UNIVERSITY OF AMSTERDAM

MASTER THESIS

Calculating the Energy Consumption of a Website

August 7, 2017

Anouk Boukema
anouk.boukema@os3.nl

Supervisor
Maarten de Waard
maarten@greenhost.nl

Abstract

With the globally increasing environmental concerns and at the same time the increasing amount of active websites, the importance of calculating the total energy used by a website becomes prominent. Within this research a step by step guideline on how this might be done on real-world data, based on the findings in related work is provided. The proposed solution is tested and validated within a proof of concept. The accuracy found during validation is not yet high enough to adequately predict the power consumption of a website within the proof of concept. This might be caused by shortcomings in the data. However, if the shortcomings are resolved or the proposed solution is tested on different data the prediction models used can be further improved to contribute to a more aware and knowledgeable future.

1 Introduction

With the growing connectedness of people and things the footprint of Information and Communication Technology (ICT) systems is responsible for the same amount of CO_2 emissions as global air travel. If this growth continues at the present pace, the energy consumption by ICT systems will endanger ambitious plans to reduce CO_2 emissions and tackle climate change [1]. Cisco's Visual Networking Index forecast 2015-2020 predicts that such a growth is bound to happen with the global IP traffic increasing nearly threefold over the next 5 years [2]. Netcraft's monthly web server survey showed that there are almost 170 million active sites in the month May 2017 [3]. These three reports indicate the importance of raising awareness on the total energy usage by a website. The motivation for this research is raising this awareness and providing a guideline on how to estimate a website's power consumption with real-world data. These two goals will be pursued by answering the following research question: *"How to calculate the energy consumption of a website?"*.

Due to the broadness of the question it will be divided in three sub-questions. The first: *"What are the energy using components of a website?"* and second *"What are valuable resource measures for calculating the energy consumption of a website component?"*. These will both be answered in the related work section 2. To answer the third sub-question *"What are the relationships between the measurable resources of a website component and the power it consumes?"* a proof of concept is conducted. The architectural setup, methodology and experimental setup used will be explained in Sections 3, 4 and 5 respectively. After which it is possible to acquire the relations for the measures done within the proof of concept in Section 6 therefore answering the third sub-question. The answer to the main questions will be given in the conclusion followed by a summary of limitations on the proof of concept conducted in the discussion. Ending with recommendations on how to improve the proof of concept in the future work section.

2 Related Work & Background

A website conforms to the client-server computing model, where the client is a web browser requesting resources of a web server [4, Chapter 19].

Because of the more dynamic, interactive and divers characteristics of websites nowadays more often the word "web-application" is used. Web-applications are logically built up out of separate layers concerned with the logical division of components and their functionality. At the highest and most abstract level any application consists of a presentation, business and data layer which all reside at the server side [4, Chapter 5]. The presentation layer interacts with the user (client side) and the business layer, the business layer then interacts with the data layer and possible other external systems as can be seen in figure 1.

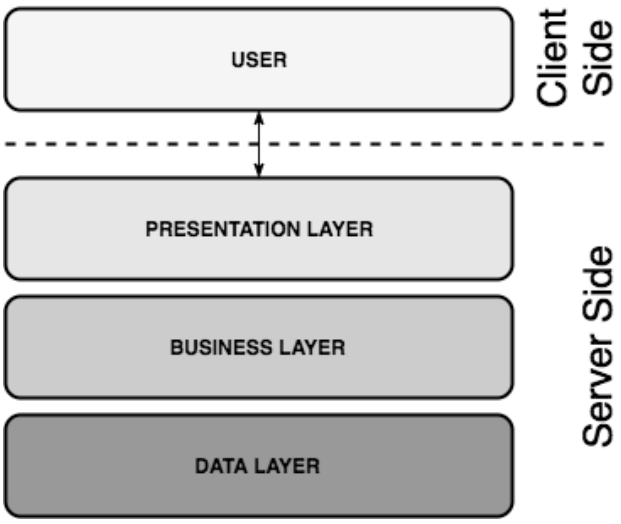

Figure 1: General Application Layers

To answer the first sub-question *"What are the energy using components of a website?"* one should consider all three server side layers and the user layer as relevant components of a website. Furthermore the energy used by the network to transfer the resources between client and server should also be taken into account.

Each layer can be presented by specialized software. A common used open source archetypal model is the LAMP stack. Where all servers run on Linux, the presentation layer consists of Apache, the Business logic layer of PHP and the data layer of MySQL. Nowadays Apache is having almost half the market share of all active sites nowadays [3].

The energy used by a website will then be a summation of the energy consumption of each layer his processes, the network usage, the client site processes and partly the idle state of each server running specialized processes for the website.

Because servers are not equipped with sensors that measure the energy usage per software process, a translation needs to be made from the measurable resources to the power usage that is measured by sensors. Several studies show that there is a causality between the measurable CPU, RAM, memory (disk) and NIC utility of a process and the power overall usage [5], [6], [7]. These measures are then the answer to the second sub-question *"What are valuable resource measures for calculating the energy consumption of a website component?"*.

In what way these measurements relate to the energy usage is platform dependent. Within this project the relationship between the CPU and disk measurements of an Apache process of a website will be researched.

This can be done in a comparable manner as suggested by the paper "Profiling Energy Usage for Ef-

ficient Consumption" [7], where the idle and stressed energy usage of hardware components by their manufactures' website or simple monitoring devices is taken as a base. With this information it is possible to find the general energy consumption of an applications hardware usage. Giving them a ranking system for determining which components of the application can be optimized to realize the largest cost savings. This however does not give any time related energy usage of a website, only estimates.

The other two papers [5] and [6] have a different approach. They calculate the energy usage of an application or VM based on the correlation of its measured resources and the overall power usage of the underlying hardware over time. This correlation is found using either linear or polynomial regression models. Where polynomial is a better model for servers using the AMD Turbo Core.

3 System Architecture

The work presented in this paper will focus solely on energy consumption of the presentation and business layer. This because the presentation layer is the only mandatory server side layer needed to generate a simple website. Also it is the front-end of a website/web-application and resides between the user and the resources it requests, meaning all the information flows through or ends at this layer and is therefore a good initial indicator of the workload of a website. The business logic layer does the computations needed to generate the resources requested. The amount of computations needed can vary greatly dependent on the request. Therefore the business logic layer is a good addition to indicate the difference in energy usage per website.

Within this paper the energy consumption of the Apache processes and PHP scripts (presentation and business layer) of websites hosted by the webhosting company Greenhost are researched. They are located on the same server and are separated from the data layers servers as shown in Figure 2.

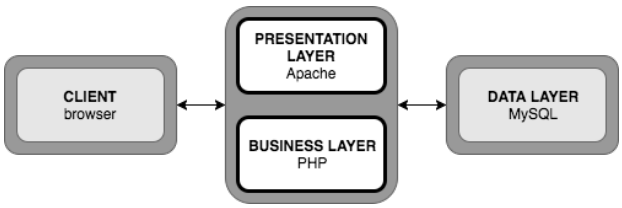

Figure 2: General Architecture Greenhost

The Apache processes and PHP scripts of a single website run in a closed environment called a hosting package. Hosting packages are the only isolated environments running on the Xen VMs called Hosting Nodes. For redundancy and scalability reasons one package can run on multiple Hosting Nodes. There is one other type of VM running on the servers: the Virtual Private Servers (VPS). Together they form the Virtualization layer. VPSs are regarding their setup different from the hosting nodes and identical among each-other. Also the hosting packages are identical amongst each-other considering their setup as are the hosting nodes. In total there are 1862 hosting packages, 48 hosting node VMs, 370 VPS VMs, and 12 servers of model Intel® Xeon® CPU E5-2630 v3, without Turbo Core. See Figure 3 for clarification.

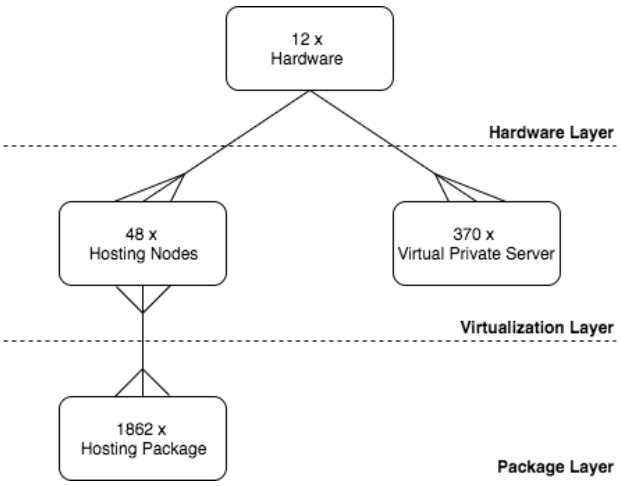

Figure 3: Greenhost Server Architecture

The resources used by each environment are measured and then stored in Round Robin Database (RRD) files. These files contain multiple Round Robin Archives (RRA), which are circular buffer based archives. Each RRA contains a fixed amount of entries that are filled with data obtained in a fixed timely interval, for example every 5 minutes. The data in the entries is usually interpolated by RRD [8]. Which resources are measured, how they are gathered and in which unit is shown in Table 1.

4 Methodology

As shown in Table 1, the power measurements are only done at the hardware layer (Phw). However, in order to answer the research question, the power usage of one hosting package needs to be acquired (Ppk). Because the internal setup of the servers within this research is different from the research by Aman Kansal et al. [5] and Ingolf Waßmann et al. [6], the relationship between CPU (CPU) and memory (MEM) with power (P) needs to be found for this setup. Because the packages are running on VMs and are not the only environments using the physical CPU and memory, the relationship between all these layers (as shown in Figure 3) need to be examined. This will reveal how much overhead is added by going from one layer to the other. Then these relationships can be combined to find the relationship between the CPU

Environment	Data	Unit	Acquired Via
Hardware	ower CPU Memory	Wattage percentage bytes	ipmitool
Hosting nodes	CPU	seconds	xentop
Virtual Private Server	CPU	seconds	xentop
Hosting Packages	CPU Memory	seconds bytes	cpuacct.usage memory.max_usage_in_bytes

Table 1: All .rrd files of Greenhost

of a package and the power it uses. How these relations mathematically relate, is explained within this section. In Section 5, the experimental setup to validate this mathematical relation is described, after which in Section 6 the parameters described in this section will be acquired and validated.

4.1 Package to Virtualization Layer

Because the only processes run on a hosting node are the hosting packages, the hypothesis is made that *"the sum of the CPU seconds measured at a certain time of all the packages running on a hosting node, is almost equal to the CPU seconds measured at that hosting node."* Which can be written as the following equation, where $CPUhn_i$ denotes the CPU measures of a hosting node $_i$, and $\sum_{j=1}^{p} CPUpk_{j,i}$ denotes the sum of the CPU measures of all hosting packages $_j$ with $j = 1..p$ and p the number of packages on the hosting node i:

$$CPUhn_i = \alpha \sum_{j=1}^{p} CPUpk_{j,i} + \beta \tag{1}$$

The parameters α and β for this equation will be acquired by using linear regression. Then the hypothesis will be validated in Section 6.1.

4.2 Virtualization to Hardware Layer

Because the only processes run on the hardware are the hosting node VMs and the virtual private server VMs the hypothesis is made *"that the sum of all CPU seconds on specific time of the Virtualization layer (hosting nodes + virtual private servers) on a server must relate in a linear way to the CPU seconds measured by the hardware of that same server"*. Which can be written as the following equation, where $CPUhw_k$ denotes the CPU measures of a server $_k$, $\sum_{i=1}^{h} CPUhn_i$ denotes the sum of the CPU measures of all hosting nodes $_i$ with $i = 1..h$ and h is the number of hosting nodes on on server $_k$. $CPUvps_l$ denotes the sum of the CPU measures of all VPS $_i$ with $l = 1..v$ and v is the number of VPSs on that server $_k$:

$$CPUhw_k = \gamma(\sum_{i=1}^{h} CPUhn_{i,k} + \sum_{l=1}^{v} CPUvps_{l,k}) + \delta$$

With the data currently available there is no way to research this hypothesis because the data does not indicate which hosting node or virtual private server runs on which hardware node. Therefore the hypothesis is generalized to *"the sum of all CPU seconds on specific time of all Virtualization Layers (hosting nodes + virtual private servers) must relate in a linear way to the CPU seconds measured by the hardware of all servers"*. Which can be written as the following equation, where $\sum_{k=1}^{w} CPUhw_k$ denotes the sum of the CPU measures of all hardware nodes/servers $_k$ with $k = 1..w$ and w as the total number of servers:

$$\sum_{k=1}^{w} CPUhw_k = \gamma(\sum_{i=1}^{h} CPUhn_i + \sum_{l=1}^{v} CPUvps_l) + \delta \tag{2}$$

The parameters γ and δ for this equation will be acquired by using linear regression. Then the hypothesis will be validated in Section 6.2.

4.3 Hardware: CPU to Power

Because (as stated in Section 2) the relation between CPU and power usage is either linear or polynomial and (as stated in Section 3) the servers used within this project do not use a Turbo Core, a linear model is a proven possible predictor model for this setup. Therefore the following hypothesis can be researched: *"the power used by one server multi-linearly relates to the CPU measured at that same server"*. Which can be written as the following equation, where Phw_k, $CPUhw_k$ and $MEMhw_k$ denote the Power, CPU and Memory measures of a server $_k$ respectively:

$$Phw_k = \epsilon CPUhw_k + \zeta MEMhw_k + \eta \tag{3}$$

The parameters ϵ, ζ and η for this equation will be acquired by using linear regression. Then the hypothesis will be validated in Section 6.3.

4.4 Overall Power Usage

To get from the CPU seconds measured by the packages to the Power, the 3 formulas found in the above subsections need to be combined. Because equation 2 only accounts for the sum of all the hardware nodes together, the other formula's have to be written in the same format, and therefore the equations 1 and 3 need to be re-formulated.

Multiform of equation 1

$$\sum_{i=1}^{h} CPUhn_i = \sum_{i=1}^{h} (\alpha \sum_{j=1}^{p} CPUpk_{j,i} + \beta)$$

Since all packages are identical among each-other, the parameter α is the same for each package and the sum of all the packages is equal to the sum of the sum of all the packages on each hosting node. Resulting in the following equation:

$$\sum_{i=1}^{h} CPUhn_i = \alpha \sum_{j=1}^{p} CPUpk_j + p\beta$$

Multiform of equation 3
Since all servers are identical among each-other the parameter η is as well.

$$\sum_{k=1}^{w} Phw_k = \epsilon \sum_{k=1}^{w} CPUhw_k + \zeta \sum_{k=1}^{w} MEMhw_k + w\eta$$

Multiform packages: CPU to Power
Combining the two multiform equations of equation 1, 2 and 3 becomes:

$$\sum_{k=1}^{w} Phw_k = \epsilon\gamma\alpha \sum_{j=1}^{p} CPUpk_j + \epsilon\gamma \sum_{l=1}^{v} CPUvps_l + f \sum_{k=1}^{w} MEMhw_k + \epsilon\gamma p\beta + \epsilon\delta + w\eta$$

Since all the constants at the end represent the idle power of all the servers, this will be denoted as constant z. For clarity reasons the coeficients used can be substituted for one letter:

$$\sum_{k=1}^{w} Phw_k = x \sum_{j=1}^{p} CPUpk_j + a \sum_{l=1}^{v} CPUvps_l + y \sum_{k=1}^{w} MEMhw_k + z \quad (4)$$

The parameters x, a, y and z for this equation will be acquired by combining the parameters found sections 6.1, 6.2 and 6.3. Then this equation will be validated in Section 6.4.

4.5 Package Power Usage

When the power used by all packages is to be calculated the CPU usage of the virtual private servers should be excluded and therefore set to zero. Since the only processes running on the hardware are the hosting nodes (sum of all the packages) and the virtual private servers, the idle power should be fairly divided over those. Meaning that only the percentage of all hosting nodes of the total amount of VMs should be taken into account as idle power. With the assumption that one hosting node uses averagely the same amount of power as a virtual private server this comes down to $\frac{h}{h+v}$ % of the idle power used by all the hosting nodes and therefore the packages:

$$\sum_{j=1}^{p} Ppk_j = x \sum_{j=1} pCPUpk_j + y \sum_{k=1}^{w} MEMhw_k + \frac{h}{h+v} z$$

Because only the relationship between the CPU of the packages and the CPU of the hardware is researched, an hypothesis has to be made about the relationship between memory of the packages and memory of the hardware. The hypothesis is that *"the memory used by all the packages is equal to the memory used by all the hosting nodes, and the memory used by all hosting nodes is equal to the memory used by all the servers minus the memory used by all the virtual private servers"*. Which can be mathematically formulated as:

$$\sum_{j=1}^{p} MEMpk_j = \sum_{i=1}^{h} MEMhn_i$$

$$\sum_{i=1}^{h} MEMhn_i = \sum_{k=1}^{w} MEMhw_k - \sum_{l=1}^{v} MEMvps_l$$

Since the memory of the VPSs is to be thought zero, the equation is as follows:

$$\sum_{j=1}^{p} Ppk_j = x \sum_{j=1} pCPUpk_j + y \sum_{j=1}^{p} MEMpk_j + \frac{h}{h+v} z$$

Every variable is now set to measurements from the package layer, therefore the equation can be transformed to its single form for a package. This means that the idle usage should now be divided by the total amount of packages:

$$Ppk_j = xCPUpk_j + yMEMpk_j + \frac{\frac{h}{h+v}}{p} z \quad (5)$$

The equations 1, 2, 3, 4 and 5 are the answers to the last sub-question: *"What are the relationships between the measurable resources of a website component and the power it consumes?"*.

5 Experimental Setup

The parameters for the equations 1, 2 and 3 addressed in the previous section will be acquired by using linear regression on a training set gathered by Greenhost. To validate the equations, they will predict measures using a test set for the input variables. These predicted measures will be validated against the true measures. In order to do so the data needs to be pre-processed. Because the equations are dependent upon each-other the data used for each equation should be of the same lengths covering the same time interval. The interval used for this research is from 2017-06-30 00:30 until 2017-07-02 21:00. Because the relations are based on the assumption that the data varies over time meanwhile the correlation stays the same, the time interval on which the data is acquired should be as small as possible. Resulting in bigger variation ranges and therefore possibly clearer correlations. The step size therefore is every 5 minutes, which was the minimal interval available. The data used from each resource has a size of 822 ordered values, because this is the maximum amount of entries in the 5 minute RRA. While pre-processing, the data

within some of the RRAs with empty values are removed, leaving the total amount of usable values in a dataset to 776. The data sets are split into a training and a test set of 80% and 20% of the total data set respectively. The following subsections describe how the data is the pre-processed so they adhere to the above mentioned requirements.

5.1 Package to Virtualization Layer

Because there are 48 hosting nodes for which the hypothesized relation will be tested and the relation should be the same among all, the data collected by each hosting node will be bundled together. Therefore one big pool of data can be used to find the parameters for the general hosting node described in equation 1. This means a total of $776 \times 48 = 37.248$ data points. Where $0.8 \times 37.248 = 29.798$ data points are reserved for the training data set and $37.248 - 29.798 = 7450$ data points for the test set.

5.2 Virtualization to Hardware Layer

Equation 2 requires the sum of a data point over all the hosting node and VPSs. Therefore the total amount of usable data points to train the linear regression model is $0.8 \times 776 = 620$ and for the validation phase $776 - 620 = 152$.

5.3 Hardware: CPU to Power

Because there are 12 hardware nodes for which the hypothesized relation mentioned in Section 4.3 will be tested and the relation should be the same among all, the data collected by each server will be bundled together. Therefore one big pool of data can be used to find the parameters for the general server described in equation 3. This means a total of $776 \times 12 = 9.312$ data points. Where $0.8 \times 9.312 = 7.450$ data points for the training data set and $9.312 - 7.450 = 1862$ data points for the test sets.

5.4 Overall Power Usage

The parameters found in the previous three sections can now be combined as proposed in Section 4.4 to generate the parameters needed for equation 4. To validate this equation measurements of CPU packages, CPU VPS and memory of the hardware are needed to generate a estimate on the overall power usage. Which then can be compared with the actual power usage measured at these same data points. Because the parameters are obtained via the other equations the test set contains all 776 data points.

5.5 Package Power Usage

The parameters found in the previous section can now be combined as proposed in Section 4.5 to generate the parameters needed for equation 5. To give an estimate on the minimal, average and maximum energy usage of a package using equation 5 two data sets are required: one containing all the CPU measures and the other the memory measures done per packages per time step in the interval used throughout the whole research, meaning a total of $1862 \times 776 = 1.444.912$ per data set. Of these data sets the minimum, maximum and average measurements will be used.

6 Results & Observations

With the equations from Section 4 and the data sets mentioned in Section 5, the parameters for the equations will be acquired and validated. Finally an estimate on the energy usage of a package will be given within this section.

6.1 Package to Virtualization Layer

Results The parameters mentioned in Section 4.1 are found using linear regression on the training set mentioned in Section 5.1. The found equation can then estimate the total CPU seconds of the hosting nodes by the independent variable CPU seconds of all the packages resided on these hosting nodes:

$$CPUhn_i = 0.97 \times \sum_{j=1}^{1862} CPUpk_j + 0.057$$

Both the training data and the linear regression line are shown in figure 4.

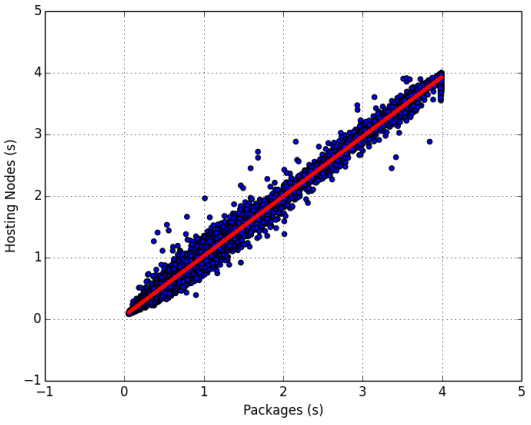

Figure 4: CPU seconds measured at the packages resided on one hosting node against the CPU seconds of that hosting nodes at certain time steps

Validation The accuracy of the formula is tested by calculating the mean squared error between the test set CPU measures of the hosting nodes and the corresponding prediction via the equation on the test set of the packages on those hosting nodes. This resulted in a mean squared error of 0.0056.

Observation The coefficient is almost one, and the constant almost zero, together representing just a little overhead. Meaning the assumption *"that the sum of the CPU seconds measured at a certain time of all the packages running on a hosting node, is almost equal to the CPU seconds measured at that hosting node"* is correct.

6.2 Virtualization to Hardware Layer

Results The linear regression model is used on the training set mentioned in Section 5.2 and generated the following parameters for equation 2. This equation can now estimate the total CPU percentages of the hardware layer by the independent variable CPU seconds of all the hosting nodes and virtual private machines:

$$\sum_{k=1}^{12} CPUhw_k = 2.82 \times (\sum_{i=1}^{48} CPUhn_i + \sum_{l=1}^{370} CPUvps_l) + 219.81$$

Both the training data and the linear line are shown in Figure 5.

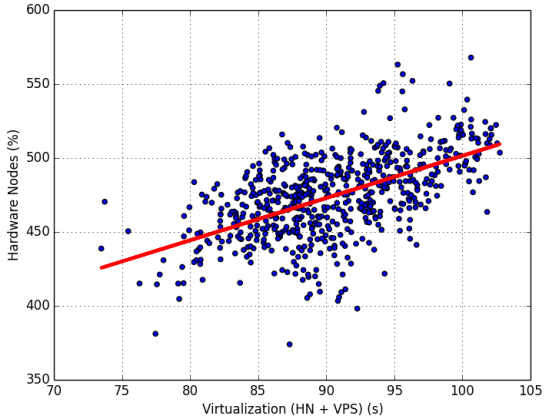

Figure 5: CPU seconds measured by all hosting nodes + virtual private servers against CPU percentages of all Hardware nodes at certain time steps.

Validation The accuracy of the equation is tested by calculating the mean squared error between the CPU hardware node test set and the corresponding prediction made by the equation on the vps and hosting node test set. This resulted in a mean squared error of 465.80. Which indicates an accuracy of about 21 CPU percentages.

Observation The data points do not adhere to such a strong correlation as they did with the hosting node and their packages. A possible explanation might be that hardware nodes might be busy processing incoming requests which are not handled yet by the VMs. Another reason might be that the hosting nodes and VPSs might measure to use 100% of their CPU, but instead the server gives them just a part of the real CPU.

6.3 Hardware: CPU to Power

Results The linear regression is done on the training set and generated the following parameters to estimate the Power in Wattage of a server by the independent variable CPU percentage of that same server:

$$Phw_k = 0.27 \times CPUhw_k + 132.97$$

Both the training data and the linear line are shown in figure 6.

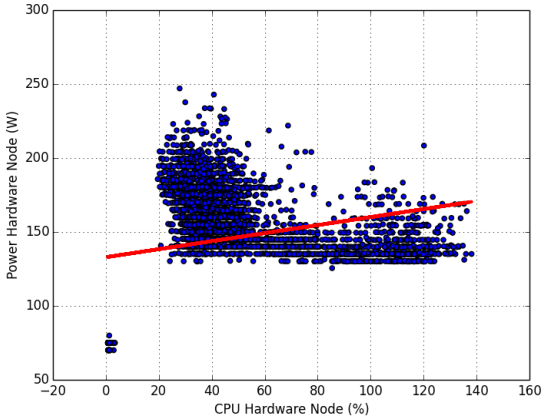

Figure 6: CPU percentage of a hardware node against Power measured in Watt of that same hardware node at certain time steps.

Validation The accuracy of the formula is tested by calculating the mean squared error between the CPU wattage test set and the corresponding prediction made by the formula on the CPU percentage test set. This resulted in a mean squared error of 987. Which indicates an accuracy of about 31 Watt.

Observation The data points do not seem to be predictable via the equation because they do not adhere to a pattern of a diagonal line, as was the case in the previous sections. Instead there is a big cluster indicating a lot of power usage on little CPU percentage. Also the coefficient is low indicating that a variation on the independent variable, CPU percentage, does have little influence on the dependent variable power. This might be caused by the absence of other resources like RAM and memory (disk) in the equation, because both also have impact on the power as mentioned in Section 2. Furthermore there is a small cluster using little power and little CPU, this cluster seemed to consists of 11 percent of the total trained data, and does not contain any empty or zero values. Therefore, they are probably no outliers but real data.

Since memory (disk utility) could have influence on the power usage and is measured at the server,

its influence will be tested by adding memory as an independent variable to the equation as presented in Section 4.3. The linear regression model is trained on the training data of both CPU percentages as the memory of the hardware layer, the following parameters where found:

Results

$$Phw_k = 0.32 \times CPUhw_k + 3.2 \times MEMhw_k + 87.34$$

To visualize the extra dimension, memory will be indicated by a color scale, as shown in figure 7.

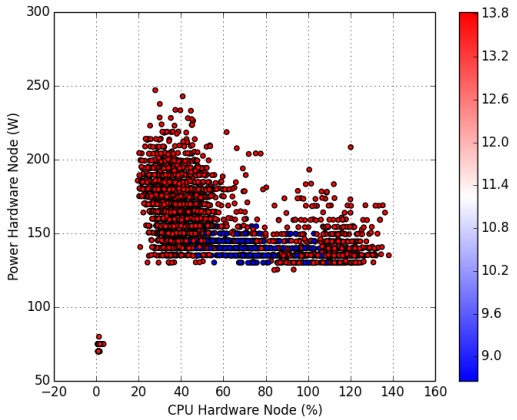

Figure 7: CPU percentage of a hardware node on x-as, the memory in MB in color against Power measured in Watt of that same hosting nodes at certain time steps.

Validation The accuracy of the formula is tested by calculating the mean squared error between the test set and the corresponding prediction made by the equation on the test set. This resulted in a mean squared error of 907. Which indicates an accuracy of about 30 Watt.

Observation The accuracy is now lower then before. Comparing these coefficients can not be done because they are multiplied against different and non normalized types of data. The coefficient of CPU is a bit higher then before, but still low, indicating little influence from CPU to power. The reason for this could be due to the lack variation in CPU usage of the machines. The maximum possible CPU usage percentage is $16 \times 100 = 19.200$ where the measured values fall in a range of 20 to 140 percentages (see Figure 7) which is 1 to 9 % of the total possible CPU percentage of a server. Creating a reliable linear model on this small range is difficult. Adding memory does not seem to give an explanation to the small cluster on the left bottom corner, nor the bigger cluster. However it increased the accuracy, and therefore will be kept into the equation. The results of this validation show that this model is not optimized enough to do accurate predictions. Nonetheless, the goal of this research is not only to give predictions on the energy usage of a website but also to give a guideline on how to do so. Therefore the equations and parameters found within this proof of concept will be used to generate a final prediction on the power usage of a package in the following two sections.

6.4 Overall Power Usage

To find the parameters of equation 4, the parameters found in the previous sections will be included as described in Section 5.4.

$$\sum_{k=1}^{12} Phw_k = 0.86 \times \sum_{j=1}^{1862} CPUpk_j + 0.90 \times \sum_{l=1}^{370} CPUvps_l + 3.3 \times \sum_{k=1}^{12} MEM_k + 1118.6$$

This equation is used to predict the overall power used by all servers by inserting the test set data of CPU packages, CPU vps and memory of the hardware. This predicted power will then be compared to the real data measured under the same time interval. This resulted in the following plot, see Figure 8.

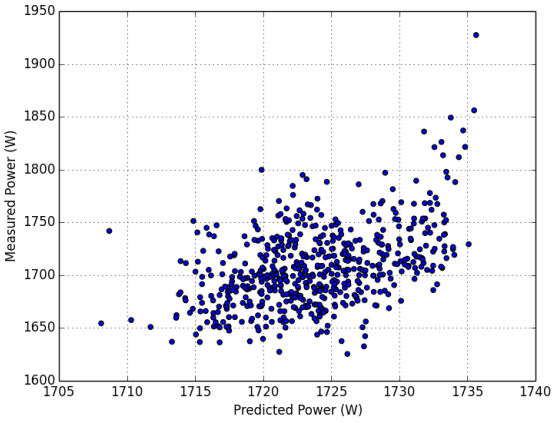

Figure 8: Total prediction power in Watt against measured total power of all the hardware nodes at certain time steps.

Validation The accuracy of the formula is tested by calculating the mean squared error between the test set and the corresponding prediction via the equation on the test set. This resulted in a mean squared error of 1595. Which indicates an accuracy of about 40 Watt.

Observation The equation to estimate the overall power used by all servers only predict values between the 1700 Watt and 1750 Watt. This is a much smaller range than the actual power usage measured within

the same interval. This indicates that the parameters found are not yet optimal, but can predict power within the actual range it was measured.

6.5 Package Power Usage

To estimate the minimal, average and maximum energy used by a package the parameters found in the previous section are used to find the parameters for equation 5 as described in Section 4.5.

Results When inserting the minimal, average and maximum measures of CPU seconds and memory of the packages the following power consumption is calculated:

Minimal power used	0.21 W
Average power used	4.23 W
Maximum power used	11.54 W

Validation As a simple validation these values are multiplied by the total amount of packages to see if it yields a plausible power consumption.

Minimal power used	391 W
Average power used	7.876 W
Maximum power used	21.487 W

Observation On a very quite moment at day, if all packages would use minimal CPU and memory they would use 391 Watt. This is a plausible consumption considering the measured minimal power of all hardware nodes together being around 1625 W during that same time interval (see Figure 8). However, considering the hardware node used only 11% for their packages (see Section 4.5), and by prediction use $\frac{391}{1625} \times 100 = 24\%$ of the total energy, this implies the equation leads to too high predictions. Which is strengthened by the total predicted energy consumption when using the average package measures as independent variables.

The prediction with the highest found package measures as independent variables greatly surpasses the maximum measured power of the test set. However, this can partly be explained by the fact that it is theoretically impossible for all packages to be using the maximum of their resources at the same time. This because they use virtualized resources. Meaning that one packages always thinks it is able to use 4 cores, but instead these 1862 packages having to share $12 \times 16 = 192$ cores.

7 Discussion

The data used within this research has a few shortcomings making it difficult to draw profound conclusions out of the results found in Section 6.

- The final independent variable CPU seconds ranged between 1 to 9 percent of the total capacity of the underlying hardware. Training and testing the linear regression model on data with a broader range would have probably resulted in a higher accuracy.
- Because of the small range, the influence of noise becomes bigger, and therefore the probability on adequate parameters lower.
- The relationship between the memory of the hosting packages and the physical memory measured by the hardware layer could not be researched because there was no memory data gathered on the intermediate virtualization layer.
- The information gathered on memory indicates only the memory used by the package on disk, not the dynamic read and writes done to it. This however would be a better independent variable in predicting the power, since these operations cost more energy then statically containing memory.
- The total amount of data possible to use were 776 time steps of 5 minutes. It would be good if the RRAs would collect data for a longer period of time, to get a larger pool which at least covers the data obtained during a week instead of 2,5 day. Also more frequent data gathering would be beneficial to get more precise and less generalized data.
- The information gathered by the hosting nodes and virtual private servers, did not contain information about the hardware node it resides on. If this would have been the case there was 12 times as much data available, which was less generalized, because it contained the raw data per hardware node, and not the summation of all.

Since this proof of concepts is based on real-world data, the findings in this report are bound to its environmental constraints as mentioned above. Therefore no comparable results could be acquired as proven to be possible by the papers mentioned in Section 2, which some of the hypothesis are based on.

8 Future Work

Within future work, the guideline and equations proposed within this research should be optimized by first gathering data without the shortcomings mentioned in Section 7 and re-train and validate the regression models.

Then, it is possible to look into the inner relationships between the equations, and with this information optimize the final equation 5.

A possible other way to optimize the final estimation is by having an initial run on the idle servers to find the base model parameters, as done in the papers [5], [6] and [7].

Also other resources used by website components have proven influence on the power consumption of a server, as addressed in Section 2, their relationship to the power consumption should be researched and included to make more precise estimations.

If then the estimate on the energy consumption of the presentation and business logic layer are found to be close to reality, a comparable model can be proposed and tested for the other components of a website, like the database layer, the network component or the client-side.

To test whether or not an estimation is close to reality, it could be validation by connecting hardware power meters to the servers.

9 Conclusion

This research investigated *"How to calculate the energy consumption of a website"*, and provided a step by step guideline on how this might be done, based on the findings in related work. Then the proposed solution was tested and validated within a proof of concept, using real-world data supplied by the hosting company Greenhost. The accuracy found during validation of the final equation was not yet high enough to adequately predict the power consumption of the presentation and business logic layers of a Greenhost website. Plausible reasons for this are the shortcomings of the data used, mentioned in Section 7. The first validation during this research however, proves high accuracy and raises the likelihood of more accurate prediction on the other parts, if these shortcomings are resolved or the proposed solution is tested on different data.

In conclusion this report has shown a possible guideline on how to calculate the energy consumption of a website component from real-world data, and although it might need further research and optimization we hope it already contributes to a more aware and knowledgeable future.

References

[1] Gerhard Fettweis and Ernesto Zimmermann. Ict energy consumption-trends and challenges. In *Proceedings of the 11th international symposium on wireless personal multimedia communications*, volume 2, page 6. (Lapland, 2008.

[2] Cisco VNI. White paper: Cisco vni forecast and methodology, 2015-2020. Technical report, Cisco, June 2016. Document ID:1465272001663118.

[3] Netcraft. May 2017 web server survey. Technical report, Netcraft, May 2017.

[4] Alex Homer Jason Taylor Prashant Bansode Lonnie Wall Rob Boucher Jr. Akshay Bogawat WJ.D. Meier, David Hill. *Microsoft Application Architecture Guide, 2nd Edition*. Microsoft, 2010.

[5] Aman Kansal, Feng Zhao, Jie Liu, Nupur Kothari, and Arka A Bhattacharya. Virtual machine power metering and provisioning. In *Proceedings of the 1st ACM symposium on Cloud computing*, pages 39–50. ACM, 2010.

[6] Djamshid Tavangarian Ingolf Waßmann, Daniel Versick. Energy consumption estimation of virtual machines. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 1151–1156. ACM, 2013.

[7] Rajesh Chheda, Dan Shookowsky, Steve Stefanovich, and Joe Toscano. Profiling energy usage for efficient consumption. *The Architecture Journal: Green Computing Issue*, 2008.

[8] Tobias Oetiker. Rrdtool. Technical report, 2015.

10 Apendix

The code used to train and validate the models can be found on the following git repository: `https://github.com/Anouk91/rp2`