ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/334101090

Modeling Energy Consumption Based on Resource Utilization

Chapter · June 2019

DOI: 10.1007/978-3-030-24289-3_18

CITATIONS
0

READS
194

3 authors:


Lucas Venezian Povoa
Rede D'Or
11 PUBLICATIONS **57** CITATIONS

SEE PROFILE


C. A. C. Marcondes
Instituto Tecnologico de Aeronautica
84 PUBLICATIONS **390** CITATIONS

SEE PROFILE


Hermes Senger
Universidade Federal de São Carlos
70 PUBLICATIONS **361** CITATIONS

SEE PROFILE

All content following this page was uploaded by Lucas Venezian Povoa on 30 July 2020.

The user has requested enhancement of the downloaded file.

Modeling Energy Consumption based on Resource Utilization

Lucas Venezian Povoa[1,2], Cesar Marcondes[2], and Hermes Senger[3]

[1] Federal Institute of São Paulo (IFSP), Caraguatatuba, Brazil
venezian@ifsp.edu.br
[2] Aeronautics Institute of Technology (ITA), São José dos Campos, Brazil
marcondes@ita.br
[3] Federal University of São Carlos (UFSCar), São Carlos, Brazil
hermes@ufscar.br

Abstract. Power management is an expensive and important issue for large computational infrastructures such as datacenters, large clusters, and computational grids. However, measuring energy consumption of scalable systems may be impractical due to both cost and complexity for deploying power metering devices on a large number of machines. In this paper, we propose the use of information about resource utilization (e.g. processor, memory, disk operations, and network traffic) as proxies for estimating power consumption. We employ machine learning techniques to estimate power consumption using such information which are provided by common operating systems. Experiments with linear regression, regression tree, and multilayer perceptron on data from different hardware resulted into a model with 99.94% of accuracy and 6.32 watts of error in the best case.

Keywords: Computer architecture · Energy consumption modeling

1 Introduction

Over the years, managing energy efficiency of Information and Communication Technologies (ICT) has increasingly emerged as one of the most critical environmental challenges. Due to ever increasing demand for computing resources, emissions footprint, increased energy price and tougher regulations, improving energy efficient became priority for datacenters, especially to the massive ones. This concern is pervasive in ICT, from development of more energy efficient devices to greener virtualization, resource consolidation, and, finally, definition of new architectures, services, and best practices.

In 2007, a Gartner's Report showed that ICT industry generated 2% of global CO2 [1] emissions. From which, 23% came from datacenters. A Greenpeace's report [2] stated that "datacenters are the factories of the 21st century in the Information Age", however, they can consume as much electricity as 180,000 homes.

Constant reduction in computation resources prices, accompanied with popularization of on-line businesses, and wide spread of Internet and wireless networks, lead to the rapid growth of massive datacenters, consuming large amounts of energy. Indeed, nowadays, datacenters that execute Internet applications consume around 1.3% of the energy produced in the world [3]. It is expected in 2020 that this amount will rise to near 8% [4]. In such scenario, improving power efficiency on ICT installations and datacenters is mandatory. To overcome this challenge, several strategies have been proposed, such as resource consolidation [5–7], and improving resources utilization [8].

In general, better energy efficiency can be achieved by means of actuation strategies which need the continuous power consumption measurement. The deployment of power meters may be prohibitive in terms of cost in datacenters with many thousands of computers. Furthermore, external metering instruments require physical system access or invasive probing [9], which can be not available. On the other hand, software estimators for power consumption can be easily deployed at almost negligible cost.

An usual approach is to use internal performance counters provided by the hardware [10] and by the operating system to derive models that estimate power consumption [11–14]. Such models can be used by on-the-fly power saving strategies which need continuous power consumption estimation. Other possible applications include simulators that evaluate the power consumption of workloads based on performance and resource usage counters (e.g., register file usage, number of page faults, number of I/O operations per second).

In a previous work [15], we studied the correlation between a set of resource utilization counters provided by an operating system and the power consumption on a typical server machine. In this paper, we propose three novel models that use counter of both performance and resource utilization as proxies for power consumption, overtaking state-of-the-art accuracy. Besides that, differently from most of the related work, our models are not limited to predict power consumption of specific components, but of whole machine. We assume a good model should include all performance counters which significantly influence the power consumption. However, the excess of parameters and non-linear relations between these variables and power consumption can produce complex and inaccurate models. Having this on mind, we also investigate which operating system counters can be used to build robust and accurate models. Now, we further elaborate on correlation analysis and estimation of power consumption from resource utilization variables (i.e., counters) provided by operating systems. With this purpose, we apply nine models based on (*i*) Multiple Linear Regression (MLR), (*ii*) Regression Tree (RET), and (*iii*) Multilayer Perceptron (MLP), an Artificial Neural Network (ANN) which are experimentally evaluated on two different hardware [4].

[4] Models were implemented in R (using RSNNS) and source code are available under the GNU General Public License version 3 at `https://github.com/lucasvenez/ecm` along with the employed dataset.

Remainder of this paper is organized as follows. Section 2 describes the modeling approach, the workload, and the testbed used for experiments. Section 3 shows our variable analysis and selection approach. Section 4 describes proposed power consumption models based on the MLR, RET, and MLP methods. Section 5 presents the analysis of each proposed model. Section 6 describes some related work and compare some of them with our results. Finally, Section 7 points out our final remarks.

2 Modeling energy consumption from resource consumption data

This paper aims to provide a characterization of the power consumption for a wide variety of machines. We propose new models which provide accurate estimations for the power consumption. Our models are based on resource utilization measurements commonly supported by the operating system used from commodity computers to datacenter servers.

2.1 Modelling Approach

In order to model power consumption for different computers, we employed a six-steps method.

1. *Data Collection:* comprehends a synthetic workload execution while an agent is used to collect data about resource utilization from the operating system [16, 17]. The agent captures forty seven variables from the directory */proc*.
2. *Feature Engineering:* this step aims to calculate new variables from the raw ones in order to improve generalization and accuracy of models.
3. *Variables Selection:* in this step variables that are influential to power consumption are selected. We employed a correlation method called Maximal Information Coefficient (MIC) [18] that evaluates the correlation of a pair variables regardless of the distribution.
4. *Model Construction:* aims to fed models with resource utilization samples and reads of the actual energy consumption measured in the testbed.
5. *Model Analysis:* focus on evaluating models accuracy through a set of different metrics with a special attention to avoid overfitting.

The final step is called *Model Selection*, where the best model for power consumption is selected.

2.2 Data Collection

We built a synthetic workload instead of using real applications or benchmarks aiming to conceive energy consumption models which are suitable for any application, while avoiding collinearity problems which may compromise regression models [19, 14].

Our workload was designed to avoid cross dependency among the variables fed to the model and produce as much as possible power consumption states for all system components such as memory, hard disk, processor, network interfaces and I/O operations [20]. It was implemented by using three open source tools: (*i*) *stress* [21] was used to produce utilization of resources such as processor, memory, hard disk and I/O operations; (*ii*) *cpulimit* [22] was utilized to generate random periods of idleness to produce several levels of processor utilization; and (*iii*) *iperf* [23] was employed to generate network traffic.

Workload was produced with the following characteristics. CPU utilization varied between 0% and 100% in several cycles, being increased in steps of 5% each. Each experiment was composed by $P_i = 2i - 1$ processes with $1 \leq i \leq N_{cpu}$, where N_{cpu} is the number of processors in the machine for the i^{th} test. Memory utilization ranged from 512MB to the physical memory size. For the i^{th} experiment, one application process allocates $M_i = 256(i+1)$ MB of memory, such that $1 \leq i \leq {}^{M_{size}}/_{256} - 1$. Hard disk utilization varied from 1GB to 64GB, being produced by one process. For each experiment, the amount of disk space allocated is $C_i = 2i - 1 \bmod 17$ GB, where i is the experiment number.

I/O workload was expressed by the number of processes that performed the message exchanges between main memory and hard disks. The amount of processors exchanging messages was given by $P_i = 10i \bmod 10^2$, where i is the experiment number.

At first, only one parameter was selected to vary for each experiment, in order to capture its influence on power consumption. Then, parameters were varied to test every all-to-all combinations of several parameter levels, in order to capture their influence on power consumption as well as parameter interactions. For each combination of parameters and level, the workload is executed for two minutes. The overall experiment took about thirty hours to be carried out, producing about 51,000 entries for each dataset, each entry containing measures from 47 variables of resources utilization and the power consumption.

2.3 Testbed Used for Experiments

Testbed used for experiments is depicted in Figure 1. Some nodes were instrumented to measure power consumption while running workloads. We employed two nodes with different architectures in the experiments, which have their hardware configuration summarized in Table 1.

Table 1. Hardware configurations with one 1Gbps network interface running Ubuntu 11.10 kernel 3.0.0-12.

Hardware	A1	A2
Processor model	Intel Core i5-2400	AMD Opteron 246
Cores	4	2
Frequency	3.10 GHz	2.00 GHz
Memory	4GB SDRAN	8GB SDRAN
Disk	1×500GB	4×240GB

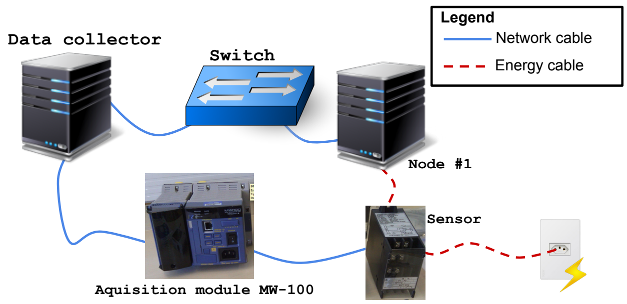

Fig. 1. Experiment environment with a node, an energy consumption meter, a module and a data storage.

In order to obtain precise power consumption measures, we used a power sensor Yokogawa model 2375A10 [24]. This device works connected to the power supply, and provides data to one data acquisition module model MW-100-E-1D [25]. The acquisition module probes and saves measures on power consumption in watts every 100 milliseconds. Our agent collected data from the acquisition module via a network interface using the telnet protocol every second along with the resource utilization variables.

3 Variables Analysis and Selection

Designing accurate models depend upon a good selection of resource utilization counters with appropriate transformations that present significant influence on power consumption and do not produce noise.

3.1 Feature Engineering

We transformed each independent variable v_j with cumulative values using equation $v_{i,j} - v_{i-1,j}$, where i is the sample index of the j^{th} variable. Because the number of processing cores of different architectures can vary, we summarized their values into a unique variable $ct = \sum_{j=1}^{m} c_{i,j}$, where m is the number of cores, i is the sample index, and $c_{i,j}$ is the data related to the i^{th} sample of the j^{th} core. This approach was also applied for multiple hard disks. These simple transformations help to improve accuracy and generalization, enabling a unique model to be applied for different hardware architectures.

3.2 Variable Selection

For the sake of clearness and understandability, a model for estimating energy consumption should be simple, i.e., to consider only a subset composed of the most influential variables on energy consumption. With this purpose we identified from the set of observed variables the subset with the highest correlation with the dependent one (i.e., the energy consumption).

In order to evaluate the correlation among variables, two main criteria should be considered. The *generality* refers to the capacity of identifying any relation type, not limited to specific types of correlation functions such as linear, exponential and periodic correlations. Later, the *equitability* is the ability to provide a unique index to express relation with the same noise level, even for functions of different types. With these two criteria on mind, we chose a method named MIC, which is part of a set of tools named Maximal Information-based Nonparametric Exploration (MINE) [18], to identify and select the most impacting variables for power consumption. MIC produces values between 0 and 1, where zero means absence of correlation between the pair of variables and 1 means full correlation.

For each architecture we generated a dataset containing the variables previously described. A third dataset (Mix) containing merged data from the two previous datasets plus one variable that describes whether the sample is related to the architecture A1 (value -1) or architecture A2 (value 1). Figure 2 shows results of the MIC between each independent variable, i.e., operating system's variables, and the dependent variable. In the chart the vertical black line represents the threshold of 10%, which was applied with the purpose of finding a reduced set of the most impacting coefficients and produce a model with good understandability.

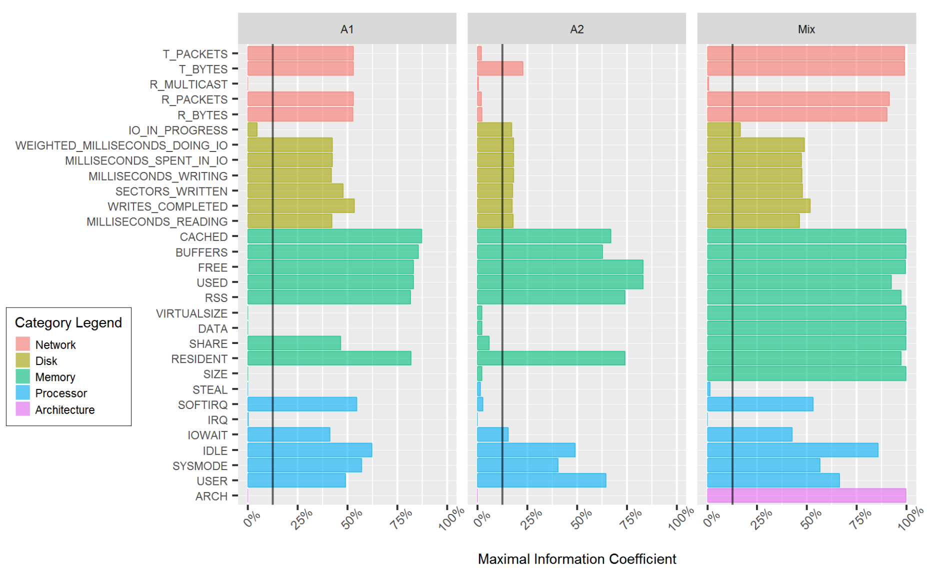

Fig. 2. Maximal Information Coefficient for the dataset of each architecture and for the mixture thereof.

3.3 Dependent Variable Analysis

Dependent variable distribution defines the method that can be used for modeling its behavior. The Kolmogorov-Smirnov test [26] resulted in *p-values* less than $2.2e-16$, which confirm that the dependent variable has no Gaussian distribution considering a significance level of 5%. This is evidenced in Figure 3. Datasets A1, A2, and Mix present an average energy consumption (watts) of

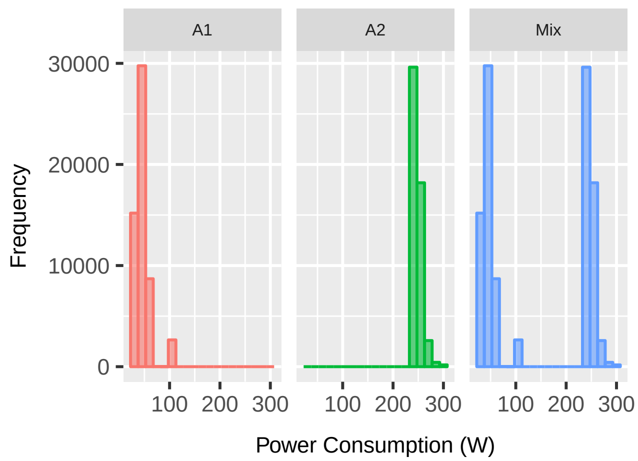

Fig. 3. Energy consumption histogram.

46.08, 249.23, and 142.62, respectively. Their standard deviation are 15.26, 7.74, and 102.19. It is noteworthy that the architectures A1 and A2 have an stable energy consumption but in different ranges.

4 Modeling Power Consumption

In this section we describe several models using different techniques for estimating power consumption based on the most influential variables described in the previous Section.

4.1 Multiple Linear Regression

A MLR is a type of regression analysis that maps a set of input values $\boldsymbol{X}$ to a response value y, requiring that $\boldsymbol{y} \sim N(\mu, \sigma^2)$. Because the datasets do not follow a normal distribution, we consider the Central Limit Theorem (CLT), which states that when the size of a given sample increases, the sampling distribution of its average or sum tends to a normal distribution [27]. CLT justifies modelling the energy consumption with the MLR defined as $\hat{y} = \alpha + \boldsymbol{\beta x} + \epsilon$, where $\hat{y}$ is the estimated value of the energy consumption, α is the intersection point of the line of adjustment with the ordinate, $\boldsymbol{\beta}$ is the regression coefficients vector, $\boldsymbol{x}$ is the vector of independent variables, and ϵ is the average random error.

This method employs the least-squares method for estimating the coefficients vector $\boldsymbol{\beta}$. Despite of the high correlation between the dependent variable and the independent variable ARCH, the MRL method cannot incorporate the former into the resultant model for the Both dataset. This limitation for generating a global energy consumption model will be detailed in Section 5.2.

4.2 Regression Tree - RET

A Decision Tree (DT) has a structure composed by leaves, branches and nodes aiming to define a nonlinear predictive model. A RET is a particular case of a DT, where values of dependent variables are continuous. Using a RET as predictor requires a sample be dropped down via the tree until a leaf, which returns the average of its values of the dependent variable [28]. A RET is created by splitting a node p into two children nodes. The tree stops to grow when the complexity index β_p of a node p is less or equals to a threshold α. For the experiments the threshold α was set as 1%.

RET models are easy interpreted, but our results show that important variables are excluded for the model, which evidence a limitation of this method for modeling energy consumption. The resulting model for the Mix dataset represents our worst model, which considers only one independent variable for defining itself. Some variation in RET's hyper-parameters was performed without improvements in final results described in Section 5.2.

4.3 Multilayer Perceptron

A MLP is an Artificial Neural Network model that maps a set of input values into a set of output values [29] after a learning process. It can be successfully applied in different areas, e.g., Biometrics [30], Thermal Engineering [31], Ocean Engineering [32], Climatology [33].

The MLP is composed by an input layer with n sensory units, h hidden layers with n_h neurons each, and an output layer with t neurons. A MLP has L layers, excluding the input layer, and its input values are propagated layer-by-layer. Its learning process can be supervised or unsupervised. Once we collected both the input and output variables, this research applied the supervised learning process. The supervised learning process was performed with the backpropagation algorithm with chunk update (also know as mini-batch), which has the following steps: *i)* forward step, where a set of input values is provided to the sensory units, and its effect is propagated layer-by-layer; and *ii)* backward step, where the weights are adjusted in accordance with an error-correction rule respecting the Mini-Batch Stochastic Gradient Descent method [34] after p (chunk size) executions of the forward step for different samples. Before starting the MLP training, all variable v_j had its values normalized, where $v_{ij} = (v_{ij} - min(v_j))/(max(v_j) - min(v_j))$.

The backward step starts by computing the error $e = {}^1\!/_n \sum_p^i (\hat{y} - y)^2$. Mean Squared Error was employed once it incorporates both the bias and the variance of a model [35]. After that, local gradient δ_j^L related to neuron j at output layer L was computed according to $\delta_j^L = e_j^L \times \varphi'(v^L)$. When one neuron j is located at a hidden layer $0 < l < L$, the local gradient δ_j^l related to neuron j at hidden layer l was computed by $\delta_j^l = \varphi'(v_j^l) \sum_{k=1}^{g} [\delta_k^{l+1} w_{kj}^{l+1}]$, where φ' is the derivative of activation function φ, and g is the number of neurons at layer $l+1$. New values for weight w_{ij}^l at layer l is defined according to $w_{ij}^l(n+1) = w_{ij}^l(n) - {}^\eta\!/_p \sum_m^n [\delta_j^l(m) y_i(m)]$, where n is the iteration number, p is the chunk size, and η is the learning-rate.

For setting the MLP's configuration for each architecture, we applied an empirical method consisting of *(i)* selecting a random and non-sequential subset of registers from our sample, 15% of all registers for train and 5% for test; *(ii)* starting the model weights with a random Gaussian distribution with values between 0 and 1; *(iii)* ranging the number of hidden layers from 1 until a descendant precision of the model; *(iv)* ranging the number of neurons at each layer from $\lceil \frac{v}{10} \rceil$ to $2v$, where v is the number of independent variables at the model; *(v)* ranging the learning-rate from 0.000 to 1.000 by 0.005; and *(vi)* calculating the model accuracy with test subset using the Coefficient of Determination R^2 metric.

In our study, the better configuration (i.e., with greatest R^2) for the MLP consists of 3 hidden layers, where each one has the number of nodes equals to double of the number of input variables, an output neuron representing the energy consumption value, a learning rate $\eta = 5$, and a chunk size $p = 50$. We employed the *tahn* function, as activation function φ, which yields larger partial derivatives with small changes in inputs [36].

5 Evaluating the Proposed Models

In this section, the power consumption models proposed are evaluated. For this purpose, different metrics and the 10-fold cross-validation (CV) method were employed.

5.1 Employed Accuracy Metrics

Four different classes of metrics were applied to evluate the proposed models: scale-dependent, percentage error, relative error, and scale-free error metrics [37]. *Scale-dependent* metrics are simple to understand and calculate, but cannot be applied to compared models of series with different scales. *Percentage error* metrics are scale independent, overcoming the limitations of scale dependent metrics. However, such metrics return infinite or undefined values when zeroes exist within the series. *Relative error* metrics are also scale independent metrics but they are restricted to some statistic methods when errors are small. Finally, *scale-free error* metrics never provide infinite or undefined values, and they can be applied to compare different estimate methods either over a single or multiple series.

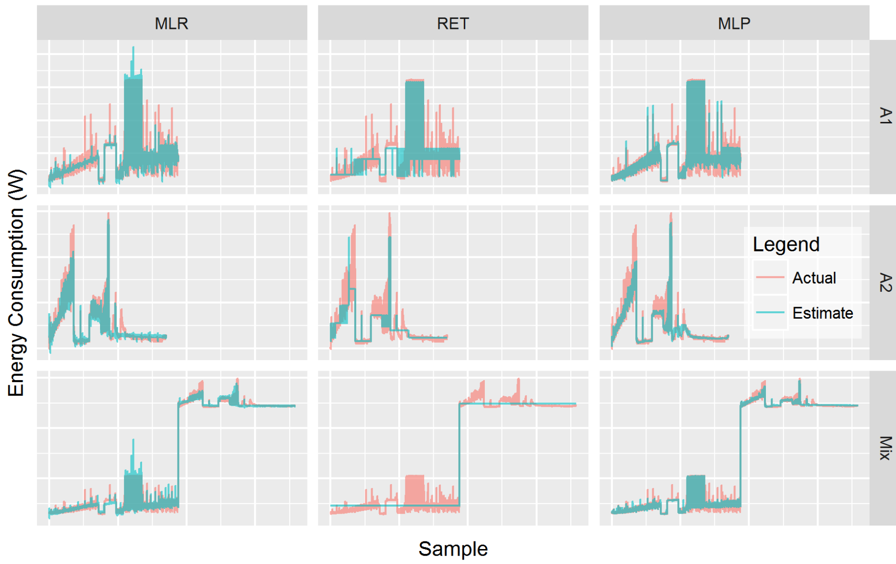

Fig. 4. Comparison between actual and estimated values.

For the sake of comparison to other models proposed in the literature, six metrics were used to evaluate the proposed models:

1. Squared Error (SE): is defined as $SE_i = (y_i - \hat{y}_i)^2$, where y_i is the i^{th} observed value, and $\hat{y}_i$ is the i^{th} estimate value.
2. Absolute Error (AE): defined as $AE_i = |y_i - \hat{y}_i|$. Although these two early metrics are scale-dependent, they are widely used in related literature (e.g., [15, 38]).
3. Percentage Error (PE): is a metric given by the ratio between the difference and defined as $PE_i = {}^{(y_i - \hat{y}_i)}/_{y_i}$ [37] . In this metric, positive and negative values can cancel each other, leading the average to approach to zero.
4. Absolute Percentage Error (APE): like PE, this is also a percentage error metric, except by using the absolute value $APE_i = |{}^{(y_i - \hat{y})}/_{y_i}|$ [37].
5. Absolute Scaled Error (ASE): this is a scale-free error metric, which is frequently used to measure accuracy [39]. ASE avoids the common problems in conventional accuracy metrics described previously. It is defined as $ASE_i = |y_i - \hat{y}_i|(\frac{1}{(n-1)} \sum_{j=2}^{n} |y_j - y_{j-1}|)^{-1}$, where n is the sample size. All the above mentioned metrics (1-5) metrics provide values closer to zero for better models and far from zero for worse ones. The PE metric can provide either negative and positive values, while the remainder metrics result only in positive values.
6. Last employed metric is $\mathrm{R}^2 = 1 - \sum_i^n (y_i - \hat{y}_i)^2 (\sum_i^n (y_i - \bar{y})^2)^{-1}$, which shows how well the estimated values produced by a model fit the actual ones. Results lie between 0 and 1, where 0 means a model does not provide any explanation about the data, and 1 refers to a perfect adjust.

5.2 Models Accuracy

Accuracy of each proposed model is evaluated applying the 10-fold cross validation method [40]. For each test, the estimated value for the power consumption is compared to the actual measured value. Table 2 presents the average and standard deviation for the six metrics. All models presented $R^2 > 91\%$. In particular, MLR models have low average errors for all metrics considering A1 and A2 architectures. However, when MLR is applied to fit the mix of architectures into a unique model, the error increases significantly. A similar effect occurs with RET models, whose accuracy is even worse than MLR models for the mix of architectures.

MLP models presented the best accuracy from the experiments. However, MLR are simpler and less costly models whose accuracy approach the MLP's accuracy. It suggests non-linear relations with low significance between the independent variables and the dependent one. This evidence is supported by the the average and standard deviation, which are close but not equal.

Noticeably, RET models present the worst accuracy from the three models. This can be explained as RET clusters data before estimating the power consumption. Indeed, power consumption cannot be explained for a small subset of dependent variables. However, all of the variables provide enough information for estimating power consumption, which hinders the clustering.

Figure 4 shows actual values compared to estimated ones generated with test set in each fold. Considering the results, we can conclude the MLP models provide better estimations for power consumption, while MLR are simpler models which present similar performance in terms of accuracy. Furthermore, experimental results also show that RET models do not provide accurate estimations for power consumption when compared to MLP and MLR models, mainly when dealing with mixed architectures in the same estimator.

6 Related Work

A large number of papers has been published on modeling computers power consumption, including some surveys [41–43]. Several models have been pro-

Table 2. Average and the standard deviation of the Squared Error (SE), Absolute Error (AE), Percentage Error (PE), Absolute Percentage Error (APE), Absolute Scaled Error (ASE), and R^2 metrics obtained by the 10-fold Cross-Validation.

		Average					**Standard Deviation**					
Arch.	**Method**	**SE**	**AE**	**PE**	**APE**	**ASE**	**SE**	**AE**	**PE**	**APE**	**ASE**	**R^2**
A1	*MLR*	7.2878	1.6858	-0.4008%	3.8598%	1.2426	56.1252	2.1064	5.6912%	4.1994%	1.5532	96.8650%
	RET	14.8066	2.8970	-0.7901%	6.8144%	2.1249	54.7590	2.5281	8.9543%	5.8610%	1.8534	93.6364%
	MLP	6.1053	1.4895	0.0332%	3.3382%	1.1040	56.3053	1.9594	5.1738%	3.9961%	1.4517	97.3777%
A2	*MLR*	4.9962	1.3446	-0.0078%	0.5332%	2.2536	21.4396	1.7845	0.8705%	0.6880%	2.9887	91.6575%
	RET	4.8958	1.3094	-0.0067%	0.5164%	2.2015	20.3679	1.7828	0.8573%	0.6844%	2.9973	91.8226%
	MLP	3.7707	1.1169	-0.0115%	0.4424%	1.8725	18.0375	1.5873	0.7533%	0.6115%	2.6572	93.7082%
Mix	*MLR*	14.0595	2.6209	-0.4336%	3.7864%	2.5888	53.7892	2.6807	5.9828%	4.6517%	2.6474	99.8654%
	RET	150.7207	7.7339	-4.1102%	11.7914%	7.6287	533.8658	9.5315	19.3902%	15.9341%	9.4011	98.5565%
	MLP	6.3264	1.5471	-0.3946%	2.3506%	1.5232	48.7559	1.9796	4.3027%	3.6313%	1.9479	99.9394%

posed to estimate the energy consumption of processors [12, 11, 44, 14]. Most of them consist of linear regression-based models which are fed with hardware performance counters. In [13], a model was proposed combining real total power measurement with hardware counters measurement to estimate per-component energy consumption. Our approach is different from those works because our goal is to estimate energy consumption for the entire machine, not limited to the processor.

Other papers address the modeling of the entire computer (e.g., from commodity computers to datacenter servers) proposing linear models composed by the summation of the energy consumed by its subcomponents [45–48]. For instance, Lewis et al. [45] propose an aggregated model which considers CPU, memory, electromechanical components, peripherals, and hard disk consumption. The models have coefficients for each component that are adjusted using linear regression. The energy consumed by virtual machines is also modeled in [47]. Other non linear models are also proposed for modeling the entire computer energy consumption [49, 50]. Our work is different as our objective is not to model energy consumption of the computer as an explicit summation of the consumption of its subcomponents. Instead, our models are fed with system variables carefully selected (by their ability to explain the model) in order to estimate energy consumption with high accuracy. Also, our work propose and compare models based on three different techniques.

As mentioned, regression models are numerous for modeling energy consumption. For instance, Piga [38] defined a global center-level approach to power and performance optimization for Web Server Datacenters. Their model is based on linear and non-linear regression techniques, while using the k-means to identify non-linear correlation and the Correlation-based Feature Selection (CFS) for removing independents variables that do not provide significant explanation for the power consumption. Our focus, instead is to model individual computers based on observable operating system measures.

Da Costa et al. [20] modeled computer energy consumption based on performance counters provided by two tools (Linux `pidstat` and `collectd`). The paper describes the methodology for reducing from a set of 165 explanatory variables to a small number of variables which can explain the model with high accuracy. The model is intended to estimate energy consumption at process level. Our work is different regarding the variety of techniques used, and modeling the whole machine energy consumption.

Comparing our models with the best related work results, considering the absolute percentage error, our best MLP specific and global models presented an error rate of 2.35%, and 0.44%, correspondingly, while [45, 38, 20, 51] have an error rate of $\sim 4.0\%$, $\sim 4.4\%$, $\sim 10.0\%$, and $\sim 6.0\%$, respectively.

By the best of our knowledge, our work provides the following novel contributions: (*i*) it proposes and compares three different models to estimate the power consumption for more than one hardware configuration; (*ii*) it employs the MIC method to analysis correlation between independent variables and the power consumption; (*iii*) the proposed models are fed with commodity system variables commonly provided by Linux, for better portability; (*iv*) it analysis accuracy using several metrics along with cross-validation in order to verify precision and overfitting issues; and (*v*) it overtakes accuracy of the state-of-the-art energy consumption models for datacenter nodes.

7 Conclusion

The management of power consumption of individual machines is a relevant feature in several environments, from small devices with limited resources to large datacenters with thousands of nodes. In this we present a characterization of energy consumption of entire machines based on resources utilization variables. Experiments were carried out using synthetic workloads in order to discover what resources and modeling methods present higher correlation to energy consumption. We show that it is possible to estimate energy consumption by sampling variables provided by common operating systems and employing MLR, RET, or MLP methods. We proposed nine models that provide accurate estimation on energy consumption with an accuracy of 99.9%, and average squared error of 6.32 watts with standard deviation of 48.76.

All models evaluated can be fully implemented in software, providing a cost-effective mechanism for estimating energy consumption. Such models can be deployed in a wide range of devices, from single small devices with limited resources to thousands of machines in a large datacenter at no aditional cost and negligible overhead. Our proposal can be useful for several aims, e.g., to provide instant information on energy consumption in a per machine basis.

List of Abreviations

ANN: Artificial Neural Network; CFS: Correlation-based Feature Selection; ICT: Infor-mation and Communication Technologies; MIC: Maximal Information Coefficient; MLP: Multilayer Perceptron; MLR: Multiple Linear Regression; RET: Regression Tree.

Acknowledgement

Authors thank CAPES and RNP for partially supporting this research. Hermes Senger thanks CNPq (Contract Number 305032/2015-1) and FAPESP (Process numbers 2018/00452-2, and 2018/22979-2) for their support.

References

1. Gartner: Gartner energy & utilities it summit 2007: an invitation only event. Technical report, Gartner, Dallas, EUA (September 2007)

2. Cook, G.: How clear is your cloud? catalysing an energy revolution. Technical report, Greenpeace International, Amsterdam, The Netherlands (April 2012). `http://goo.gl/yd1FAS`
3. Gao, P.X., Curtis, A.R., Wong, B., Keshav, S.: It's not easy being green. In: Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. SIGCOMM '12, pp. 211–222. ACM, New York, NY, USA (2012)
4. Koomey, J.: Growth in Data Center Electricity Use 2005 to 2010 vol. 1, p. 2010. Analytics Press, Oakland, CA (2011)
5. Orgerie, A.-C., Assunção, M., Lefèvre, L.: Energy aware clouds. In: Cafaro, M., Aloisio, G. (eds.) Grids, Clouds and Virtualization. Computer Communications and Networks, pp. 143–166. Springer, ??? (2011)
6. Lee, Y., Zomaya, A.: Energy efficient utilization of resources in cloud computing systems. The Journal of Supercomputing **60**, 268–280 (2012)
7. Song, Y., Wang, H., Li, Y., Feng, B., Sun, Y.: Multi-tiered on-demand resource scheduling for vm-based data center. In: CCGRID 0́9., pp. 148–155 (2009)
8. Barroso, L.A., Hölzle, U.: The case for energy-proportional computing. Computer **40**(12), 33–37 (2007)
9. Song, S., Su, C., Rountree, B., Cameron, K.W.: A simplified and accurate model of power-performance efficiency on emergent gpu architectures. In: Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium On, pp. 673–686 (2013). IEEE
10. Naveh, A., Rajwan, D., Ananthakrishnan, A., Weissmann, E.: Power management architecture of the 2nd generation intel®core™microarchitecture, formerly codenamed sandy bridge. In: Hot Chips (2011)
11. Contreras, G., Martonosi, M.: Power prediction for intel xscale® processors using performance monitoring unit events. In: Low Power Electronics and Design, 2005. ISLPED'05. Proceedings of the 2005 International Symposium On, pp. 221–226 (2005). IEEE
12. Joseph, R., Martonosi, M.: Run-time power estimation in high performance microprocessors. In: Proceedings of the 2001 International Symposium on Low Power Electronics and Design, pp. 135–140 (2001). ACM
13. Isci, C., Martonosi, M.: Runtime power monitoring in high-end processors: methodology and empirical data. In: Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture, p. 93 (2003). IEEE Computer Society
14. Bertran, R., Gonzalez, M., Martorell, X., Navarro, N., Ayguade, E.: Decomposable and responsive power models for multicore processors using performance counters. In: Proceedings of the 24th ACM International Conference on Supercomputing, pp. 147–158 (2010). ACM
15. Venezian Povoa, L., Bignatto Junior, P.W., Monteiro, C.E., Mueller, D., Marcondes, C.A.C., Senger, H.: A model for estimating energy consumption based on resources utilization. In: IEEE Symposium on Computers and Communications (ISCC), pp. 1–6 (2013). IEEE Computer Society
16. Dusso, P.M.: A monitoring system for wattdb: An energy-proportional database cluster. Graduation thesis, Informatics Institute, Federal University of Rio Grande do Sul, Porto Alegre, Brazil (July 2012). `http://goo.gl/8gheFW`
17. Dusso, P.M.: energyagent (2012). `https://github.com/pmdusso/energyagent`
18. Reshef, D.N., Reshef, Y.A., Finucane, H.K., Grossman, S.R., McVean, G., Turnbaugh, P.J., Lander, E.S., Mitzenmacher, M., Sabeti, P.C.: Detecting Novel Associations in Large Data Sets. Science **334**(6062), 1518–1524 (2011)

19. Pazzani, M.J., Bay, S.D.: The independent sign bias: Gaining insight from multiple linear regression. In: Proceedings of the Twenty-First Annual Meeting of the Cognitive Science Society, pp. 525–530 (1999)
20. Da Costa, G., Hlavacs, H.: Methodology of measurement for energy consumption of applications. In: 2010 11th IEEE/ACM International Conference on Grid Computing, pp. 290–297 (2010). IEEE
21. Waterland, A.: stress (2014). http://people.seas.harvard.edu/\texttildelowapw/stress/
22. Marletta, A.: cpulimit (2012). https://github.com/opsengine/cpulimit
23. NLANR/DAST: Iperf - The TCP/UDP Bandwidth Measurement Tool (2011). https://iperf.fr/
24. Yokogawa Electric Corporation: 0.5 Class Transducer for Power Application. (2009). http://goo.gl/f43ZwV
25. Yokogawa Electric Corporation: Data Acquisition Unit MW100. (2013). http://goo.gl/7mcNV8
26. Marsaglia, G., Tsang, W.W., Wang, J.: Evaluating kolmogorov's distribution. Journal of Statistical Software **8**(18), 1–4 (2003)
27. Rice, J.R.: Mathematical Statistics and Data Analysis, 3rd edn. Thomson Books/Cole, United States of America (2007)
28. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Chapman and Hall, New York, USA (1984)
29. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall, ??? (1998)
30. Semwal, V.B., Raj, M., Nandi, G.: Biometric gait identification based on a multilayer perceptron. Robotics and Autonomous Systems **65**, 65–75 (2015). doi:10.1016/j.robot.2014.11.010
31. De Lozzo, M., Klotz, P., Laurent, B.: Multilayer perceptron for the learning of spatio-temporal dynamics—application in thermal engineering. Engineering Applications of Artificial Intelligence **26**(10), 2270–2286 (2013). doi:10.1016/j.engappai.2013.07.001
32. Altunkaynak, A.: Prediction of significant wave height using geno-multilayer perceptron. Ocean Engineering **58**, 144–153 (2013). doi:10.1016/j.oceaneng.2012.08.005
33. Velo, R., López, P., Maseda, F.: Wind speed estimation using multilayer perceptron. Energy Conversion and Management **81**, 1–9 (2014). doi:10.1016/j.enconman.2014.02.017
34. Li, M., Zhang, T., Chen, Y., Smola, A.J.: Efficient mini-batch training for stochastic optimization. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '14, pp. 661–670. ACM, New York, NY, USA (2014). doi:10.1145/2623330.2623612
35. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, ??? (2016). http://www.deeplearningbook.org
36. Ng, W.W.Y., Zeng, G., Zhang, J., Yeung, D.S., Pedrycz, W.: Dual autoencoders features for imbalance classification problem. Pattern Recognition **60**, 875–889 (2016). doi:10.1016/j.patcog.2016.06.013
37. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. International Journal of Forecasting, 679–688 (2006)
38. de Paula Rosa Piga, L.: Modeling, characterization, and optimization of web server power in data centers. PhD thesis, Institute of Computing (IC), University of Campinas (UNICAMP), Campinas, Brazil (November 2013)

39. Hyndman, R.J.: Another look at forecast accuracy metrics for intermittent demand. Foresight: The International Journal of Applied Forecasting (4), 43–46 (2006)
40. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2. IJCAI'95, pp. 1137–1143. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)
41. Reda, S., Nowroz, A.N.: Power modeling and characterization of computing devices: a survey. Foundations and Trends in Electronic Design Automation **6**(2), 121–216 (2012)
42. Orgerie, A.-C., Assuncao, M.D.d., Lefevre, L.: A survey on techniques for improving the energy efficiency of large-scale distributed systems. ACM Computing Surveys (CSUR) **46**(4), 47 (2014)
43. Dayarathna, M., Wen, Y., Fan, R.: Data center energy consumption modeling: A survey. IEEE Communications Surveys Tutorials **18**(1), 732–794 (2016). doi:10.1109/COMST.2015.2481183
44. Bertran, R., Gonzalez, M., Martorell, X., Navarro, N., Ayguade, E.: A systematic methodology to generate decomposable and responsive power models for cmps. IEEE Transactions on Computers **62**(7), 1289–1302 (2013)
45. Lewis, A.W., Ghosh, S., Tzeng, N.-F.: Run-time energy consumption estimation based on workload in server systems. HotPower **8**, 17–21 (2008)
46. Orgerie, A.-C., Lefèvre, L., Guérin-Lassous, I.: Energy-efficient bandwidth reservation for bulk data transfers in dedicated wired networks. The Journal of Supercomputing **62**(3), 1139–1166 (2012)
47. Xiao, P., Hu, Z., Liu, D., Yan, G., Qu, X.: Virtual machine power measuring technique with bounded error in cloud environments. Journal of Network and Computer Applications **36**(2), 818–828 (2013)
48. Lent, R.: A model for network server performance and power consumption. Sustainable Computing: Informatics and Systems **3**(2), 80–93 (2013)
49. Fan, X., Weber, W.-D., Barroso, L.A.: Power provisioning for a warehouse-sized computer. In: ACM SIGARCH Computer Architecture News, pp. 13–23 (2007). ACM
50. Tang, C.-J., Dai, M.-R.: Dynamic computing resource adjustment for enhancing energy efficiency of cloud service data centers. In: System Integration (SII), 2011 IEEE/SICE International Symposium On, pp. 1159–1164 (2011). IEEE
51. Dhiman, G., Mihic, K., Rosing, T.: A system for online power prediction in virtualized environments using gaussian mixture models. In: Design Automation Conference (DAC), 2010 47th ACM/IEEE, pp. 807–812 (2010). doi:10.1145/1837274.1837478

View publication stats