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
December 7, 2017

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):

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

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 deals and 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 the functional dependence rather than recognising the form of dependence. This acts as a form of test to determine the nonexistence of a solution to prevent the unneeded search.

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 friends and family for supporting me in my work.

Table of Contents

1 Introduction 5

1.1 Motivation 5

1.2 Approach 6

1.3 Structure of report 6

2 Background Research 7

2.1 Energy consumption and Performance Events 7

2.2 Proofs 8

2.3 Applications 10

3 Work done / Work plan 11

3.1 Work done 11

3.2 Work Plan 11

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 [3], 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. This 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 [4].

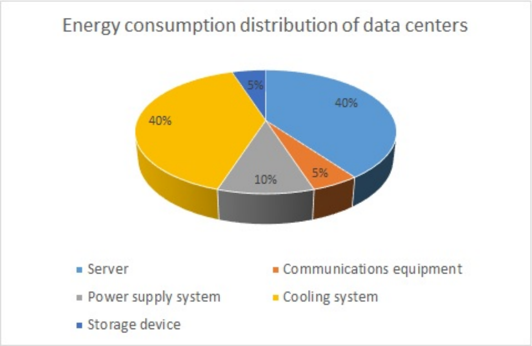

Figure 1.1: Energy consumption distribution of data centers.

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. From the figure 1.1, you can see they both account to 80% of the total consumption with both accounting for 40% each [4]. Hence by finding a relation between different events and energy consumption, we could minimize the power consumption by the systems. Minimising energy consumed will also decrease the heat generated by the system which will in turn lower down the usage of cooling system. Leading us to minimise cost and save environment.

Thus, if we could discover the functional relational between the demand of the power and the performance events (PMCs), that could enable us the ability to adjust, and predict the energy consumption for certain computations.

Above is just only one field that possible usage of the functional relationship in data. In real world there are much more application that required such technique. Such as hybrid vehicle power management, power supply of auxiliary power units etc. There are many past work in this field but only few were focus on the relationship between dynamic energy consumption and PMCs. Hence, in this report we will try to observe the relationship between energy consumption and PMCs. We will explore the monotonicity of the relationship between dynamic energy consumption and performance events (PMCs) and suggesting the non-existence of a functional relationship as well.

1.2 Approach

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

Our first approach to do so is to find two tuples such that n number of performance events are equal within some tolerance, but their dynamic energy consumption is the different. Existence of such a tuple in database will lead us to prove the non-existence of a functional relationship.

Second approach is to group $n-1$ performance events and find dataset where the group size is greater than equal to three. In this approach we first assume here that the dataset provided has dynamic energy consumption equal to a linear combination of its performance events. Then using arithmetic, we compute the constant relating to the nth event. This constant will then be used to find the dynamic energy consumption of the third tuple in the group. If this test fails then we would be able to conclude the non-existence linear functional relation.

The third approach which follows the second is to operate on the existing records and make new records by simple addition as these performance events and consumption are additive in nature. And then on these new records, the first approach is run which means finding different dynamic energy consumption and validating the equality of the performance events associated with it.

The above three approaches help us to find the non-existence of linear relation or a functional relation. Our approaches are based on finding one irregularity which helps us to defy the assumptions that the dataset is linear or functional. This methodology does not prove or find the relationship. It rather defies the possibility of having one.

But we need to prove these approaches for the sake of our user to understand why the dataset cannot represented as a function. The proves for the above will be done in later chapter.

1.3 Structure of report

In this report, you must have already seen the motivation behind the project and brief description of the approaches that will be taken to reach the objective.

This Introductory chapter is followed by Background research, work done and work plan and the timeline of the project deliverables.

Background research explains about performance events and how do they influence energy consumption by the system. It shows some other reasons to test for functional relation. It explains the approaches mentioned above with proofs. So that one can be sure about the correctness of the program and understand the result.

Following Background research is the work done. Work done contains simple observation that one found and analysis which will reflect the work plan. Work plan contains the timeline for the project. It lists out the deliverables of the project.

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. Reading of 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 we can see a strong casual relationship between the cache miss and the main memory power consumption [1].

We can use number of other performance events like Instructions executed. As we know on each instruction being executed, more units of the system are on. Hence, power is consumed as opposed to when the processor is in its idle state [2].

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.

2.2 Proofs

The Proofs below are for different approaches that have been discussed to find the non-existence of a functional relations between energy consumption and number of performance events.

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

$E_1,\ x_{11},\ x_{12},\ \dots x_{1k}$
$E_2,\ x_{21},\ x_{22},\ \dots x_{2k}$
$\dots$
$E_n,\ x_{n1},\ x_{n2},\ \dots x_{nk}$
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 to prove the approaches:

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.* [5]

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:

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

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

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

Restating the above we can say dataset does not contain a functional relation.

Prove: Assuming that the dataset given has linear relationship and if we are able to find the constant for any one of the events. And if it does not apply to the other tuples of data that means linear relationship doesnot exist between the dataset.

Proof:
Let us assume that the energy consumption is a linear combination of the performance events.

$f(x_{i1},\ x_{i2},\ \dots x_{ik}) = E_i$
$f(x_{i1},\ x_{i2},\ \dots x_{ik}) = (\alpha_1 \times x_{i1}) + (\alpha_2 \times x_{i2}) + \dots + (\alpha_k \times x_{ik}) + \alpha_{k+1}$
where α_i are the constants in the linear combination of performance events and $i \in [1 \dots k+1]$

Lets find α_i where $i \in [1 \dots k+1]$
To find this we will have to find atleast 3 records which have their parameter events equal

$x_1,\ x_2,\ \ldots x_k$ except x_i where this x values are value belonging to a row in the dataset.

If we found three records such as:

$E_a,\ x_{a1},\ x_{a2}, \ldots x_{ak}$
$E_b,\ x_{b1},\ x_{b2}, \ldots x_{bk}$
$E_c,\ x_{c1},\ x_{c2}, \ldots x_{ck}$
where the tuples
$(x_{a1} \ldots\ x_{a(i-1)},\ x_{a(i+1)}\ \ldots x_{ak})$, $(x_{b1} \ldots\ x_{b(i-1)},\ x_{b(i+1)}\ \ldots x_{bk})$ and $(x_{c1} \ldots\ x_{c(i-1)},\ x_{c(i+1)}\ \ldots x_{ck})$ are equal to each other, except x_i for some $i \in [1 \ldots k+1]$ where $a, b, c \in [1, n]$ and a, b, c are not equal to each other.

Then

$E_a - E_b$
$= f(x_{a1},\ x_{a2},\ \ldots x_{ak}) - f(x_{b1},\ x_{b2},\ \ldots x_{bk})$
{ By our assumption that E is a linear combination of its parameters }
$= ((\alpha_1 \times x_{a1}) + \cdots + (\alpha_k \times x_{ak}) + \alpha_{k+1}) - ((\alpha_1 \times x_{b1}) + \cdots + (\alpha_k \times x_{bk}) + \alpha_{k+1})$
{ Gathering terms }
$= \alpha_1 \times (x_{a1} - x_{b1}) + \ldots + \alpha_k \times (x_{ak} - x_{bk}) + (\alpha_{k+1} - \alpha_{k+1})$
{ Since we know except x_{mi} and x_{ni} all are equal }
$= \alpha_i \times (x_{ai} - x_{bi})$

From the above we get:

$\alpha_i = (E_a - E_b)/(x_{ai} - x_{bi})$
where $(x_{ai} - x_{bi}) \neq 0$ as $x_{ai} \neq x_{bi}$ by above during our finding phase.

Then we know that using the α_i and applying to result to this equation $E_a - E_c = \alpha_i \times (x_{ai} - x_{ci})$ must be true as well.

If this is false then α_i is not a constant which contradicts our assumption that our E is linear combination of its parameter is false.

Hence using proof by contradiction we can say that the dataset is not linear combination of its parameters.

Prove: Keeping our assumption that the dataset is additive. Creating Pseudo records by simple addition with other records if results into a record whose energy consumption lies in the dataset but with different performance events shows that the dataset is not linear.

Proof:
Let us assume that there exists a linear function f such that:

$f(x_{i1} + x_{j1},\ x_{i2} + x_{j2},\ \ldots x_{ik} + x_{jk}) = E_i + E_j$
and $f(x_{i1},\ x_{i2},\ \ldots x_{ik}) + f(x_{j1},\ x_{j2},\ \ldots x_{jk}) = E_i + E_j$
where α_i are the constants in the linear combination of performance events and $i \in [1 \ldots k+1]$

We know that if f is additive, hence new records can be generated via addition of records in the dataset.

Let V be the set of all dataset rows and dataset rows possible by combining various records in the dataset and pseudo records.

Now if we are able to find data record with $(x_{i1},\ x_{i2},\ \ldots x_{ik}) = (x_{j1},\ x_{j2},\ \ldots x_{jk})$ where $i \neq j$ and $E_i \neq E_j$, where both records belong to the set V.

Then, by the first proof, f is not an linear function. Which contradicts our assumption.

Hence using proof by contradiction we can say that the dataset does not contain a linear function.

2.3 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: **Work done / Work plan**

In this section, we will discuss the progress of the report till now and the work plan for the future. We will be discussing the deliverables with their timeline.

3.1 Work done

As seen earlier, the main objective of this project is to prove the nonexistence of functional relationship or proving that linear combination does not exist in the data records. To do so, we need to first prove that the approaches we are using is correct and true.

Hence, you can see the proofs for approaches in the previous chapter. The proofs show that the discrepancy found in the data can help us to prove that non-functional relationship of the data. It also verifies the correctness of the approaches.

We know that the dataset provided can be small or large. Small dataset can be verified trivially by writing an inefficient code which does all the approaches one by one. But we know that the program must work for large dataset as well. And larger the dataset the slower the program will become. And we would want to write an efficient algorithm.

Simple python command line software was prototyped to create groups of the datasets by using one less performance event column. The software took input of an csv file and then grouped the data together and printed it out in a csv file with a different name in the same directory. This was done to understand the dataset and the project.

A deep understanding of the project and research was done to understand various applications and models that have been created regarding the performance event and energy consumption.

3.2 Work Plan

Our Work plan is to first code the approaches which we have talked about. Test them with known data. Testing is an important part of the project specification.

Second main import task to make the algorithm as efficient as possible which would require creating multiple algorithms, refining and analysing them achieve maximum performance and stability as possible.

We also need to take care about the limit of data that can be processed as data can be large to compute. Hence dividing the work into multiple small task and combining the results would be one of the options. A way to let the user enter the tolerance which could be column specific or dataset specific will be taken for the equality comparison of the values inside the column.

GUI is needed to make it easier for users to interact with the software. The GUI would allow user to take in multiple csv dataset files which would then be processed. If data discrepancy is found that proves that the dataset is not functional, would be displayed to the user in a user-friendly format. For them to know if there is a fault in the record or it might give them insight to the dataset as well.

Deliverables can now be listed:

1. GUI
   (a) Allow users to input multiple files
   (b) A way to input tolerance for multiple column
   (c) Display of records that failed the test
   (d) Display for the result at the end computation with simple measurements
2. Core
   (a) Component to read csv files
   (b) Multi-thread usage
   (c) Finding the discrepancy in the records that fail any one of the three approaches

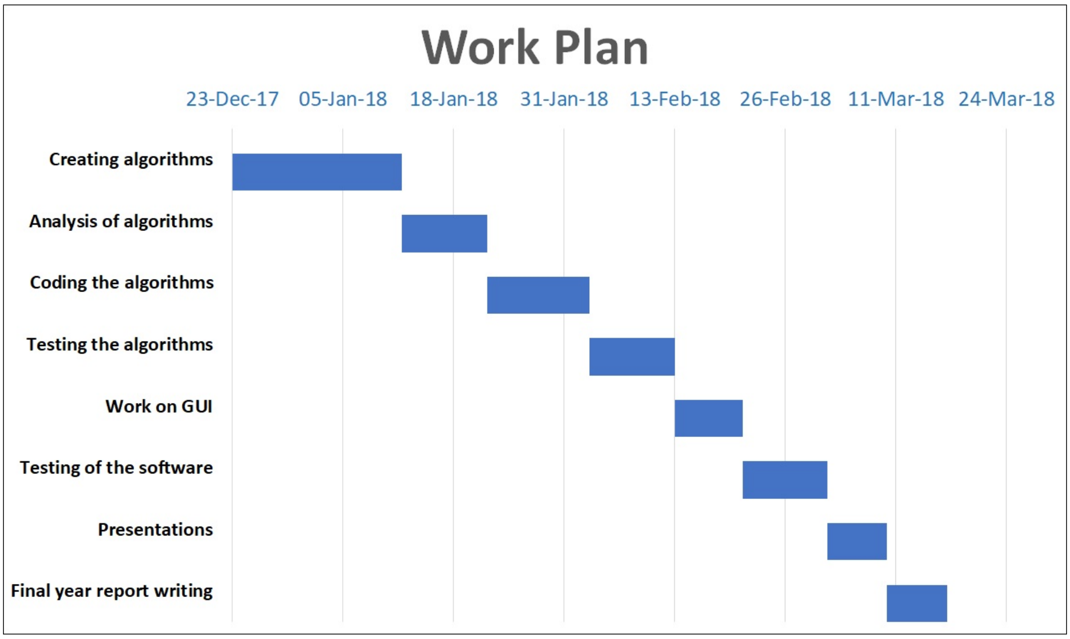

Figure 3.1: Work timeline

As per the Fig: 3.1, First, we will be working on the algorithm which will consist of combining all the three approaches in one and keeping track of which record broke which of the approaches rule to prove the non-existence of the function in the dataset. Second task is to analyse the speed of the algorithm, feasibility and stability. All the programs will then be coded into a compiled programming language. The code will be tested again several input and bugs will be fixed. GUI is the next step to give the user a friendly interface to interact with the program. Software will then be tested. Last two tasks are for completion of the presentation and final year report writing.

Bibliography

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.