Final Year Project Report

Automated Software to Understand Functional Relationship Between Dynamic Energy and Performance Events

Sukrat Kashyap

A thesis submitted in part fulfilment of the degree of

BSc. (Hons.) in Computer Science

Supervisor: Ravindranath Reddy Manumachu

UCD School of Computer Science

University College Dublin

April 4, 2018

Project Specification

General Information:

A energy model representing a relationship between dynamic energy consumption and performance events (PMCs) is constructed experimentally and the experimental dataset has the following format typically (k events, n records):

$E_1,\ x_{11},\ x_{12},\ x_{13} \ldots x_{1k}$
$E_1,\ x_{21},\ x_{22},\ x_{23} \ldots x_{2k}$
$\ldots$
$E_n,\ x_{n1},\ x_{n2},\ x_{n3} \ldots x_{nk}$

where E_i is the experimentally obtained dynamic energy consumption of i-th record and xij are the experimentally obtained performance events (PMCs).

Given such an experimental dataset as an input, the goal is to determine/understand the functional relationship between the dynamic energy consumption and performance events (PMCs).

Two real-life datasets will be provided to the student.
Core:

The goal is to write automated software that will detect the following:

1. Existence of records where the dynamic energy consumption is the same (within an input tolerance) but all PMCs (with the exception of one) have same values. Then the relationship between energy and the one PMC is visualized to see the nature of the functional relationship.
2. Having accomplished step (1), understand the monotonicity of the relationship between dynamic energy consumption and performance events (PMCs).
3. Existence of records where the dynamic energy consumptions are different (within an input tolerance) but all PMCs have same values (within an input tolerance) suggesting the non-existence of a functional relationship.

The software must be written using any one mainstream language but preferably one of the following: C, C++, Python

The software must be well documented and tested.

Advanced:

Given an experimental energy model dataset as an input, the goal is to write software that performs intelligent but computationally feasible simulations where combinations of inputs are generated to study the existence/non-existence of a functional relationship between dynamic energy consumption and PMCs.

The software must be written using any one mainstream language but preferably one of the following: C, C++, Python.

The software must be well documented and tested.

Abstract

With the advent of technology, the demand of energy has also increased dramatically. Increasing number of electric driven equipments such as personal devices, hybrid vehicles and embedded systems has made energy management crucial. To help manage the dynamic energy consumption by the systems. This project attempts to create a software that tries to analyse the dependence of some low-level events with the energy consumption by finding the existence of functional relationship between them. Our project detects and analysis the functional dependence rather than recognising the exact form of dependence. This acts as a form of test to determine the nonexistence of a solution to prevent the unneeded search. It also tries to understand the behaviour and contribution of various parameters towards the output. To understand the behaviour linearity between input parameters and output is assumed.

Acknowledgments

I would like to thank my supervisor Ravindranath Reddy Manumachu and mentor Alexey Lastovetsky for the assistance and guidance of this project. I would like to thank my family and friends for supporting me in my work.

Table of Contents

Chapter 1: **Introduction**

1.1 Motivation

Modern day technology has developed under incredible speed in recent decade and the computing power growth rate is truly phenomenal and lasting impact can be felt and benefit us in many ways. It is important to realise the worldwide effect on environment by the increase in consumption of power by these technology advancements.

According to [6], the power consumption growth rates of PCs are about 7.5% per year. Data Centres and network play much larger role as they both have power consumption rate of 12% each. This considerable growth is due to increasing data to be accessed, stored and processed. The constant expansion of energy consumption leads to increase in carbon emissions. CO_2 emissions from ICT (Information and communications technology) are increasing at a rate of 6% per year, at such rate by 2020 it will account to 12% of worldwide emissions [7].

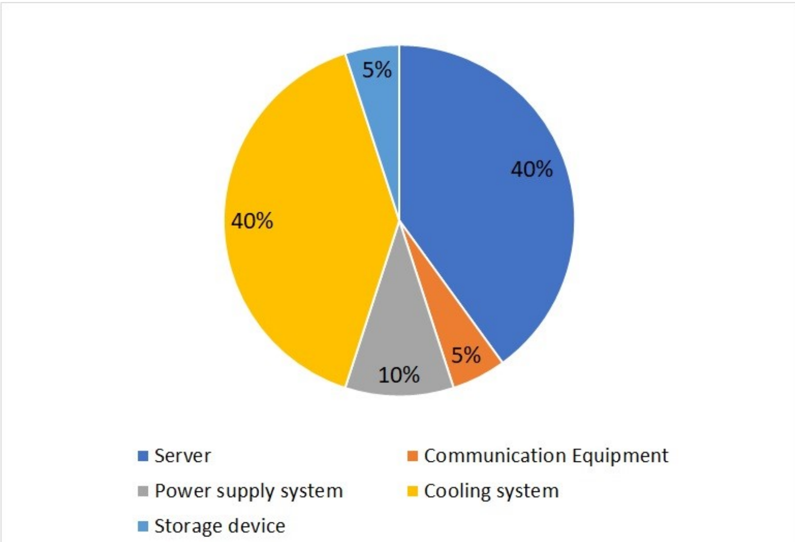

Figure 1.1: Energy consumption distribution of data centers.

The major roles of power consumption by a data ware house is played by the servers and the cooling system which is used to cool down the server physical parts. The figure 1.1, shows that both the servers and the cooling systems account to 80% of the total consumption with both accounting for 40% each [7]. Making power and energy one of the key challenges in system and software designs. In order to improve processor's performance while limiting power consumption, designers have increasingly utilized dynamic hardware adaptions. But to guide to these hardware adaptions it is necessary to measure the power of systems accurately. Such adaptations can reduce power consumption and chip temperature by reducing the amount of available performance. Temperature sensors are slow in reponse due to the thermal inertial of the micro processor. Relying on them would give slow response and detection of temperature changes. Many demonstrations have been done to show that performance counter / performance events are effective proxies of power measurement [1].

The study of the relations between different performance counters and energy consumption has become crucial. The existence and analysis of relations can enable minimization of power consumption leading to less heat generation from the CPU. This decrease in temperature can further reduce the usage of cooling system. This all in turn leads to cost minimization and a step towards eco-friendly environment.

Thus, discovery of the functional relation between the power consumption and the performance events (PMCs) of system, will enable the ability to adjust, and predict the energy consumption for certain computations.

1.2 Aims

The project aims to create a piece of software that will help understanding and finding relations between energy consumption and performance counters. The software must aim to be extensible, easy to use and performant. It should be able to perform analysis like regression and clustering on the data. These actions will help in showcasing the relationship of various parameters and the output. In this case, the parameters are the performance events and output is the energy consumption by the systems. Following are the objectives that are attempted to be accomplished.

1.2.1 Existence of functional relation

Datasets are mostly pair of multiple inputs and a single output. And the possibility of the existence of a relation is what have to be found. Is it possible to define the data in a form of function? Question like this is one of the aims. Defining the data in a form of function can be thought of as explaining the dataset in a form of a formula which can combine various parts of the input parameters and match the output. It is quite impossible to tell that whether a function definetly exists as the data is always a subset of the population and there are number of records which are either not in the dataset or it is not know. But, regardless it can definitely explain the non-existence of a formula by finding a pair of input parameters and outputs that violate the definition of a function.

The pair which violates the definition of a function must be looked at as there is high probability of that data set record being corrupt or if not it can actually act as a contradictiona and explain the non existence. If this outlier is in fact corrupt, this process can be used for data cleanups as well.

1.2.2 Analysis of functional relation

Once, it is not possible to prove functional non existence in the dataset. The dataset must be analysed. In order to better understand the the functional relation of the parameters with the output. The analysis of the relation explains the contribution of a parameter to the result. It gives better understanding about the correlation between two quantities. Does high page faults correlates to high energy consumption? Questions like this is what the analysis of change in parameters with their corresponding change in output answers. This can also be thought of an explaination of the resultant parameter.

In other words, understanding the correlation is the other aim. Once it is know that there is high correlation. Further investigation can be done in order to find the form of the relation.

1.3 Approach

For each of the aims, two approaches have been employed. Both the approaches tries to achieve the same conclusions but with different methodology.

The experimental data sets provided has the following format:

$E_1,\ x_{11},\ x_{12},\ x_{13} \ldots x_{1k}$
$E_1,\ x_{21},\ x_{22},\ x_{23} \ldots x_{2k}$
$\ldots$
$E_n,\ x_{n1},\ x_{n2},\ x_{n3} \ldots x_{nk}$

where k is the number of parameters (performance events), n is the number of records in the dataset. E_i is the experimentally obtained dynamic energy consumption and x_{ij} are the experimentally obtained performance events (PMCs).

1.3.1 Existence of functional relation

The main objective here is to find the nonexistence of functional relationship. In other words, it means proving that the dataset cannot be explained in terms of a function/formula.

First approach here is to find two performance events tuples $(x_{i1},\ x_{i2},\ x_{i3} \ldots x_{ik})$ and $(x_{j1},\ x_{j2},\ x_{j3} \ldots x_{jk})$ that are equal within some tolerance, but their corresponding E_i and E_j dynamic energy consumption are different. Existence of such a tuple in database will lead us to prove the non-existence of a functional relationship.

It is infact easier to find two equal records by ordering the dataset by their parameters and going through the entire dataset once to find equal records and comparing their dynamic energy consumption. The order of complexity of such is $O(N \log(N))$ which is the complexity of sorting as that is the only heavy duty task involved. But it gets complicated when tolerance comes into play. Likewise, task at hand is to find similar records rather than equal records. The project employs 2 methods to measure the similarity between the two data records.

First approach:

The data records are imagined as data points in $k - space$ as we have k number of parameters. Then the whole space is divided into small $k - dimensioncubes$ with dimensions $(t_1, t_2, \ldots t_k)$ where t_i is the tolerance of each parameter. The data points are then put in their respective cubes. And each cube is then analysed to see whether the output of the data points are similar to each other as well.

Second approach:

The data points in $k - space$ are similar if the euclidean distance between them is less than the t provided, where t is the total/max tolerance. Data points which are infact found to be close to each other, their output is compared to see if they are similar to each other.

1.3.2 Analysis of functional relation

Many different type of relations can exists between two variables. But, since correlation betweent the variables is the objective been set. A more general trend of the dynamic energy consumption with the performance events is analysed. Many researches have been done that successfully using linear models, to understand the relation between energy consumption and performance events. [5] The preference given to linear models is due to the trickle down effect of the performance events. [2] Hence, Linear regression is used to analyse the relationship.

Linear regression is done with 2 variables but there are k variables relation to energy consumption making it hard to understand the nature. Multiple linear regression tries to find the best fit to the data. Best fit to the data hinders the tolerance factor in the dataset and also undermines the actual relation. As multiple regression tries to find the best fit and is interested in predicting the output variable. Hence to analyse one of the variable, the data point must be grouped such that their other $k-1$ variables are constant. Each cluster is then visited and linear regression is performed. Two approaches are employed to cluster which are similar to finding similar records in finding the existence of the relation.

First approach:

Data points are clustered by putting the data points in the respective $k - 1 dimension cube.$ Forming a cluster with almost similar $k - 1 variables.$

Second approach:

Euclidean distance is used to isolate similar $k - 1 variables.$ A $k - 1 dimension sphere$ can be imagined for simplicity with its radius being the total/max tolerance.

In both approaches, linear regression is performed on the cluster with the variable which was not used in clustering and the output variable which corresponds to the dynamic energy consumption.

1.4 Structure of report

In this report, the motivation behind the project and brief description of the approaches that will be undertaken to reach the objective are seen already.

This Introductory chapter is followed by Background research, design aspects of the software, implementation of the software, testing and evaluation of the tool followed by the conclusion and the future works.

Background research explains about performance events and how do they influence energy consumption by the system. It also explains the approaches in detail mentioned above and analyses their complexity and how they can be optimised.

Following Background research, design and implementation of the software is deep dig into as we want an extensible, performant tool. Testing and evaluation of the software is explained. How the software is tested and its evaluation on its performance. The report ends with conclusions and future works that shows how the project can be extended and applied to various other domains.

Chapter 2: **Background Research**

Many designers are increasingly utilizing dynamic hardware adaptations to improve performance while limiting the power consumption. Some are using software to decrease power usage for e.g. putting the system in sleep mode when it's in idle state. The main goal remains the same, which is to extract maximum performance while minimizing the temperature and power. Whereas, we want to study and examine the relationship affecting the consumption and then analyse the result to minimize or predict the consumption of energy.

2.1 Energy consumption and Performance Events

First let's look at energy consumption. Energy consumption is the power (Usually in watts) consumed by a system. This system could be the processor/CPU, memory, disk, I/O (Input/Output) system, chipset or the whole computer system itself. So, one can take any of the peripherals and read the power consumption for various performance events. Then analyse if there exists a functional dependence to begin with. If the system is not able to disprove the dataset, one could then try to find a function which could help understand relation between each event and predict for any given system. The values can be read for these performance events can be during a idle state as well as running certain computations.

Now let's look at what are performance events, performance events can be any event which can affect the consumption of energy in some way. Selection of performance events is quite challenging. A simple example would be the effect of cache misses in the processor. For a typical processor, the highest level of cache would be L3 or L2 depending on the type of processor. Now for some transaction which could not be found in the highest level of cache (cache miss) would cause a cache block size access to the main memory. Thus, number of main memory access would be directly proportional to the cache misses. Since these memory access is off-chip, power is consumed in the memory controller and DRAM. Even though, the relation is not simple as it seems but a strong casual linear relationship between the cache miss and the main memory power consumption [2].

A number of other performance events like Instructions executed. The more instruction being executed, will turn on and use more units of the system. Hence, power is consumed as opposed to when the processor is in its idle state [4].

Cache miss, TLB misses are also a good performance events as they seem to have a strong relationship between the power consumption as processor needs to handle memory page walks. Same can be said for Page faults where a program is not able to find mapped address in physical memory as it has not been loaded yet. This causes a trap which can result into number of situations, one of them which is to get the data from disk. In simple terms it is longer walk from cache miss. This walk to the disk and raising of exceptions would consume more energy by the disk as well as the CPU. Another thing to note here most of the relations that we saw above are directly proportional to each other. Increase in the variable like cache miss, number of cycles in CPU etc gives rise to energy consumption by the system. Making linear model a good point to start when analysing the relationship between the performance events variables and energy consumption.

2.2 Related Work

In this section, some of the prior researches are discussed. Hardware performance counter's links to processor power consumption was first demonstrated by Bellosa. In [1], Bellosa demonstrated the high correlation between performance counters such as memory references, L2 cache, floating point operations to processor power consumption.

Gilbert in [4], predicted the power consumption for Intel XScale processors using performance monitoring unit events. Since power consumption is greatly dependent on executing workload, power estimation was done using HPCs (Hardware performance counters) such as Instruction cache miss, TLB misses etc. Linear parmaterisation of power consumption based on performance events was done based on performance events.

In [9], proposed a full system power model for CPU-intensive and memory intensive applications with active cycles, instruction retired and LLC missed as performance events. A full power model for I/O intensive applications was also proposed considering the system level utilization as a complement for performance events. Many machine learning based algorithms like logistic regression, elastic net and k-nearest neighbours were applied to real world application.

Bircher approached in a distinct way by using events local to the processor and eliminating the need for sensors spread across various parts of the systems [3]. Linear regression modeling was done in order to predict the power consumption at runtime. Multiple linear and polynomial regression was done only when accuracy was not obtained.

High correlation between performance events and power consumption was demonstrated by all of them. Many different models were used to predict the power consumption using CPU events. But in this project, an attempt is made in order to understand the monotonic relation between the events and power.

2.3 Existence of functional relation

Definition: *Given a dataset of pairs (x_i, y_i) where $i \in [1, n]$ of two variables x and y, and the range X of x, y is a function of x iff for each $x_0 \in X$, there is exactly one value of y, say y_0, such that (x_0, y_0) is in dataset.* [10]

Above is the definition of functional relation. In our case the $x_i = (p_1, p_2, \ldots p_k)$ and $y_i = E_i$ where p are the performance events and k is the number of events and E is the energy consumption.

In other words, functional relationship is an one to one mapping between our input variables $p_1, p_2, \ldots p_k$ and output E_i. This reasoning can be explained quite intuitively as assuming functional relation is there we can formulate a $f(p_1, p_2, \ldots p_k) = E_i$ where f is function. Now if one $(p_1, p_2, \ldots p_k)$ can give more than one output E_i then that f function is either not correct or f does not exist. A question that immediately arises is what if two different $(p_1, p_2, \ldots p_k)$ gives same output E_i. The answer is it is possible and it doesnot violate the functional relation definition as there is still 1 to 1 mapping from input to output. The only difference is the functional relation is not surjective any more which means that one cannot figure out the input values from output values (i.e. the other way around). But we are only interested in predicting the output rather than inputs from the output variables.

The Proof of the existence of functional relation is given below:

But first let's look at our dataset that will be provided. We know that the data will be in the following format:

Let k be the number of parameters for the energy and n be the number of records in the dataset

$E_1,\ x_{11},\ x_{12},\ \ldots x_{1k}$
$E_2,\ x_{21},\ x_{22},\ \ldots x_{2k}$
$\ldots$
$E_n,\ x_{n1},\ x_{n2},\ \ldots x_{nk}$
where E_n is the dynamic energy for the nth tuple and x_{nk} corresponds to the kth performance event for nth record.

We will use mathematical definition of functional relationship given above to prove the approaches:

Prove: We need to prove that finding atleast 2 equal performance events with different dynamic energies ensures that there exists no functional relationship in the dataset.

Proof. Let us assume that there exists a functional relation such that:

$f(x_{n1},\ x_{n2},\ \ldots x_{nk}) = E_n$
where f is the functional relation for the dataset.

Our task is to find $f(x_{i1},\ x_{i2},\ \ldots x_{ik}) = E_i$ and $f(x_{j1},\ x_{j2},\ \ldots x_{jk}) = E_j$ where $i \neq j$ and $E_i \neq E_j$ and $(x_{i1},\ x_{i2},\ \ldots x_{ik}) = (x_{j1},\ x_{j2},\ \ldots x_{jk})$.

If such i and j exists. Then, we can conclude that the f is not a function by using the definition of a function as this assumed function has two images.

Which contradicts from the hypothesis stated above. Hence by proof of contradiction we could say that f is not a function on the dataset. □

Now the task is to find similar $x_{11},\ x_{12},\ \ldots x_{1k}$ input variables. If it was equality, it could be performed trivially by sorting the data records on the basis of $x_{11},\ x_{12},\ \ldots x_{1k}$. Followed by going throught the data records in that order to find equal records. As equal records will be next to each other when sorted.

But the dataset accumulated is collected from experimental setup. Experimental setups data have some error/tolerance associated to it. As the equipment or software that cannot accurately measure and have some tolerance associated to it. e.g. Energy consumption is measured by a power meter which will have its own error. The tolerance associated with the dataset no longer allows to perform equality on input variables. Hence, a method is needed to measure the equality with the tolerance.

Now after knowing the tolerances the data records in the dataset now looks more of the form:

$E_n \pm e_E,\ x_{n1} \pm e_1,\ x_{n2} \pm e_2,\ \ldots x_{nk} \pm e_n$
where e is now the error associated with the variable A point to note hear is that since the variable represent different entities their units will be different and so will their error associated with it.

2.4 Analysing functional relation

The above analysis of data which corresponds to finding the existence of a functional relation. Now this section corresponds to analysing the functional relation. A functional relation can of many forms, there can be logarithmic functional relation, exponential, linear, polynomial etc. According to [2], shows that functional relation between performance event and dynamic energy is of the linear form. This is due to the trickle down effect of the performance events. Hence, Linear model is assumed to understand the relationship between the events and the consumption.

This project is more interested in finding whether a strong relation of monotonicity exists in the relation or not. And linear regression are best suited for this kind of task. Linear regression is an approach determine how strongly two variables correlate with each other. Correlation not necessarily means causation. Looking at the data points and linear regression values such as Pearson coefficient, R^2 can determine how strong the correlation is between 2 values but it cannot explain the cause of it. Since, it is in the interest of the project to analyse functional relationship which is known to have a linear relation linear regression is proposed.

Dataset contains k parameter variables, to analyse relation between the parameter variables and the output variable. One parameter is chosen at a time. To see the change in the parameter variable chosen and output variable, isolation of the other variables is required. Isolation is done by grouping the data by $k-1$ parameter variable forming a number of clusters with similar $k-1$ paramters. These clusters are then visited and linear regression is performed on the k^{th} variable and the output variable.

From the steps above, it can be observed the parameter to be analysed is isolated by grouping the dataset with the variables that are not being analysed and are equivalent. Since, clustering of dataset is performed again but with different vector dimension of $k-1$, the clustering algorithms are needed by both the objectives.

2.5 Clustering Methodologies

This section introduces two clustering algorithm that was used to cluster data points which are similar or close to each other needed by both of the above objectives. There are many clustering methodologies out there [8]. No clustering algorithm can be universally solve all the problems. Algorithm which favor certain observations, assumptions and favor some type of biases are designed and used.

Clustering involves grouping similar data by their attributes. There were number of clustering algorithms like Heirarchial clustering, K means clustering, Graph based clustering e.g. Chameleon. But we chose, Grid based clustering and Distance based clustering.

Most of clustering algorithms like K means clustering requires the number of clusters to be formed. In the dataset, number of clusters need to be form is not know. IN Heirarchial clustering, heirarchy of dataset is used but the relation is functional and not heirarchial. Graph based clustering usually requires edges, and edges could be formed but they would increase the space complexity exponential. As more storage will be required to store the edges of the dataset.

The two types of clustering algorithms chosen, uses some facts about the dataset. Clustering algorithm which uses tolerances as a measure to cluster the data points would be best suited for this project.

2.5.1 Grid based clustering

Since tolerance must be used as a measure of clustering datapoints. The definition of equality changes from $x_{ia} = x_{ja}$ to $|x_{ia} - x_{ja}| \leq e_a$. Now, if simple sort and search for similar variable would be employed, it would not work correctly as equivalent records would not be next to each other. They might non-equivalent records in between.

Example data to illustrate the same:

E	x_1	x_2
3.5	4.5	6.5
4.1	4.6	10.6
0.2	4.7	6.5
1.6	4.7	7.6

The above three records are sorted by their parameters x_1, x_2. We can see that if the $e = 0.5$ is the absolute error. Then record number 1 and 3 are similar to each other but donot lie next to each other. This increases the complexity of finding similar records from $O(N\log(N))$ to $O(N^2)$ as we donot know where the similar records will lie, so N into N search must be performed which is quite inefficient.

To make it effecient, instead of finding similar records in the dataset. The whole $k-dimensional$ space is divided into small $k-cubes$ whose dimensions are $(e_1 * e_2 * \ldots e_k)$. Each $k-cube$ has its own integer coordinates in space. The data records are grouped together with respect to their respective $k-cube$ coordinates. Since the coordinates are integers (equality can be performed). They can be grouped in $O(N\log(N))$ time. The calculation of the coordinates in which the data point belongs to can be calculated in $O(1)$ time by the following.

$$coordinate = (\left\lfloor \frac{x_{n1}}{e_1} \right\rfloor, \left\lfloor \frac{x_{n2}}{e_2} \right\rfloor, \ldots \left\lfloor \frac{x_{nk}}{e_k} \right\rfloor)$$

Every data point will have a corresponding coordinate they belong to. Data points are then grouped together by their coordinates. Each and every grouped coordinate is a $k-dimensional$ cube. Each of these cubes are then visited and analyse their output variables value. Data points in the same cube with very output variables are collected. They are used for further analysis.

If the user agrees that there is no fault in the dataset collected, then it is concluded that no functional relation exists between the output and the parameters provided. If the user feels that the following data might be faulty, they can be removed from the dataset and the analysis is done again.

If no data points are found that violates the functional relation, it is then said that there might be an existence of relation. The word might is used here because the dataset are always sample of the population. Hence, there is a possibility of existence of a record in population that could violate our functional relation.

After this using the same clustering method, but clustering them by $k-1$ parameters, regression is performed on the k^th parameter chosen and the output.

It can be seen, since the space is divided into cube of certain dimension. The cubes are actually next to each other and they donot overlap. It can be imagined as a building of boxes stack on top and side of each other. There will be points which lie on the edges of the $n-cube$. Since the cubes donot overlap. They will be close to some of points in the cubes next to them. But in the algorithm one data point is assigned only one cluster. This problem is solved by the next approach

Algorithm 1 Grid based clustering

procedure $\text{GRID}(a, b)$
    Input $dataset$: a list of data points, e: errors for each variable
    Result clusters created with each cluster having its unique index
    $list \leftarrow List.empty$
    for each $point$ **in** $dataset$ **do** ▷ N iterations
        $coordinate \leftarrow (\lfloor point.x_1/e_1 \rfloor, \ldots \lfloor point.x_k/e_k \rfloor)$
        $list.add([coordinate, point])$
    sort $list$ by the $coordinate$ value ▷ $N \log(N)$ for sorting
    $result \leftarrow$ group $list$ by the $coordinate$ value ▷ N iterations
    return $result$

which is the distance based clustering. The problem that is seen here comes with performance advantage. Finding and assigning each point clusters of their comes with very time and space complexity.

The algorithm complexity can be measured.

- Time complexity $O(Nlog(N))$: This is because of sorting which is the slowest operation in the algorithm, but neverthless optimizations can be done by using a Dictionary or Binary search tree to group data with equal coordinates. Using Dictionary will give complexity of $O(N)$
- Space complexity $O(N)$: For each data point only the corresponding $coordinate$ is stored.

2.5.2 Distance based clustering

In this method, data points are thought of as vectors in euclidean space of $n - dimension$ where n is number of parameters considered when clustering. The equality of vectors according to which clustering should take place is done by using the euclidean distance between two vectors. Two vectors are said to be equivalent if the distance between the two vectors is less than the tolerance specified.

Let u and v be two vectors in space with dimension n,
then the euclidean distance is define as:

$$dist(u, v) = \sqrt{(u[1] - v[1])^2 + (u[2] - v[2])^2 + \ldots (u[n] - v[n])^2}$$

Now, two vectors u and v are said to be equal or close enough when:

$$dist(u, v) < tol$$

where tol is the maximum distance between two points to call them neighbours.

This tolerance can be thought of as the total tolerance of the between two data points. Since, the difference between two points in all of their different dimensions are squared and "added" together and then square rooted and then limited by a tolerance. It signifies the total tolerance that is allowed between two datapoints.

This clustering is creating spheres of $n - dimension$ for each data point. Here, overlapping of spheres is allowed. Incuring huge performance cost.

In this, algorithm each data point creates it's own sphere with itself in the centre. Data points which lie in that sphere are found. This cluster is then analysed to see if output of each data point to similar to each other. If data points are found which violate the relationship are then returned to the user as a proof of the non existence functional relation.

Algorithm 2 Distance based clustering

procedure DISTANCE_BASED(a, b)
  Input $dataset$: a list of data points, tol: tolerance, $oTol$: output tolerance
  Result subset of the data points that violate functional relation
  $cluster \leftarrow Dictionary.empty$
  for each $point$ **in** $dataset$ **do** ▷ N iterations
    $cluster[point] \leftarrow List.empty$
    for each $neighbour$ **in** $dataset$ **do** ▷ N iterations
      if $dist(point, neighbour) < tol$ **then**
        $cluster[point].add(neighbour)$
  return $cluster$

The algorithm complexity can be measured.

- Time complexity $O(N^2))$: This can be seen as for every data point in the dataset, every datapoint is again visited to find its neighbour. Optimizations like using indexes and parallelisation are used to optimize the running time.
- Space complexity $O(N^2)$: For each data point all of its neighbour are stored. This overhead can be reduced by instead of storing the all the neighbours, whatever computation is needed done and stored the cluster neighbours are thrown away.

2.6 Software

2.7 Applications

As you can see from the proofs above, if any of the conditions above is satisfied then we are able to show that there does not exist any functional relationship between the events and the power consumption. If none of the conditions are satisfied then that shows that there might be an existence of a functional relation. However, it does not guarantee the existence of any form. Non-existence of a function and linear functions are validated. The reason for making a software like this verifies and gives the user the confidence. If data do not fit any functional hypothesis in a space, much time could be saved by preventing the unneeded search of the form of hypothesis as the software will only test the basic conditions that are not supposed to be there for a functional hypothesis.

The software is not restricted to the use of only on dataset which consists of performance events and power consumption. It is a general-purpose software which will for work for any kind of dataset in which user wants to know the existence of functional relation. The refuting of the claim of functional relation on dataset is the objective of the software.

We also know that the dataset provided is usually experimentally measured values which are not accurate. Every measuring device has some margin of error. The software will be flexible in the sense that the equality comparison of values in the approaches will always be done keeping in the margin of error provided.

Chapter 3: **Design and Implementation**

Chapter 4: **Testing and Evaluation**

Chapter 5: **Conclusions and Future Work**

Bibliography

[1] Frank Bellosa. The benefits of event: driven energy accounting in power-sensitive systems. In *Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*, pages 37–42. ACM, 2000.

[2] W Lloyd Bircher and Lizy K John. Complete system power estimation: A trickle-down approach based on performance events. In *Performance Analysis of Systems & Software, 2007. ISPASS 2007. IEEE International Symposium on*, pages 158–168. IEEE, 2007.

[3] William Lloyd Bircher and Lizy K John. Complete system power estimation using processor performance events. *IEEE Transactions on Computers*, 61(4):563–577, 2012.

[4] C Gilberto and M Margaret. Power prediction for intel xscale processors using performance monitoring unit events power prediction for intel xscale processors using performance monitoring unit events. In *ISLPED*, volume 5, pages 8–10, 2005.

[5] Kenneth Obrien, Ilia Pietri, Ravi Reddy, Alexey Lastovetsky, and Rizos Sakellariou. A survey of power and energy predictive models in hpc systems and applications. *ACM Computing Surveys (CSUR)*, 50(3):37, 2017.

[6] Mario Pickavet, Willem Vereecken, Sofie Demeyer, Pieter Audenaert, Brecht Vermeulen, Chris Develder, Didier Colle, Bart Dhoedt, and Piet Demeester. Worldwide energy needs for ict: The rise of power-aware networking. In *Advanced Networks and Telecommunication Systems, 2008. ANTS'08. 2nd International Symposium on*, pages 1–3. IEEE, 2008.

[7] Huigui Rong, Haomin Zhang, Sheng Xiao, Canbing Li, and Chunhua Hu. Optimizing energy consumption for data centers. *Renewable and Sustainable Energy Reviews*, 58:674–691, 2016.

[8] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[9] Shu Yang, Zhongzhi Luan, Binyang Li, Ge Zhang, Tianming Huang, and Depei Qian. Performance events based full system estimation on application power consumption. In *High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on*, pages 749–756. IEEE, 2016.

[10] Robert Zembowicz and Jan M Zytkow. Testing the existence of functional relationship in data. In *Proceedings of AAAI Workshop on Knowledge Discovery in Databases*, volume 93, pages 102–111, 1993.