that the prediction could only be improved if the weather measurements were better. However, the evaluation in Section III will show that the quality is already sufficient for our purposes.

2) Photovoltaic Energy Prediction: The photovoltaic energy prediction is more complicated in our scenario because we do not have exact measurements for the installed panels. Instead, we have the energy exchange of one of the LV grids with Paderborn's MV grid. The LV grid includes not only the production of the panels, but also the consumption of the respective residential area. Additionally, since the installed photovoltaic power of the whole grid (3.5 MW peak) is about ten times larger than the installed power for the monitored grid (330 kW peak), we have to extrapolate the measurements of the LV grid accordingly.

We determine the significant weather attributes using correlation and apply linear regression to estimate the energy. Table II shows that the two relevant attributes are cloud coverage and temperature. Yet in this case, further attributes are reasonable: the irradiation angle of the sun, day of the week and time. The latter two are required so that the system can learn the behaviour of the consumers in the LV grid which is assumed to be day-of-the-week and time-of-the-day dependent. The irradiation angle and the cloud coverage determine the amount of solar energy reaching the Earth's surface. For the calculation of the irradiation angle, we use a tool called *Pysolar*[13].

The analysis of the cloud coverage forecast quality reveals an average error of 28.6%. It is respectively higher (46.3%) or lower (21.2%) if a cloudless or overcast sky is predicted. The measured energy values of the low voltage grid cover a range of 322 kW surplus and 122 kW demand where the periods of energy demand are almost always at night. At these times, the values show little variance in the consumer behaviour and can be predicted with an average error of less than 20 kW. Surplus

[13] http://pysolar.org

situations, on the other hand, have an average error of 50 to 85 kW.

3) Energy Consumption Prediction: For reasons of privacy, individual energy consumption readings necessary for pattern matching are not available. Yet, even though the precision of the prediction is limited, one can still get a fair estimate by training a linear model using time data like the day of the week, date and time of the day. This allows to roughly predict the general consumption during the day, week or season. In a future Smart Grid, the grid provider could publish anonymized or generalised usage statistics allowing a more elaborate prediction of the consumption.

4) Prediction of the Grid Exchange: The grid exchange estimated by the *Grid Model* defines the amount of energy that has to flow to balance the surplus or shortage in the local grid. With respect to the bidirectional nature of the energy exchange, the *Grid Model* is similar to the *Photovoltaic Model*.

The measured values range between 30.6 MW surplus and 21.8 MW demand. The average error of the prediction grows with the green energy surplus and is between 2.5 and 10 MW. We believe that the rather poor precision could be improved by using better weather data, for instance provided by on-site weather stations.

In our scenario we use the grid exchange to derive an energy price that would be provided by a Smart Meter. Since the data centre is integrated into the Smart Grid, we conclude that it has access to the Smart Meter and that the energy price is available at runtime so that it can be used to correct the forecast.

B. Scheduler

In the reference implementation (Figure 2), the *Scheduler Component* is embedded into an OpenStack environment where it functions as an energy-aware batch-processing system. The main components are the *Batch Queue* managing the incoming batch jobs and the *Scheduler* generating a schedule and running the jobs accordingly. Three further components are needed to integrate the service into OpenStack: The *Nova Client* is used to keep track of the cloud infrastructure, to monitor the state of the cloud, i.e. the available resources, and to start and stop virtual machines. *Keystone*, accessed by the *Keystone Client*, is OpenStack's identity service and handles the user and project management. *Horizon* finally is a web-based graphical user interface into which we integrated a read-only view of the batch processing system.

The system creates a home directory and an input and output subdirectory for every cloud user. New jobs and any input parameters are placed in the input directory, the results in the output directory. The home directory is mounted into the job's virtual machine by a *cloud-init* script. If the jobs come with deadlines, they are sorted by them, otherwise they are simply processed in first-in first-out (FIFO) order.

In the following, we describe the different implementations of the *Scheduler* subcomponent, namely the *FIFO scheduler*, the *MATH scheduler*, the *green scheduler*, the *enhanced green scheduler* and an *optimal scheduler*. They differ in the way in which they choose the times at which the jobs are run. Once this is decided, the placement of the virtual machines is done in a best-fit manner. The scheduler views the available resources

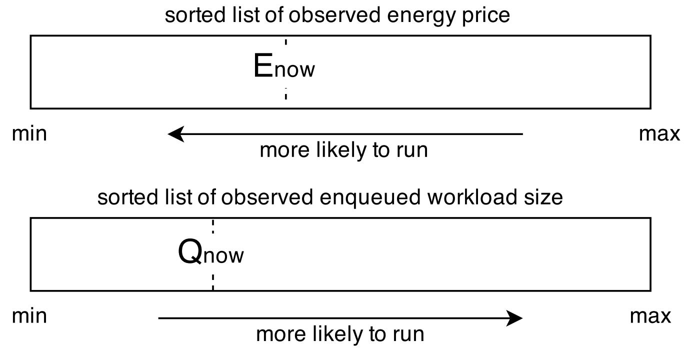


Fig. 3: Graphical representation of the mathematical model

of all hosts and assigns the virtual machine accordingly. In doing so, it bypasses the OpenStack scheduling mechanism and directly accesses the administrator interface, transparent to the cloud infrastructure.

1) FIFO Scheduler: The FIFO scheduler is a CPU-optimised scheduler ignorant of the Smart Grid that does not try to utilise green or locally produced energy. It processes the jobs in the order in which they arrive and starts the next job as soon as a job is finished. This is a plausible strategy in batch processing systems that consider neither deadlines nor priorities.

In our system evaluation this strategy is used for comparison and marks the baseline energy consumption that the other strategies are set in relation to.

2) MATH Scheduler: The MATH scheduler uses the sliding window technique to decide whether new jobs shall be scheduled. For this, it needs to have access to the workload queue and to the currently given energy price as provided by a Smart Meter. As outlined in Figure 3, it keeps two lists, L_E and L_Q, of the energy prices and queue sizes of the last n hours, where the list size n is configurable. In order to decide whether to start new jobs, the scheduler sorts the lists in ascending order. Let E_{now} and Q_{now} be the indices pointing at the current entries in the sorted lists. Then new jobs are only started if $E_{now} \leq Q_{now}$.

The indices essentially express the urge to process workload: the lower E_{now}, the lower the energy price compared to recent situations and, thus, the higher the incentive. Likewise, the higher Q_{now}, the busier the queue and, therefore, the greater the need to reduce it. Figure 3 illustrates this relationship. If the current queue is short, the scheduler will not schedule new jobs unless the energy price is also relatively low. If the queue is long, even unfavourable energy costs will be accepted.

To reduce the average turnaround time jobs that can be finished within the next time slot are always processed. During the evaluation, different sliding window sizes (i.e. list sizes) are used: from one week (7 days) to half a year (182 days).

3) Green Scheduler: As described in Section II-A, the green scheduler uses an energy forecast to identify the best time slots. Weather forecasts, and subsequently also energy forecasts, are provided for the next 48 hours, and the forecast period is divided into one hour slots. The green scheduler sorts these slots in descending order, meaning the first slot in the sorted list is the most energy-efficient. For each of the time slots the jobs from the queue are placed on the cloud hosts and the requested runtime is decreased by one hour. The algorithm ends when all jobs have no unscheduled runtime left or when all time slots are fully utilised. During longer periods of low renewable energy production all of these slots might be undesirable. Therefore, we added a threshold X to the queue so that the scheduling algorithm stops if no renewable energy is available and the sum of the remaining jobs' runtime is less than X data centre hours. Given the currently available cloud resources this results in postponing the processing of jobs with a total runtime of X hours. Since new jobs will arrive during the 48 hours scheduling window, the runtime of these new jobs is estimated and added to the remaining runtime. Since we assume access to a Smart Meter we can correct the energy forecast for the next scheduling window. Instead of relying on the previously generated energy forecast for the next 60 minutes, we replace the forecast with the most recent Smart Meter measurement. This reduces the damage induced by the forecast quality issue.

A few long running jobs with a low degree of parallelism can sabotage the green scheduler. The sum of their remaining runtime can exceed the queue limit and force the scheduler to process jobs during unfavourable time slots. Since these jobs will not saturate the cluster, additional shorter jobs are processed as well. This might lead to the situation that not enough jobs are left to saturate the cluster at more favourable situations, contradicting the incentive of the scheduler. Therefore, instead of FIFO ordering, the jobs are processed in descending order regarding their remaining runtime. The turnaround time improvement mentioned in II-B2 is applied as well.

Traces including deadlines need to abide to them and if necessary overrule the slot utilisation given by the scheduler accordingly.

4) Enhanced Green Scheduler (ENHG): In Section II-A, it turned out that the energy forecast is fairly inaccurate, which limits the efficiency of the green scheduler. For this reason, it is interesting to determine how well the green scheduler would work if it had access to a perfect energy forecast. We call this scheduler the *enhanced green scheduler*.

5) Optimal Scheduler (OPT): While the enhanced green scheduler uses perfect energy forecasts, it is not optimal because it is limited by the 48 hour forecast window and by the fact that it has no knowledge about the future workload situation. The *optimal scheduler*, on the other hand, has perfect knowledge of the future workload and energy and used it to create an approximately optimal schedule. It is only an approximation because the problem of scheduling jobs of different runtimes and resource requirements is known to be NP-hard.

The optimal scheduler uses the following greedy approach: It keeps an array J of jobs sorted by their timestamps in descending order and an array S of energy slots sorted by their energy price in ascending order. For every slots $s_i \in S$ it iterates over all jobs $j_k \in J$. j_k is assigned to slot s_i if $remainingRuntime(j_k) > 0$, $timestamp(j_k) \leq timestamp(s_i)$ and $cores(j_k) \leq unusedCores(s_i)$.

III. Evaluation

For the evaluation, we compare the scheduling strategies described in the previous section. They are executed in a simulated environment using real weather and energy data and workload traces. To quantify the quality of the strategies, we use three measures: the first is the share of green energy, the second is the energy costs in our hypothetical price model and the third is the average turnaround time of the jobs. Before summarising the execution and the outcomes of the experiments, we start by describing and motivating the price model.

TABLE III: Percentiles of the energy price in ct/kWh

P(05)	P(10)	P(25)	P(50)	P(75)	P(90)	P(99)	mean
10.0	11.35	14.8	17.25	18.9	20.0	20.0	16.49

A. Price Model

As pointed out before, the grid provider has an incentive to reduce the local surplus and sell the energy where it is produced. For this reason, we assume that the customer price will drop once the locally generated energy exceeds the customer demand and model the price as a function $p(x)$ of the grid exchange x between the Paderborn grid and the HV grid (see Section II-A4):

$$p(x) = R + y(x).$$

R is the reference price of 15 ct/kWh and $y(x)$ the dynamic portion defined as follows:

$$y(x) = \begin{cases} -V, & \text{if } x \leq -10 \\ V, & \text{if } x \geq 10 \\ \frac{x}{10} \cdot V, & \text{otherwise} \end{cases}$$

where the variance V is set to $V = \frac{1}{3} \cdot R = 5$ ct/kWh.

According to *Eurostat*[14], the 15 ct/kWh baseline is the average price for small industry consumers with an annual energy consumption of 20 to 500 MWh in Germany. This price includes all non-recoverable taxes and levies.

Since the prices of the EPEX[15] intra day spot market usually vary by more than 50 €/MWh, we consider the variance of 5 ct/kWh a conservative estimate. Increasing or decreasing the variance would have an impact on the achievable savings, but, regardless of the exact value, the cost analysis would show the same trend. Table III displays the percentiles of the resulting energy price for the simulation period.

B. Simulation and Results

To simulate a realistic data centre, we use workload logs from the *Potsdam Institute for Climate Impact Research*[16] (PIK). Each simulation covers a full year and is executed in one hour steps. We assume that the batch jobs executed are CPU-intensive and consume the full *thermal design power* (TDP) of the assigned cores.

The *PIK cluster* is a *320 node IBM iDataPlex Cluster* and has 2560 cores. The *Intel Xeon Harpertown CPU* has a TDP between 50 and 150 watts. We simulated the cluster assuming a TDP of 120 watts per CPU and 30 watts per core. Based on measurements in our own data centre, we set a server's

[14] http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=nrg_pc_205&lang=en
[15] http://www.epexspot.com
[16] http://www.cs.huji.ac.il/labs/parallel/workload/l_pik_iplex

TABLE IV: PIK simulation results using different scheduling strategies and configurations

Strategy	PIK I (11/12)				PIK II (10/11)				PIK III (09/10)			
	Costs in EUR	Cost ratio	avg. TAT in hours	Renewable energy share	Costs in EUR	Cost ratio	avg. TAT in hours	Renewable energy share	Costs in EUR	Cost ratio	avg. TAT in hours	Renewable energy share
FIFO	49012	1.000	11.58	26.31	36867	1.000	9.63	24.14	42881	1.000	8.09	28.68
MATH(7)	46289	0.944	55.04	36.94	35069	0.951	39.55	32.68	40440	0.943	34.67	40.25
MATH(28)	46065	0.940	34.61	37.58	34540	0.937	27.68	36.71	40214	0.938	22.88	40.01
MATH(91)	45899	0.936	32.10	38.66	34233	0.929	23.50	37.49	39685	0.925	22.35	43.48
MATH(182)	45860	0.936	32.07	38.40	34082	0.924	24.61	38.29	39585	0.923	24.46	44.01
GREEN(3)	44808	0.914	34.11	43.74	33936	0.920	18.99	40.54	39101	0.912	26.62	46.73
GREEN(6)	44574	0.909	37.01	44.99	33729	0.915	20.07	42.18	38876	0.907	29.13	48.32
GREEN(12)	44129	0.900	45.20	47.62	33343	0.904	22.29	45.20	38440	0.896	34.38	51.76
GREEN(24)	43305	0.884	62.69	52.87	32665	0.886	28.28	50.97	37742	0.880	41.77	57.48
GREEN(48)	42282	0.863	123.36	60.46	31814	0.863	42.56	59.46	36672	0.855	51.14	66.66
ENHG(3)	45003	0.918	28.82	43.50	33875	0.919	18.66	41.99	39293	0.916	18.95	46.75
ENHG(6)	44716	0.912	32.46	45.11	33700	0.914	19.46	43.27	39027	0.910	20.53	48.56
ENHG(12)	44177	0.901	40.44	48.14	33332	0.904	21.65	46.23	38488	0.898	24.93	52.26
ENHG(24)	43229	0.882	60.85	53.90	32660	0.886	28.40	51.81	37637	0.878	32.11	58.61
ENHG(48)	42273	0.863	138.71	61.00	31728	0.861	43.66	60.19	36524	0.852	46.51	67.74
OPT	40749	0.831	389.39	65.27	30002	0.814	478.37	63.26	35798	0.835	228.22	69.22

idle power consumption to 80 watts and the power-off / *Wake on LAN* consumption to 7 watts. The workload logs contain 742965 jobs using up to 1024 cores and having individual runtimes between a few seconds and 30 days. We extracted three traces that we name *PIK I*, *PIK II* and *PIK III*. They cover the period between the 1st June and the 31st May of the years 11/12, 10/11 and 09/10, respectively.

The scheduling strategies of Section II-B are simulated with different configurations. The MATH scheduler is run with sliding window sizes of 7, 28, 91 or 182 days, the (enhanced) green scheduler with queue sizes of 3, 6, 12, 24 or 48 hours. Since the improved efficiency of these schedulers is achieved by delaying processing of jobs during unfavourable situations, at the end of the simulation unfinished jobs remain in the queues. The difference in the computation time of these schedulers compared to the FIFO scheduler is evened out at the worst case costs of 20 ct/kWh.

Table IV shows the results of the three *PIK* simulations for the proposed scheduling strategies. The parameter of the MATH scheduler defines the size of the sliding window (in days) and the parameter for the (enhanced) green scheduler the queue limit. The first column of each trace lists the total processing costs. For easier comparison, the second column normalises these values with FIFO results. Hence, it shows the cost reduction of the respective scheduling strategy. Delaying the execution of workloads inevitably increases the turnaround time. The fourth column finally states the share of renewable energy in percent. Figure 4 is a graphical representation of the results. The bar charts display the share of renewable energy achieved by each scheduler and the scatter plots set the cost efficiency in relation to the turnaround time.

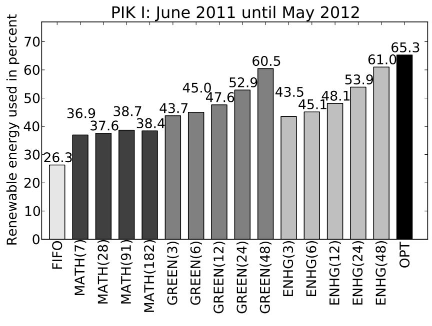
In the following we can restrict our analysis to PIK I because the values for PIK II and III are fairly similar. During the PIK I simulation, the FIFO scheduler needs 7.034 million core hours and energy for €49012. The average turnaround time of the finished jobs is 11.58 hours, and 26.31% of the consumed energy is locally generated.

For the MATH schedulers, we observe that the use of green energy increases slowly with the window size, while the average turnaround time differs quite significantly and ranges from 32.07 to 55.04 hours. The schedulers with longer sliding windows perform better because schedulers with short windows do not necessarily recognize good energy values during long periods of renewable energy surplus. This happens when most of the window values are good and relatively better than the current one.

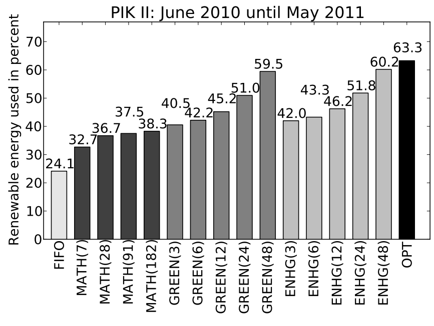
The green scheduler improves the usage of renewable energy and cost savings further. Depending on the queue size, the share of renewable energy can be increased up to 60.46% which results in cost savings of 13.7%. However, the higher the share of renewable energy or savings, the longer the turnaround times. In fact, Figure 4d suggests that the ratio of cost savings to turnaround time decreases exponentially with the queue size. Compared to the MATH scheduler, small queue sizes of 3, 6 and 12 hours result in a better efficiency while the turnaround time increases only slightly. Figure 4a shows that the enhanced green scheduler is not necessarily better than the green scheduler for the same queue size. This does not contradict the optimality claim of the scheduler because here *optimal* only refers to the 48 hour energy forecast. The enhanced green scheduler does not have a better workload forecast so that postponed scheduling decisions can, for example, result in longer queues that have to be processed at unfavourable times.

The optimal scheduler achieves a maximum utilization of renewable energy of 65.3% and is therefore only 4.3% better than ENHG(48) and 4.8% better than GREEN(48). Nevertheless, shifting the huge amount of workload necessary to achieve these results inevitably inflates the average turnaround time.

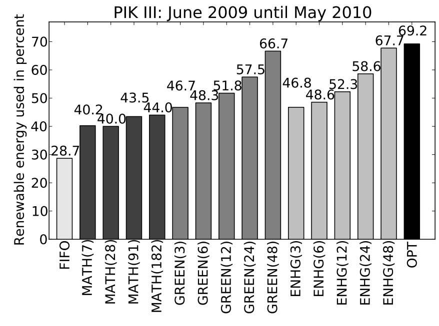
Considering the large average error of the energy forecast, the green schedulers' results are surprisingly close to the enhanced green scheduler results. This indicates that a reasonable estimate of the expected energy situation is sufficient to make an informed decision, as long as a Smart Meter can be queried to correct forecast errors at runtime. This is an important outcome and allows the conclusion that even a simple forecast with an error margin of up to 30% can yield good results. Whether a better forecast is worth the higher operational overhead and potentially expensive input data depends on the size of the data centre and, neglecting the environmental aspect, the possible cost savings.



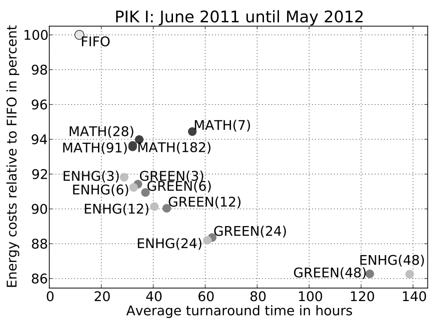
(a) Share of renewable energy (PIK I)



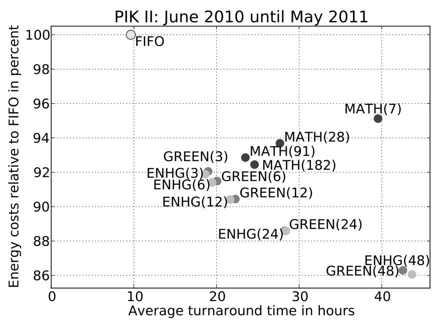
(b) Share of renewable energy (PIK II)



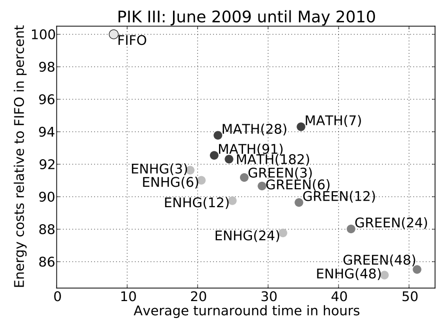
(c) Share of renewable energy (PIK III)



(d) Costs versus turnaround time (PIK I)



(e) Costs versus turnaround time (PIK II)



(f) Costs versus turnaround time (PIK III)

Fig. 4: Comparison of the PIK simulation results with different configurations

IV. Conclusion

This paper has presented and evaluated two Smart Grid-aware scheduling strategies. We assessed the increase in the utilisation of locally produced renewable energy and the monetary benefit based on a hypothetical, yet conservatively estimated price model. We also quantified the performance penalty in terms of a higher turnaround time. The assumed Smart Grid is based on real measurements of Paderborn's energy grid, and, to the best of our knowledge, this paper is the first in making predictions for a Smart Grid itself while most of the previous work predicted the generation of on-site power plants using the grid merely as a backup energy source.

For the more simple scheduler based on a mathematical model, recording Smart Meter values is already sufficient to increase the renewable energy consumption by up to 58 % compared to the reference FIFO scheduler. The *green scheduler*, on the other hand, also requires weather data to predict the future energy surplus in the grid and to schedule the incoming jobs accordingly. While it is possible to reach a nearly optimal share of green energy, the downside is a considerable rise in the turnaround time. Our future work will aim to reduce the average turnaround time in order to reach an acceptable trade-off between green energy utilisation and customer satisfaction.

V. Acknowledgment

We would like to thank *Weser Westfalen Energy* (WWE) and especially Uwe Granzow for providing and explaining the energy data to us. The WWE was one of the partners in the BMWi project "GreenPAD".

References

[1] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper. Free lunch: exploiting renewable energy for computing. In *HotOS'13 Proceedings of the 13th USENIX conference on Hot topics in Operating Systems*, pages 17–17, 2011.

[2] B. Aksanli, J. Venkatesh, L. Zhang, and T. Rosing. Utilizing green energy prediction to schedule mixed batch and service jobs in data centers. *ACM SIGOPS Operating Systems Review*, 45(3):53–57, 2012.

[3] R. Baïle, J. F. Muzy, and P. Poggi. Short-term forecasting of surface layer wind speed using a continuous random cascade model. *Wind Energy*, 14(6):719–734, 2011.

[4] A. Beloglazov, J. Abawajy, and R. Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5):755–768, May 2012.

[5] J. L. Berral, Í. Goiri, T. D. Nguyen, R. Gavalda, J. Torres, and R. Bianchini. Building green cloud services at low cost. In *IEEE 34th International Conference on Distributed Computing Systems (ICDCS)*, pages 449–460. IEEE, 2014.

[6] B. G. Brown, R. W. Katz, and A. H. Murphy. Time series models to simulate and forecast wind speed and wind power. *Journal of Climate and Applied Metereology*, 23, 1984.

[7] M. Brown and J. Renau. Rerack: Power simulation for data centers with renewable energy generation. *ACM SIGMETRICS Performance Evaluation Review*, 39(3):77–81, 2011.

[8] F. Cassola and M. Burlando. Wind speed and wind energy forecast through kalman filtering of numerical weather prediction model output. *Applied Energy*, 99:154–166, 2012.

[9] W. Deng, F. Liu, H. Jin, B. Li, and D. Li. Harnessing renewable energy in cloud datacenters: opportunities and challenges. *IEEE Network*, pages 48–55, 2014.

[10] G. Giebel, R. Brownsword, G. Kariniotakis, M. Denhard, and C. Draxl. *The State-Of-The-Art in Short-Term Prediction of Wind Power: A*

Literature Overview, 2nd edition. ANEMOS.plus, 2011. Project funded by the European Commission under the 6th Framework Program, Priority 6.1: Sustainable Energy Systems.

[11] Í. Goiri, R. Beauchea, K. Le, T. D. Nguyen, M. E. Haque, J. Guitart, J. Torres, and R. Bianchini. Greenslot: scheduling energy consumption in green datacenters. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, page 20. ACM, 2011.

[12] Í. Goiri, K. Le, T. D. Nguyen, J. Guitart, J. Torres, and R. Bianchini. Greenhadoop: leveraging green energy in data-processing frameworks. In *Proceedings of the 7th ACM european conference on Computer Systems*, pages 57–70. ACM, 2012.

[13] A. Kusiak, H. Zheng, and Z. Song. Short-term prediction of wind farm power: a data mining approach. *Energy Conversion, IEEE Transactions on*, 24(1):125–136, 2009.

[14] M. Lange. *Analysis of the uncertainty of wind power predictions*. PhD thesis, Universität Oldenburg, 2003.

[15] C. Li, A. Qouneh, and T. Li. iswitch: coordinating and optimizing renewable energy powered server clusters. In *Computer Architecture (ISCA), 2012 39th Annual International Symposium on*, pages 512–523. IEEE, 2012.

[16] C. Li, W. Zhang, C.-B. Cho, and T. Li. Solarcore: Solar energy driven multi-core architecture power management. In *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, pages 205–216. IEEE, 2011.

[17] Y. Li, D. Chiu, C. Liu, L. T. Phan, T. Gill, S. Aggarwal, Z. Zhang, B. T. Loo, D. Maier, and B. McManus. Towards dynamic pricing-based collaborative optimizations for green data centers. In *Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on*, pages 272–278. IEEE, 2013.

[18] M. Mohandes, T. Halawani, S. Rehman, and A. A. Hussain. Support vector machines for wind speed prediction. *Renewable Energy*, 29(6):939–947, 2004.

[19] O. Niehörster, A. Keller, and A. Brinkmann. An energy-aware saas stack. In *IEEE 19th International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 450–453, July 2011.

[20] No author given. Die Energie der Zukunft. Technical report, Bundesministerium für Wirtschaft und Energie (BMWi), December 2014.

[21] C. Ren, D. Wang, B. Urgaonkar, and A. Sivasubramaniam. Carbon-aware energy capacity planning for datacenters. In *Proceedings of the 2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, MASCOTS '12, pages 391–400, Washington, DC, USA, 2012. IEEE Computer Society.

[22] N. Sharma, S. Barker, D. Irwin, and P. Shenoy. Blink: managing server clusters on intermittent power. In *ACM SIGPLAN Notices*, volume 47, pages 185–198. ACM, 2011.

[23] Y. Zhang, Y. Wang, and X. Wang. Greenware: Greening cloud-scale data centers to maximize the use of renewable energy. In F. Kon and A.-M. Kermarrec, editors, *Middleware 2011*, volume 7049 of *Lecture Notes in Computer Science*, pages 143–164. Springer Berlin Heidelberg, 2011.