CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2017; **00**:1–15
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe

A Data-Driven Approach to Modeling Power Consumption for a Hybrid Supercomputer

Alina Sîrbu[1,3*], Ozalp Babaoglu[2]

1. Department of Computer Science, University of Pisa, Italy.
2. Department of Computer Science and Engineering, University of Bologna, Italy.
3. Science Division, New York University Abu Dhabi, United Arab Emirates.

SUMMARY

Power consumption of current High Performance Computing systems has to be reduced by at least one order of magnitude before they can be scaled up towards ExaFLOP performance. While we can expect novel hardware technologies and architectures to contribute towards this goal, significant advances have to come also from software technologies such as proactive and power-aware scheduling, resource allocation and fault-tolerant computing. Development of these software technologies in turn relies heavily on our ability to model and accurately predict power consumption in large computing systems.
In this paper we present a data-driven model of power consumption for a hybrid supercomputer (which held the top spot in the Green500 ranking in June 2013) that combines CPU, GPU and MIC technologies to achieve high levels of energy efficiency. Our model takes as input workload characteristics — the number and location of resources that are used by each job at a certain time — and calculates a predicted power consumption at the system level. The model is application-code-agnostic and is based solely on a data-driven predictive approach, where log data describing the past jobs in the system are employed to estimate future power consumption. For this, *three different model components* are developed and integrated. The first employs *support vector regression* to predict power usage for jobs before these are started. The second uses a simple *heuristic* to predict the length of jobs, again before they start. The two predictions are then combined to estimate power consumption due to the job at all computational elements in the system.The third component is a *linear model* that takes as input the power consumption at the computing units, and predicts system-wide power consumption. Our method achieves highly-accurate predictions starting solely from workload information and user histories. The model can be applied to power-aware scheduling and power capping: alternative workload dispatching configurations can be evaluated from a power perspective and more efficient ones can be selected. The methodology outlined here can be easily adapted to other HPC systems where the same types of log data are available.

Received . . .

KEY WORDS: power modeling; job power prediction; job length prediction; hybrid HPC system; workload; energy efficiency.

1. INTRODUCTION

Power consumption of computational systems has become a major concern in the computing community. Today, it is not uncommon for a large data center, such as those hosting HPC systems, to consume as much power as a mid-size city, with the obvious economic and environmental consequences. Furthermore, large power needs have negative implications for the systems themselves, for example by requiring complex and expensive cooling structures and by limiting their scalability. Accurate models of power consumption in large computing systems will

*Correspondence to: alina.sirbu@unipi.it

be extremely important for optimizing their energy usage. Models allow for prediction of system behavior under various scenarios, enabling advanced scheduling and fault tolerance techniques that are essential for making Exascale computing sustainable.

In this paper we model and predict system-level power consumption starting from workload measures for Eurora [1], an experimental hybrid High Performance Computing (HPC) system with CPUs, GPUs and MICs. Our predictive model consists of *three components*. First, power consumption of jobs is predicted from workload data using a *Support Vector Regression (SVR) approach* [2]. Second, we introduce a *simple heuristic* that enables data-driven prediction of job length, which allows us to estimate which jobs will run in the system at a future time. The two predictions are then combined to estimate the power used by computing units. Third, we develop a relation between power used by computing units and the total system power, including networking, IO system and other elements, using a *linear model*.

This work makes various contributions to modeling power consumption for HPC systems. First, the relation between power of computing units and system power is investigated, and a clear linear dependency between the two is observed, in agreement with other studies. We take this result one step further by building a complete power model for the entire system. Our second contribution is a model, constructed out of three components, that is capable of predicting system-level power starting from workload data. Our approach does not require knowledge of application code or hardware counters for power prediction. This makes the methodology easily extendable to other systems, since only simple workload measures that are common to all HPC systems are used. Our final contribution is an investigation of the possible applications of our model to power-aware scheduling.

The rest of the paper is organized as follows. We first discuss the data and our prediction approach in Sections 2 and 3. The transition from workload to power consumption of computing units using job power and length prediction is investigated in Section 4.1, while the integration of all three model components is described in Section 4.2, where results at system level are presented. Section 5 discusses potential applications of our model. State-of-the-art is surveyed in Section 6 and Section 7 summarizes and concludes the paper.

2. EURORA DATA

This work uses workload and power measurements from Eurora, a hybrid HPC system installed at CINECA, the largest data center in Italy [3]. The system consists of 64 nodes each equipped with 2 CPUs (8-core Intel Xeon E5 CPUs) and 2 accelerators. The accelerators on half of the nodes are GPUs (Nvidia Tesla Kepler), while on the other half they are MICs (Intel Xeon Phi). Thus the system is equipped, in total, with 1028 CPU cores, 64 GPUs and 64 MICs. Eurora was number one on the Green500 list in June 2013 and runs a custom monitoring framework [4] that collects system logs related to workload (hundreds of thousands of jobs from hundreds of users) and power consumption at both computing unit and at system levels. The data is stored in a MySql database, henceforth referred to as the "Eurora database". The measurements span the period from March 2014 to August 2015, resulting in over 250GB of data.

In our study, we start from workload information and build a prediction of system-level power consumption. In order to build our model components, we require several data types: workload (information on jobs submitted to the system), recorded power for computing units, power measured at system level. All these data were extracted from the Eurora database. A preprocessing stage was required to correct missing data before extracting the features useful for modeling, and to synchronize the timestamps for the various data types. This meant replacing missing power of idle computing units with a default value, and removing all data points where it was unclear whether the missing data was due to idle units. More details on the procedure can be found in previous work [5].

After data correction and synchronization, workload data consists of 57,183 jobs from 401 unique users. Job information includes user, queue, and timestamps for submission, start and finish, together with number of computing units used (CPU cores, GPUs and MICs) and their allocation in the system. These data were used to extract job length to be predicted by our heuristic, and also input

DOI: 10.1002/cpe

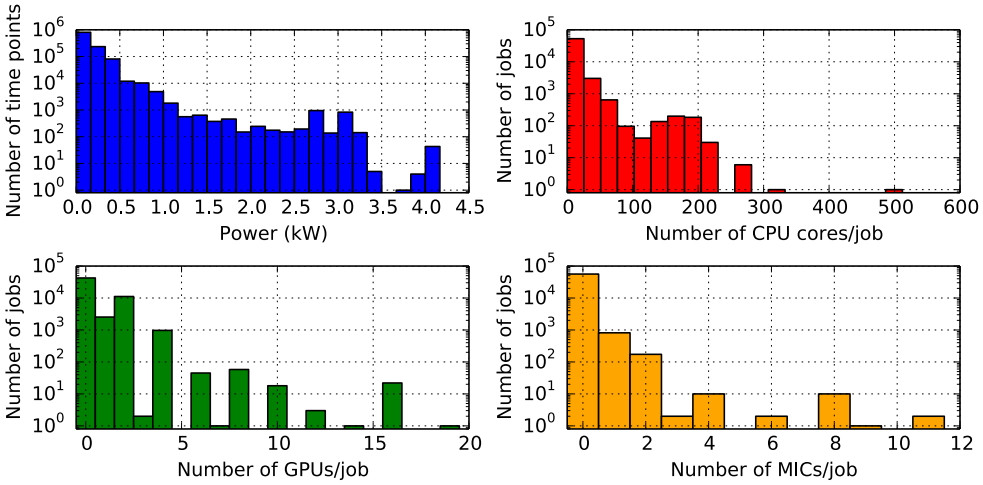

Figure 1. Workload properties for corrected data. The top-left plot shows the distribution of power consumption per job, measured at 5-minute intervals. The rest of the histograms show the distribution of resources used by each job (CPU cores, GPUs and MICs).

features for the SVR model predicting job power consumption. Figure 1 shows the distribution of the number of computing units used by each job. Many jobs use only CPUs with about 26% of jobs employing GPUs and about 2% employing MICs.

Power consumption for computing units (CPUs — including RAM power, GPUs and MICs) is available in the dataset at a 5-second resolution. Power is available at the level of CPU, GPU and MIC, and not at core level. These data were used to compute two measures. First, we computed power profiles for jobs: power at 5-minute intervals for the computing units used by that job. These profiles became the regression target of the first component of our model, SVR. Figure 1 also shows the distribution of power levels recorded at 5-minute intervals for each job. Second, we computed the total power of all active computing units. This was used as an input feature for the third model component, the linear model mapping computing to system power. Additionally, it was employed to validate the integration of the first two model components (see Section 4.1 bellow).

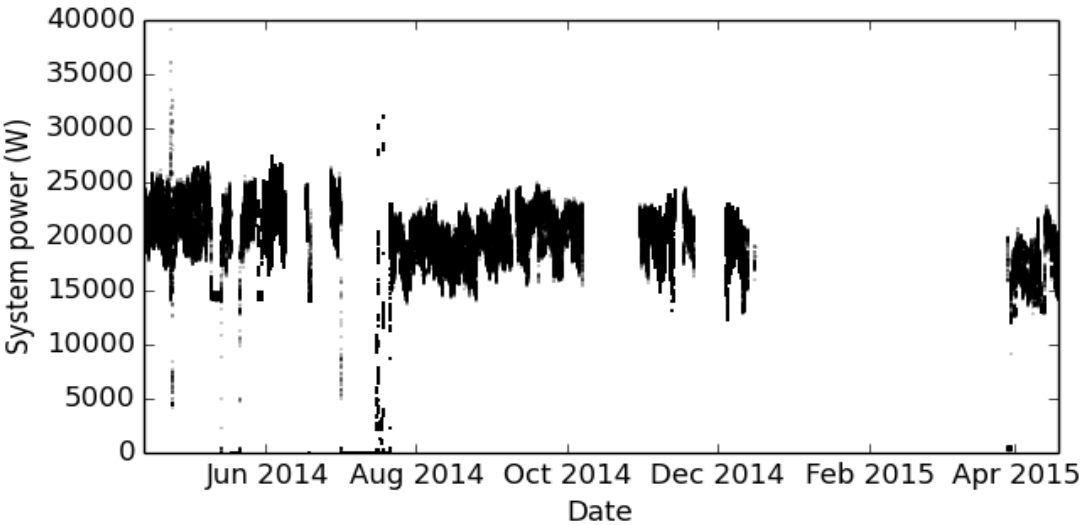

Figure 2. Availability of power measurements at system level.

DOI: 10.1002/cpe

The third piece of data required for our study is power measurements at system level, i.e., power consumed not only by the computing units but the entire rack. The Eurora database includes a table that contains measurements at the main electric panel for the system, at 5-minute intervals. These measurements were the target of our integrated model. Figure 2 shows power measurements available at this level. Several gaps that are evident in the figure are due to both system shutdowns and monitoring issues. For this reason we concentrate on the period July-November 2014 which contains enough contiguous data for training and testing our models.

3. PREDICTION APPROACH

Using the data described above, we predict power consumption at system level based only on workload measures. This involves the integration of three modeling components. Figure 3 presents graphically the approach adopted. The first component is a SVR model that is able to predict power profiles for jobs, in advance. The second component is an heuristic that enables prediction of job length. The third and last component is a linear model that maps power of the computing units to power at system level. Computing unit power is extracted by combining the first two modeling components. In the following we will describe each component individually and provide details on their integration.

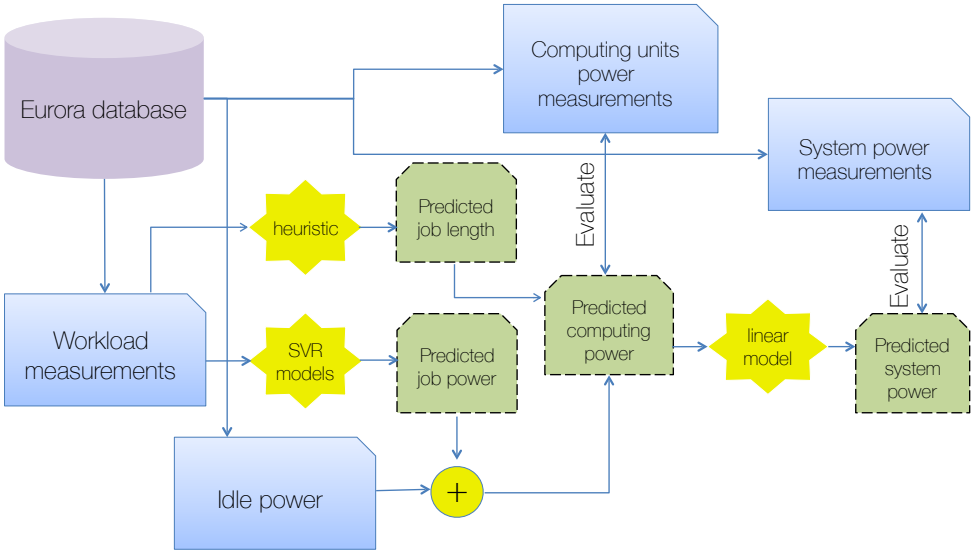

Figure 3. System-level power model. The three model components are represented by the star boxes: *SVR models* to predict job power, *heuristic* for prediction of job length and *linear model* to obtain system power. Prediction resulting from the first two components are combined to predict computing power, which is then used as input for the third component. Predictions at computing unit and system level are evaluated against real data.

3.1. Job power prediction

The first component of our model predicts *power consumption of jobs* in Eurora starting from workload measures. The analysis uses only job characteristics and collocation information, and was introduced in detail in previous work [6]. The method uses SVR to build one model per user. For each job, a wide set of regression features are employed to predict its power profile in time. A job is described by independent features, which are job name, number of CPU cores, GPUs and MICs used by the job and number of nodes allocated, but also by features that describe the workload and resource allocation globally, for example the number of cores/GPUs/MICs in use by other jobs collocated on the same nodes as the job being analyzed. This choice enables prediction of power

DOI: 10.1002/cpe

	Number of users	Average jobs per user	Average CPU cores	Average GPUs	Average MICs	Average duration (minutes)
Train data	34	12796	13.2	0.34	0.07	85.03
Test data	34	1579	12.62	0.36	0.06	95.01

Table I. Summary of data used for training and testing the SVM.

interference across jobs, so that different mapping of resources can result in differences in power levels.

The analysis is based on workload and power data extracted from the Eurora database. Specifically, we use measurements of computing unit power to calculate the exact power used by each job at 5-minute intervals (the regression target) while features are extracted from the information about resource allocation that exists in the database. For each user we first build one SVR model to predict the power consumed by jobs on each component type (CPUs, GPUs, MICs), which are then summed to compute the overall power for jobs. This is applied at 5-minute intervals to compute a power profile for each job of that user. We train each model with data up to September 2014 and then apply to jobs from the first week of October 2014. Further details of job power prediction with this approach can be found in the original paper [6].

For training, we applied the SVR method only to users with at least 1000 data points coming from at least 100 different jobs. Out of 84 users who accessed Eurora during the first week of October 2014, 34 had enough data to train the SVR model. Table I shows general statistics of the workload for these users, for the training and testing periods. For the rest of the users, who had less data available, we used an *Enhanced Average Model* (EAM), also introduced in [6]. It is important to note that the data related to the activity of the remaining users consist of much fewer data points compared to those for the 34 regular users.

To apply the EAM, we computed for each user u an average power per unit type (CPU core: $\bar{P}^u_{CPU}$, GPU: $\bar{P}^u_{GPU}$, MIC: $\bar{P}^u_{MIC}$) from the limited amount of existing training data. For each job j belonging to the user, we count the number of units used by the job, denoted as n^j_{CPU}, n^j_{GPU} and n^j_{MIC}. The predicted job power can then be computed as:

$$P^*_j = n^j_{CPU} \times \bar{P}^u_{CPU} + n^j_{GPU} \times \bar{P}^u_{GPU} + n^j_{MIC} \times \bar{P}^u_{MIC} \quad (1)$$

For instance, if a job uses 10 CPU cores, one GPU and no MIC, and the average power per CPU core for that user is 7.9W, while the average GPU power is 47.5W, then the predicted power for the job will be $10 \times 7.9 + 47.5 = 126.5$W. This value is used at all time points t when the job is active, hence the job profile is static. In the rare case where no user data for training existed, we used a global (over all users) average power consumption per unit in the EAM.

In previous work we compared the SVR approach with the EAM and we observed that performance improved for most of the users when using SVR. However, when an SVR model cannot be trained due to lack of data, the EAM provides a valid replacement since we obtained R^2 values greater than 0.5 over all user jobs.

3.2. Job length prediction

A second component of our analysis is concerned with prediction of job length. In order to predict computing power in advance, clearly we need to predict not only the power consumption for all jobs but also estimate their lengths. This allows us to predict which jobs will be present in the system at a future time.

The job length prediction component of our model is based on a very simple heuristic: search the user history for a similar job and use the duration of that job as the predicted duration of the current job. We chose this approach based on observations made on the data, that we will outline below. With these observations, we devised a simple set of rules to find a similar profile in the user history. For this, we inspected the workload and extracted a so-called *job profile* for each job. This includes

DOI: 10.1002/cpe

the job name, the queue name, the user-declared wall-time, and the number of resources of each type (CPU, GPU, MIC, nodes) used. We analyzed users separately.

A first observation was that jobs with identical profiles did not always show similar durations over time. Instead, a sort of step function was observed: consecutive jobs with the same profile will have very similar running times for a period, then switch to a new set of similar running times, that could be larger or smaller. This indicates a user pattern where the user switches between states. A possible scenario is that at first the user tests the application on a small dataset, running several simulations, then moves to a larger one, again running one or more instances, then switches to a different dataset, and so on. Every time there is a change, we see in the data a shift in job duration, followed by a set of jobs with similar running times. The size of these sets ranges from one to tens or even hundreds of jobs. So, a good strategy for estimating job duration is to look at the last job with an identical profile and take that duration. This leads to good predictions most of the time, and large errors only when the user switches state.

Sometimes, however, a job with an identical profile may not be present in the user history. Hence we need to understand whether we can make a correspondence with different jobs. We inspected the job names and observed that while using the same job name is common, another common pattern employed by users is to have a common prefix for the job name, followed by a number, such as 'run1', 'run2', etc. Hence, if we do not find an identical job in the user history, we look for a job with the same prefix name, with the rest of the profile unchanged.

If this match is also not possible, we look for the last job that had the same name and was submitted to the same queue and with the same wall-time, even if the resources used are slightly different. The reason we use the queue as a criterion is that Eurora uses several queues intended for jobs of different sizes. If also this search fails, we look for the same match but with the name prefix rather than the exact name. If none of these rules give a match, we look for the last job with the same name, or, as a last resort, the same name prefix.

For instance, if we take as an example the job profile {job-name:TRANSFER1, wall-time:06:00, queue:parallel, ncpu:1, ngpu:0, nmic:0, nodes:1} then the first rule will provide a match if an identical profile is found in the user history. If not, the second rule would match the profile {job-name:**TRANSFER***, wall-time:06:00, queue:parallel, ncpu:1, ngpu:0, nmic:0, nodes:1}, i.e., where the job name has the same prefix but the trailing number is different. The third rule would match the last job that has the profile {job-name:TRANSFER1, wall-time:06:00, queue:parallel,**ncpu:*, ngpu:*, nmic:*, nodes:***}, i.e. regardless of resources used. The fourth rule would match a profile {job-name:**TRANSFER***, wall-time:06:00, queue:parallel, **ncpu:*, ngpu:*, nmic:*, nodes:***}. The last two rules will match the last profile that has the job name 'TRANSFER1' or 'TRANSFER*', regardless of the value of the other fields.

This procedure provided a match over 92% of the time. If all rules fail, then we take the user-declared wall time as the predicted duration. After applying all rules, prediction is capped by the wall-time, since the Eurora scheduler kills all jobs that exceed it.

3.3. System power model

The third component of our analysis is a model of *system-level power consumption*, taking as input the *power of the computational units*. We extracted measurements of power consumption at system level at 5-minute intervals from the Eurora database. Power of each individual CPU, GPU and MIC was also extracted and summed to obtain total computing power. Figure 4 plots power of the system versus total computing power for our dataset. It is clear that there is a strong relationship between the two power measures, with a very high Pearson correlation coefficient (0.939). Thus, we built a *linear model* of system power, denoted as LM, starting from the individual computing units. The model provides an *estimate* of the system-level power $P_S^*(t)$ at time t as a linear function of the *measured* power of computing units $P_C(t)$ at the same time t:

$$P_S^*(t) = LM(P_C(t)) \tag{2}$$

DOI: 10.1002/cpe

In order to evaluate the model, we opted for a classical cross-validation approach, where the model is trained and tested on separate datasets. In the following we will show results from training with all data from September 2014 and testing on data from the first week of October 2014.

3.4. Model integration

Having obtained job power profiles (predicted at 5-minute intervals using the SVR component of our model), we can calculate the total *predicted* power of computing units $P^*_C(t)$ at time t by summing the *predicted* power of individual jobs running on the system at time t, $P^*_j(t)$, together with the power of the idle units, $P_{idle}(t)$:

$$P^*_C(t) = \sum_{j \in \text{Jobs}} P^*_j(t) + P_{\text{idle}}(t) \tag{3}$$

Idle power can be measured once for each unit type, with P_{idle} being the sum over all idle units. For instance, if at a certain time t there are 3 jobs running, using in total 160 CPU cores, 2 GPUs and 1 MIC, then the predicted system power $P^*_C(t)$ will be the sum of the power of each job, predicted by the SVM model, plus the idle power for the rest of the components (864 idle CPU cores, 62 idle GPUs and 63 idle MICs). In the following we will show results where SVR models were trained with data before October and applied to predict the total power for the first week of October 2014.

In Equation 3, "Jobs" denotes the set of all jobs running at time t. Hence, it is assumed that the set of active jobs is known in advance. In a real setting, however, this set also needs to be predicted, and we completed this task by using job length predictions resulting from our second model component. Specifically, at every time step we predict which job will still be running at the following time step (5 minutes in the future) by considering the set of all the jobs whose predicted length indicates they will still be online. At the same time, every 5 minutes we can ensure that jobs that are known to have finished are removed from the set of online jobs. Hence, if we consider the set of jobs predicted to be running at time t to be Jobs^*, then Equation 3 becomes:

$$P^{**}_C(t) = \sum_{j \in \text{Jobs}^*} P^*_j(t) + P_{\text{idle}}(t) \tag{4}$$

Then, to obtain the desired *system-level power predictions*, one only needs to apply the linear model LM described in Section 3.3 to the predicted computing power P^{**}_C from Equation 4 :

$$P^*_S(t) = LM(P^{**}_C(t)) = LM\left(\sum_{j \in \text{Jobs}^*} P^*_j(t) + P_{\text{idle}}(t)\right) \tag{5}$$

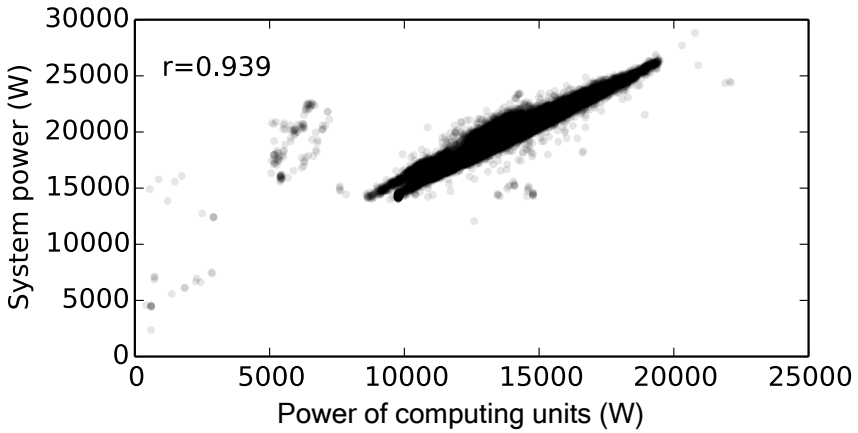

Figure 4. Power consumed by the entire system versus power of computing units only.

DOI: 10.1002/cpe

It is important to note that the linear model LM is trained using the *real* (measured) power consumption of computing units, $P_C(t)$ from Equation 2. However, in the final model it is applied to the *predicted* power of computing units, which uses both job power and length prediction, $P_C^{**}(t)$.

Again, application of the system-level model on test data from the first week of October 2014 will be shown below. Both linear regression and SVR were performed using the *scikit-learn* Python package [7], while data preprocessing for feature extraction was performed using the BigQuery cloud platform [8].

3.5. Evaluation criteria

At each step, the models were evaluated using two standard criteria for regression: the normalized-root-mean-squared-error (NRMSE) and R-squared (R^2).

$$\text{NRMSE} = \frac{\sqrt{(\sum_{i=1}^{N}(P_S(t_i) - P_S^*(t_i))^2)/N}}{\bar{P}_S} \tag{6}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(P_S(t_i) - P_S^*(t_i))^2}{\sum_{i=1}^{N}(P_S(t_i) - \bar{P}_S)^2} \tag{7}$$

where N is the number of time points considered, $P_S^*(t_i)$ and $P_S(t_i)$ are the predicted and real system-level powers at time t_i, respectively, while $\bar{P}_S$ is the average of the real system power over all N data points.

NRMSE measures the error between prediction and real data as a fraction of the average measured power. It takes positive values only, with small values meaning errors are much smaller than the average power levels. The R^2 criterion includes information on variability in the data and compares the errors to the natural variability. It tells us how the model performs compared to the so-called "average model" — a model where power is predicted to be the average of all power levels measured. A value close to 0 indicates that the model is no better than the average model (error is comparable to the standard deviation of the data), while larger values correspond to models better than random (with 1 corresponding to perfect prediction).

4. RESULTS

In this section we discuss the performance of our integrated model. We first evaluate the prediction of power at the level of computing units, then we move on to system-level power.

4.1. From workload to power of computing units

Power consumption of computing units is predicted by our model based on workload measures. We employ job power predictions by SVR (first model component) and job length predictions using our heuristic (second model component). In this section we will evaluate the two mechanisms separately.

First, we discuss the predicted computing power obtained by summing job-level predictions and considering the job length to be known (Equation 3). Figure 5 displays the total predicted power after summing over all users, compared to the measured time series.

The model performance is extremely good, with errors under 3% and very high R^2 values. Job power prediction, as showed in [6] provided accurate predictions (NRMSE under 20% and R^2 values over 0.5) for over 80% of the users, however for some users results were less accurate. Considering all users together allows for errors for some of the users to be compensated by others so that final performance is very good. This means that prediction of total power of computing units can be successfully achieved starting from job power predictions.

Secondly, we introduce the prediction of job length, to be used together with predicted job power. To evaluate our approach for job length prediction, we computed the absolute error between the real length and that obtained with our heuristic. For all jobs ever logged in the system, the mean absolute error was 38.9 minutes, which is a much better prediction compared to the user-defined wall-time,

DOI: 10.1002/cpe

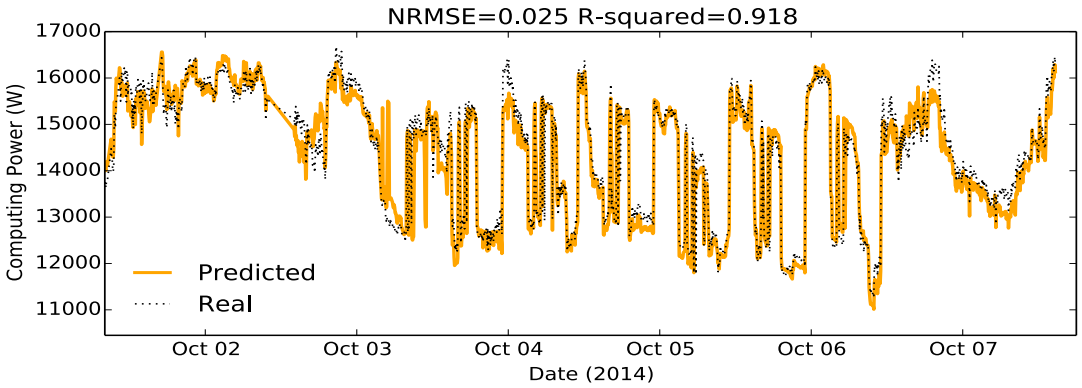

Figure 5. Power consumed by computing units, obtained by predicting power for individual jobs (Equation 3). Here we assume that job length is known. The dashed line shows the real (measured) power of computing units.

which is typically used to estimate job duration, and which would have provided a mean error of 225.11 minutes. When looking at the the first week of October 2014 only, the mean absolute error was 6.94 minutes. Figure 6 shows the cumulative distribution of October errors for our approach. We can see that our method obtains an error smaller than 1 minute for over 65% of the jobs, while almost 90% of jobs display errors under 10 minutes. For some jobs we have higher errors. These jobs correspond to shifts in user state (e.g. the same job that was short now starts to last much longer, possibly due to a change in input data).

The question now is how does this additional error in job length affect computing and then system-level power prediction. For this, we apply equation 4 to obtain predicted computing power starting from workload measures, hence combining the power and length prediction results (the first two components of our model). Figure 7 shows a comparison between the predicted and measured power consumption at the level of computing units.

As the figure shows, computing power is still reproduced very well. As expected, the error grows, from 2.5 to 6.8% with some periods predicted much better than others. We can observe some periods where our model underestimates computing power, and this corresponds to underestimation of job length during prediction.

4.2. From computing units to system-level power

In order to be able to predict system-level power consumption starting from the power of computing units, we first need to train the linear model that is the third component of our integrated approach.

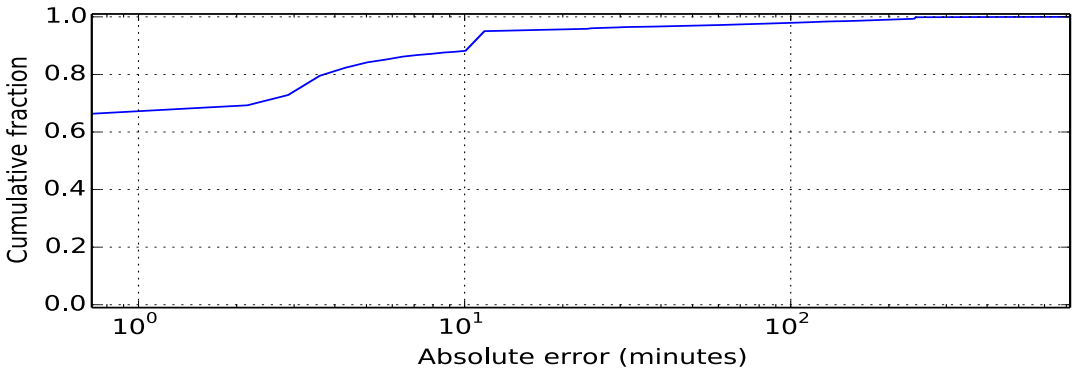

Figure 6. Cumulative distribution of errors for job length prediction, tested on the jobs from the first week of October 2014.

DOI: 10.1002/cpe

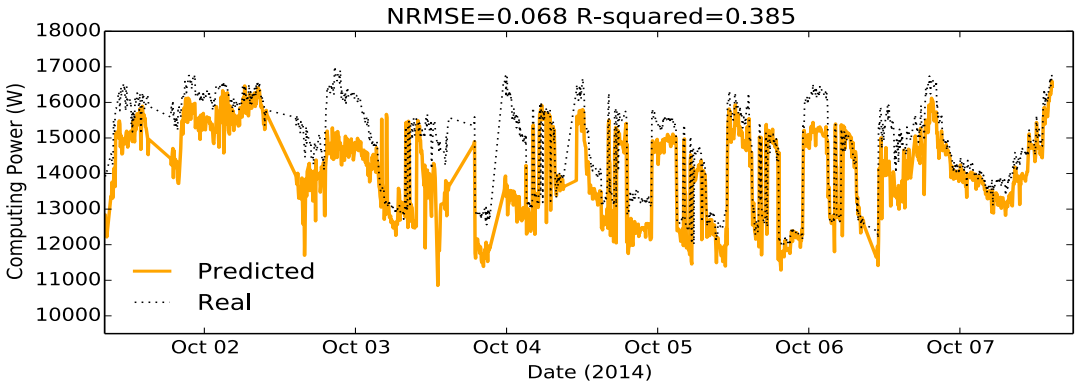

Figure 7. Power consumed by computing units, obtained by predicting both power and duration for individual jobs (Equation 4). The dashed line shows the real (measured) power of computing units.

For cross-validation, the model was trained with Eurora data from September 2014 and tested on data from the first week of October 2014. Increasing the amount of training data did not improve performance, so we decided to limit the training period to one month, meaning that in an online setting previous historical data could be discarded thus saving storage resources.

Figure 8 shows the real and estimated power time series, with NRMSE and R^2 values included. The linear model provides a very good fit, with errors bellow 5% and high R^2 value. This is a strong indication that a linear model can extrapolate very well from computing power to system power. It is important to note that this first evaluation at system level starts with *measured* computing power. In the rest of this section we will discuss performance on the *predicted* computing power introduced in Section 4.1.

We thus combine the three models to obtain system-level power predictions from predicted workload measures. We apply the linear model evaluated in Figure 8 to the predictions shown in Section 4.1. Again, we first evaluate performance when employing only job power predictions, so considering the job length to be known. Figure 9 displays the prediction result.

As the figure shows, prediction is very similar to the measured system power, with overall errors for the first week of October 2014 of under 3% and very high R^2. We can observe a slight trend of underestimating large power levels, especially for singular peaks, and overestimation of very low power levels. The former appears to be due to underestimation at step 2 (Figure 5) while the latter seems to be caused by the linear model overestimating lowest values (Figure 8).

However, this prediction is rather unrealistic, since in a real setting job length would not be known in advance. To estimate a more realistic performance, we apply the linear model also to the power of computing units shown in Figure 7, i.e., employing both power and length prediction for jobs. Results are displayed in Figure 10.

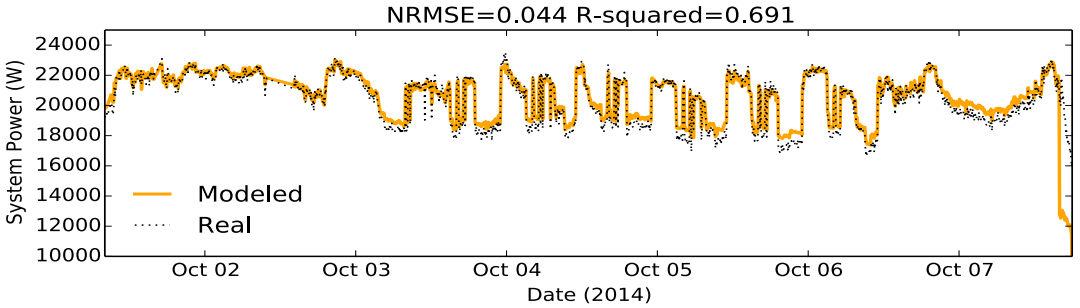

Figure 8. Power consumed by the entire system, estimated, using a linear model, from power of computing units. The dashed line shows the real (measured) system-level power consumption.

DOI: 10.1002/cpe

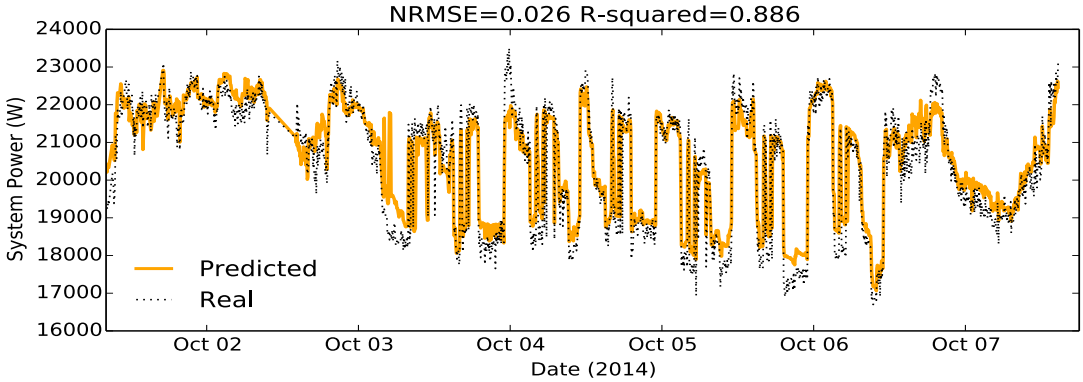

Figure 9. Power consumed by the entire system, forecasted with the linear model from predicted computing power obtained from job power predictions. Here we consider the job length to be known. The dashed line shows the real (measured) system-level power consumption.

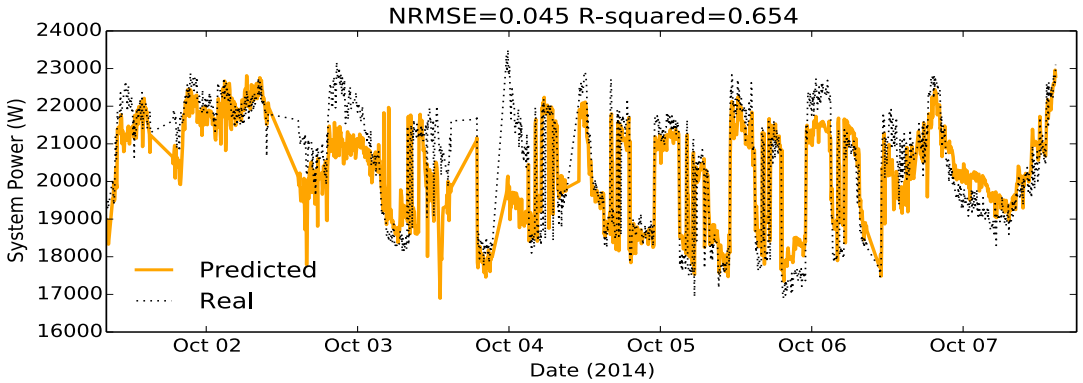

Figure 10. Power consumed by the entire system, forecasted with the linear model from predicted computing power obtained from both job power and job length predictions. The dashed line shows the real (measured) system-level power consumption.

We can observe again a slight increase in error, compared to the case when job length was assumed to be known: 4.5% versus 2.6% errors. However, this was to be expected, and predicted power remains very close to the real measured power. In conclusion, after integration of the three components, our model is able to faithfully predict system-level power. It is important to note that this prediction does not use as features any measurement of power at any level, nor any performance measures, but only information on workload (number of computing units used by jobs for the various users). Hence, online application of the model would require monitoring only the workload. However, historical power measurements are necessary for model training.

5. DISCUSSION AND APPLICATIONS

The analysis presented was performed off-line for historical data. In practice, it is intended for on-line use, for instance when using the live monitoring system to extract feature values and compute power predictions in real time for future time windows. As we have discussed in previous work [6], prediction of job power profiles does not imply a large overhead to the system. Job length prediction is very simple, requiring only to record the previous job profiles and corresponding lengths for each user, and requiring limited processing power. Training and applying the linear model also has low overhead, with negligible running times required. Hence, all in all, the entire model is straightforward to employ.

DOI: 10.1002/cpe

The results presented (Figure 10) are related to predicting future system-level power for a 5-minute time window. We restricted the time window in order to be able to take into account the fact that job length overestimations can be corrected simply by looking at the system itself: even if the predicted job length suggests that one job may still run for a long time, it may happen that the job actually finishes earlier. In this case *prediction* is overridden by *measurement*, improving performance. In a real setting, this window can of course be varied. The maximum future time window for which prediction can be achieved depends on factors such as the frequency of job submissions by the user, job length and load of the system. Every time a new job is ready to start, our model can predict the power profile for the system, given also the other jobs currently running on the machine, with their respective predicted job lengths. Thus, a complete system-level profile can be obtained at least until a new job is started. Then the profile changes according to the new job. On a very busy system with long queues, prediction can be obtained for longer periods of time, since we can know in advance what jobs will be scheduled next. On a lightly used system, the prediction window depends practically only on the time between job submission by users, which is more difficult to foresee. HPC systems in general work with relatively long and heavy jobs and queues that are always busy, making our approach very useful in this context.

The method presented here is easily applicable to any HPC system. Of course, this involves training models of job power consumption and job duration for the users of the new system, and learning the dependencies between power at computing units and system level, using the methodology outlined in this paper. It is possible that the linear relation between system and unit power does not hold in some systems, in which case the linear model can be replaced by a non-linear regression model. Once the three model layers are learned, they can be combined into one prediction framework. Thus, on the new system, the required measurements are workload (number of units of each type used by the jobs) and power consumption at computing and system level. These are typically available on HPC systems (e.g. [9]). When power of computing units is missing, temperatures can be used as their proxies [9]. With regard to applications outside the HPC domain, this is still possible but would depend on the system type. The data-driven approach is based on a certain stability of the system, in terms of components and user behavior. Hence, while for certain homogeneous systems such as clouds this may be the case, for other heterogeneous systems like grids, our method would not be able to learn power patterns.

System-level power predictions can have several applications for optimization of behavior for HPC systems. Besides providing a tool for operators to be aware of future power values, the model can also be used to decrease power consumption by adjusting job scheduling and resource allocation. Power for various scheduling and allocation schemes for the same workload can be predicted, and the scheme with the lowest power employed. This is made possible due to the wide range of workload features considered, which include global description of resources allocated to jobs. Specifically, the number of nodes that a job uses and the number of cores in use by other jobs change from one allocation scheme to another, changing thus power consumption. Our task would be to construct a low-power mapping of resources to jobs, which would require some form of search-based optimization. For instance, methods relying on constraint programming are already widely used for HPC scheduling and could be extended to take into account power and job length predictions [10]. Evolutionary techniques could also be a possibility. The disadvantage of these methods, when facing currently used HPC schedulers, is their running time, hence suitable implementations are needed.

Another example of a power-aware technique that is commonly explored, especially given the increased power needs of HPC infrastructures, is power capping. This technique is not concerned with decreasing overall energy consumption, although this may happen as a side effect, but concentrates on maintaining total system power at bay, so that the capacity of the energy provisioning system is not exceeded. Here too, our model can provide important information on power for future system states, so that scheduling meets the power capping needs.

DOI: 10.1002/cpe

6. RELATED WORK

With energy needs becoming a major concern for large computational infrastructures, numerous recent research efforts have focused on analysing and reducing power usage [11, 12]. A large amount of work regards modeling power for various types of computing units, starting from load, frequency and other hardware counters. For instance, single and dual core CPU power is modeled in [13] by considering the relation between the probability distribution functions of load and power, while servers with up to 8 cores are studied in [14, 15]. GPU power is estimated from load measures in [16]. These methods do not allow for advance prediction in real life scenarios, since load and hardware counters cannot be known in advance, unless they can be predicted through other methods. Our method is significantly different in that we model total system power starting exclusively from workload measures, without the need to monitor the individual components, enabling *advance prediction of power*.

Power requirements of HPC applications has also been analyzed in recent years. For instance, the US Department of Defense are using application signatures to predict power consumption across different architectures [17]. Performance counters are used to model application power on three small scale HPC platforms by [18]. GPU CUDA kernels are analyzed in [19], again based on job performance counters. Recently, we have introduced a method [6] based on Support Vector Regression (SVR), which builds one power model per user to predict job power consumption based on workload in Eurora. This method has the advantage over others in that it does not require instrumenting the applications to extract signatures and performance counters, but only needs the number of resources required, making it much more straightforward to apply. In this work, the SVR method was employed to predict power of computing components, from which we then obtained system-level power. Recently, another method for predicting power solely from workload data was introduced [20], however the authors concentrate on predicting mean power for jobs and not full power profiles like we do.

In the quest for Exascale computing systems, where energy needs will be much greater, it is important to analyze power of large computing infrastructures *at system level*. Related work cited above looks only at individual computing components or jobs, while we concentrate on total system power including hardware other than computing components, such as networking and I/O. Very few other examples of power analysis at system level exist in the literature, despite a recognized need for development in this direction [21]. For example, recently Google has introduced a method [22] of modeling *Power Usage Effectiveness* (PUE) through an Artificial Neural Network which takes as input workload, cooling, power, together with other external information such as outside temperature, wind speed, etc. This allowed for testing various data center scenarios and improving PUE for the system under analysis. Resource usage indicators (such as operations per second) are used in [23] to model power at system level for a heterogeneous datacenter. Using nonlinear transformation of raw resource usage indicators, the authors generate a set of features of interest that are then mapped to power consumption using linear regression. Again, all these models are useful for estimating power consumption while measuring the input features, but do not allow for power prediction.

Some work also exists for prediction of job length on HPC systems. Typically, the prediction task is solved using machine learning techniques, where various job properties are used as input to a model that can predict the remaining running time for the job. For instance, [24] record the memory and CPU usage for jobs in time, and provide an estimate of the remaining running time, using a statistical classifier. A similar approach is [25], where kernel logs are used to monitor job execution, in order to predict remaining running time. The authors use a Hidden Markov Model to perform the prediction. In both examples, jobs need to be monitored after they have started to understand their characteristics and respond to changes. Our work instead uses only general job information available just before the job is started, hence it is much more flexible. Additionally, our heuristic does not require any complicated training procedure, while producing very good prediction results. This provides an advantage for on-line usage.

DOI: 10.1002/cpe

Job duration prediction has also been analyzed in the context of job scheduling [26]. Here predictions are performed using linear regression, using the past running times and resource utilization as features. The authors show that slowdown during scheduling can be improved by 28% through prediction of job length. Our predictions can also be employed to optimize scheduling approaches, as we intend to do in future work, with the additional advantage that we can obtain power-aware schedulers by combining power and length predictions for jobs. Examples of power-aware techniques include power-capping, such as the works of [27, 28]. Studies along these lines could benefit from accurate system-level power prediction that we introduce in this paper. Power-reducing scheduling techniques have also been investigated using Dynamics Voltage and Frequency Scaling in virtualized cloud environments [29, 30].

Our predictions can also enable power-reducing scheduling, due to the fact that we employ features related to resource allocation. That means that different load levels for a node are incorporated in the model. Different loads typically lead to different voltage/frequency scales.Thus power usage under various such scales can be learned by our model provided enough data is available.

A different application of power models is prediction or identification of anomalous behavior. For instance, the Google PUE model [22] allowed for identification of anomalies in monitoring, when the model did not fit the data any more. Similarly, a decrease in modeling performance can predict system failures as well. Our model can also be used to predict anomalous behavior, as we discussed in previous work [5].

7. CONCLUSIONS

We have presented a 3-layer model of system-level power consumption for Eurora, a hybrid HPC installation containing CPUs, GPUs and MICs. The model takes as input workload parameters, namely job names and resources allocated to each job. It first computes a predicted profile of power consumption for each job using Support Vector Regression, and forecasts job duration with a simple heuristic. The two predictions are then used to estimate power for computing units. This estimation is provided as input to a linear model able to predict total power at system level, including also networking, IO and other elements.

The approach achieves very good performance on test data, with errors under 5% for the first week of of October 2014. The methodology can be easily applied to other systems since the data types used are generally available in most HPC systems.

We have discussed applications of our predictions. One is power optimization through job dispatching. Being able to forecast job length and power can enable dispatchers to choose among different allocation schemes that reduce consumption. A different application could be capping power usage for HPC systems, again in the context of scheduling algorithms. Both applications are considered for future work, possibly together with investigating alternative power models based on hardware counters, which could be adapted to perform advance power prediction rather than real time estimation. Knowledge of the application code for jobs could also help, however these data are not currently available in our system. The methodology outlined here can be easily adapted to other HPC systems for which the same data types are available.

ACKNOWLEDGEMENT

BigQuery analysis was carried out through a generous Cloud Credits grant from Google. The Eurora database was developed by Prof. L. Benini and his group in the University of Bologna in collaboration with the HPC group at CINECA. We are particularly grateful to Dr. A. Bartolini for useful discussions regarding the data and the system as a whole and to Dr. E. Rossi and Dr. C. Cavazzoni for providing access to the CINECA systems. We acknowledge the CINECA ISCRA PACNA and PM-HPC awards allowing access to HPC resources. This work has been partially funded by the European project SoBigData Research Infrastructure — Big Data and Social Mining Ecosystem under the INFRAIA-H2020 program (grant agreement 654024).

DOI: 10.1002/cpe

REFERENCES

1. Cavazzoni C. Eurora: a european architecture toward exascale. *Future HPC Systems: the Challenges of Power-Constrained Performance*, ACM, 2012.
2. Smola A, Vapnik V. Support vector regression machines. *Adv Neural Inf Process Syst.* 1997; **9**:155–161.
3. CINECA: The Italian Interuniversitary Consortium for High Performance Computing. www.cineca.it.
4. Bartolini A, Cacciari M, Cavazzoni C, Tecchiolli G, Benini L. Unveiling eurora-thermal and power characterization of the most energy-efficient supercomputer in the world. *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2014.
5. Sîrbu A, Babaoglu O. Predicting system-level power for a hybrid supercomputer. *2016 International Conference on High Performance Computing & Simulation (HPCS)*, 2016; 826–833.
6. Sîrbu A, Babaoglu O. Power Consumption Modeling and Prediction in a Hybrid CPU-GPU-MIC Supercomputer. *Euro-Par 2016: Parallel Processing – 22nd International Conference on Parallel and Distributed Computing*, Springer International Publishing, 2016; 117–130.
7. F Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 2011; **12**:2825–2830.
8. Tigani J, Naidu S. *Google BigQuery Analytics*. John Wiley & Sons, 2014.
9. Sîrbu A, Babaoglu O. A Holistic Approach to Log Data Analysis in High-Performance Computing Systems: The Case of IBM Blue Gene/Q. *Euro-Par 2015: parallel Processing Workshops, LNCS 9523*, Springer, 2015; 631–643.
10. Bonfietti A, Lombardi M, Milano M. Embedding decision trees and random forests in constraint programming. *Integration of AI and OR Techniques in Constraint Programming*. Springer, 2015; 74–90.
11. Fraternali F, Bartolini A, Cavazzoni C, Benini L. Quantifying the impact of variability and heterogeneity on the energy efficiency for a next-generation ultra-green supercomputer. *IEEE Transactions on Parallel and Distributed Systems* 2017; .
12. Silvano C, Bartolini A, Beccari A, Manelfi C, Cavazzoni C, Gadioli D, Rohou E, Palermo G, Agosta G, Martinovič J, *et al.*. The antarex tool flow for monitoring and autotuning energy efficient hpc systems. *SAMOS 2017-International Conference on Embedded Computer Systems: Architecture, Modeling and Simulation*, 2017.
13. Dargie W. A stochastic model for estimating the power consumption of a processor. *IEEE Transactions on Computers* 2015; **64**(5):1311–1322.
14. ITakouna, Dawoud W, Meinel C. Accurate Mutlicore Processor Power Models for Power-Aware Resource Management. *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC)*, 2011.
15. Kim M, Ju Y, Chae J, Park M. A Simple Model for Estimating Power Consumption of a Multicore Server System. *International Journal of Multimedia and Ubiquitous Engineering* 2014; **9**(2):153–160.
16. Ma X, Dong M, Zhong L, Deng Z. Statistical power consumption analysis and modeling for GPU-based computing. *ACM SOSP Workshop on Power Aware Computing and Systems (HotPower)*, 2009.
17. Olschanowsky C, Rosing T, Snavely A, Carrington L, Tikir M, Laurenzano M. Fine-Grained Energy Consumption Characterization and Modeling. *DoD High Performance Computing Modernization Program Users Group Conference (HPCMP-UGC)*, 2010; 487–497.
18. Witkowski M, Oleksiak A, Piontek T, Weglarz J. Practical power consumption estimation for real life HPC applications. *Future Generation Computer Systems* 2013; **29**(1):208–217.
19. Nagasaka H, Maruyama N, Nukada A, Endo T, Matsuoka S. Statistical power modeling of GPU kernels using performance counters. *International Green Computing Conference (IGCC)*, 2010; 115–122.
20. Borghesi A, Bartolini A, Lombardi M, Milano M, Benini L. Predictive modeling for job power consumption in hpc systems. *International Conference on High Performance Computing*, Springer International Publishing, 2016; 181–199.
21. Dayarathna M, Wen Y, Fan R. Data center energy consumption modeling: A survey. *Communications Surveys & Tutorials* 2015; .
22. Gao J. Machine learning applications for data center optimisation. *Google White Paper* 2014; .
23. Canuto M, Bosch R, Macias M, Guitart J. A methodology for full-system power modeling in heterogeneous data centers. *Proceedings of the 9th International Conference on Utility and Cloud Computing*, ACM, 2016; 20–29.
24. Ejarque J, Micsik A, Sirvent R, Pallinger P, Kovacs L, Badia RM. Semantic resource allocation with historical data based predictions. *Proceedings of CLOUD COMPUTING 2010 : The First International Conference on Cloud Computing, GRIDs, and Virtualization*, IARIA, 2010; 104–109.
25. Chen X, Lu CD, Pattabiraman K. Predicting job completion times using system logs in supercomputing clusters. *Dependable Systems and Networks Workshop (DSN-W), 2013 43rd Annual IEEE/IFIP Conference on*, IEEE, 2013; 1–8.
26. Gaussier E, Glesser D, Reis V, Trystram D. Improving backfilling by using machine learning to predict running times. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ACM, 2015; 64.
27. Borghesi A, Collina F, Lombardi M, Milano M, Benini L. Power Capping in High Performance Computing Systems. *Principles and Practice of Constraint Programming (CP), LNCS 9255*. 2015.
28. Borghesi A, Conficoni C, Lombardi M, Bartolini A. MS3: A Mediterranean-stile job scheduler for supercomputers-do less when it's too hot! *International Conference on High Performance Computing & Simulation (HPCS)*, 2015; 88–95.
29. Shojafar M, Canali C, Lancellotti R, Abawajy J. Adaptive computing-plus-communication optimization framework for multimedia processing in cloud systems. *IEEE Transactions on Cloud Computing* 2016; .
30. Shojafar M, Canali C, Lancellotti R, Abolfazli S. An energy-aware scheduling algorithm in dvfs-enabled networked data centers. *Proc. CLOSER* 2016; :387–397.