A Methodology to Predict the Power Consumption of Servers in Data Centres

Robert Basmadjian
University of Passau
Innstrasse 43
Passau, Germany
basmadji@fim.uni-passau.de

Nasir Ali
University of Passau
Innstrasse 43
Passau, Germany
nasirali@fim.uni-passau.de

Florian Niedermeier
University of Passau
Innstrasse 43
Passau, Germany
niederme@fim.uni-passau.de

Hermann de Meer
University of Passau
Innstrasse 43
Passau, Germany
demeer@fim.uni-passau.de

Giovanni Giuliani
HP Italy Innovation Center
Milano, Italy
Giuliani@hp.com

ABSTRACT

Until recently, there have been relatively few studies exploring the power consumption of ICT resources in data centres. In this paper, we propose a methodology to capture the behaviour of most relevant energy-related ICT resources in data centres and present a generic model for them. This is achieved by decomposing the design process into four modelling phases. Furthermore, unlike the state-of-the-art approaches, we provide detailed power consumption models at server and storage levels. We evaluate our model for different types of servers and show that it suffers from an error rate of 2% in the best case, and less than 10% in the worst case.

Categories and Subject Descriptors

C.0 [**GENERAL**]: Systems specification methodology; C.4 [**Performance of Systems**]: Design studies, Performance attributes, Modeling techniques.

General Terms

Design, Measurement, Performance.

Keywords

Data centre, modelling, IT resources, power consumption.

1. INTRODUCTION

A Gartner press release [13] presented estimations of the global impact of the ICT sector by considering PCs, servers, cooling systems, fixed and mobile telephony, local area networks, office telecommunications and printers. It was shown that these equipments are responsible for 2% of the global CO_2 emissions, which is approximately equivalent to the fuel consumption from the airline industry.

Data centres, due to their housing of powerful ICT[1] equipments, are high energy consumers and therefore accountable for large quantities of emissions. Furthermore, it was stated in [11] that data centres can consume as much energy as a whole city if the number of ICT resources reaches a certain level. As a consequence, power and energy are first-order concerns in such infrastructures due to *economical* (increase of energy costs) as well as *ecological* (world wide desire to reduce CO_2 emissions) reasons. Therefore, minimizing the data centres' energy consumption, on one hand acknowledges the potential of ICT for saving energy across many segments of the economy, on the other hand helps ICT sector to show the way for the rest of the economy by reducing its own carbon footprint.

In [4], it was stated that savings of the order of 20% can be achieved in server and network energy consumption of data centres with respect to the current levels. This goal has been investigated thoroughly in [3] through the application of specific energy aware optimization policies, without violating any Service Level Agreements the data centre has contracted with its users.

To realize energy reduction in data centres, a "plug-in" is developed in the context of [3], which is able to operate on top of current data centre automation and management tools, in order to orchestrate the allocation of ICT resources as well as to slow down under-utilized and to turn off unused equipments. This "plug-in", whose building blocks are depicted in Figure 1, is composed of a set of software modules and has three main roles:

1. Extract monitoring information from existing data centre management tools and provide them to the core modules in a dedicated data structure (*data centre model*) functional to the optimization algorithms.

[1]Consists of all technical and communication technologies used to handle digital information and aid communication, including both computer and network hardware as well as necessary software of a system.

2. Apply the optimization policies, using the *data centre model* updated with *fresh data* by the monitoring module and generate a set of actions to change the data centre into an optimal state from the energy consumption perspective.

3. Enact the list of actions suggested by the optimizer module.

The *data centre model* has the central role of describing all ICT resources (physical, virtual and software) inside the data centre, their interconnections and the respective load applied on the infrastructure. The algorithms of the *optimizer module* are using the *power computation module* which contains specific power consumption prediction models and formulas based on the state of the data centre, as represented in the model instance with updated values from the monitoring activity.

Given the complexity and heterogeneity of data centre infrastructures, the provision of a generic *data centre model* becomes a very complex and a tedious task. In this paper, we introduce a methodology to derive a generic *data centre model* by decomposing the modelling process into 4 phases: *ICT Resources* modelling, *Server* modelling, *Storage* modelling and *Services* modelling. For each of these modelling phases, we identify the most relevant energy-related aspects. Based on this generic *data centre model*, we develop power consumption prediction models (formulas) for the ICT resources. Some of our key findings and contributions are:

- We provide the power consumption model of a server by breaking it down into corresponding power consumption models of processor, memory, hard disk, network interface card, fan and power supply unit.

- Power consumption models are given at SAN devices level.

- For tower-like servers, our power consumption model provides estimations with an error rate of 6% for a CPU utilization more than 60%, and an error rate of 8% for a CPU utilization of less than 60%.

- For blade servers, our power consumption model suffers from an error rates of 2% and 9% for a CPU utilization of 90% and less than 60% respectively.

The rest of this paper is organized as follows: Section 2 discusses related approaches of modelling server and storage systems. After representing in Section 3 the generic data centre model, we introduce in Section 4 our power consumption prediction models for servers and storage systems of data centres. An evaluation is provided in Section 5 and the paper is concluded in Section 6.

2. RELATED WORK

Several models have been proposed in the literature for the purpose of predicting the power consumption (in Watts) of servers and storage systems.

In [14], the authors propose a model that simply predicts a constant (an average) power consumption regardless of a certain utilisation of the server. The two main benefits of this model are the followings : (1) it is easy to compute and no dynamic information is needed, (2) it is similar to the method of estimating a system's power consumption based on the manufacturer's specifications. However, it has the drawback of providing very rough predictions especially for heterogenous systems where not all the servers have similar characteristics. A linear model was introduced in [7] which estimates the power consumption according to the server's CPU utilisation. The main drawback of such a linear model is that it's not suitable for systems that are not CPU-dominated (i.e. file-servers) or workloads that are not CPU-intensive (i.e. streaming, sorting). The model of [8] predicts the power consumption of servers similar to the linear CPU dependent model of [7], while taking into account also the utilisation of the hard disk. Moreover, it was stated in [14] that the model of [8] is able to provide more accurate estimations than the linear CPU dependent model of [7]. Furthermore, several models have been proposed that consider the dynamic power consumption of the hard disk by using the number of I/O requests or the number of disk transfers as parameters, in order to get some idea about the balance of random vs. sequential I/O. However, the authors of [14] found these kinds of models to be not more accurate than those simply using disk utilisation. A model was proposed in [6] that extends the CPU and disk utilisation model by additionally looking at performance counters of the system, as far as available (for instance the amount of instruction-level parallelism, the activity of the cache hierarchy, or the utilisation of the floating-point unit). This model turned out to be the most accurate one in [14]. However, performance counters are accessed differently on each processor type. As a matter of fact, this model is not usable across heterogeneous systems.

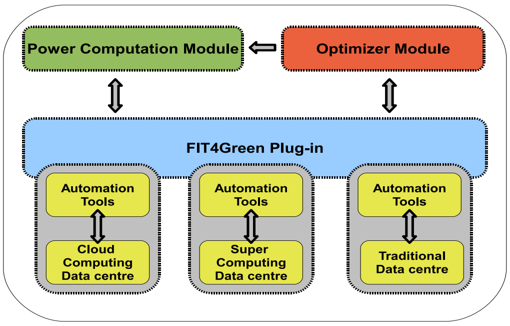

Figure 1: The plug-in flow diagram.

There are several detailed power consumption models for storage systems. In [1] and [16] the power consumption of storage devices is modelled. STorAge Modeling for Power (STAMP) has been developed in [1]. STAMP is a method to provide workload-aware power estimation for enterprise storage. It is able to provide power estimations for a storage controller, an array, or a single disk. The disk simulation environment Dempsey is presented in [16] and includes accurate modelling of disk power consumption. Dempsey demonstrated to be able to model hard-disk power consumption efficiently and accurately.

3. DATA CENTRE MODEL

The most relevant energy-related ICT resources of a data centre are represented in our generic *data centre model*. The physical resources existing in the data centre site are mapped into objects of specific classes, i.e. the *data centre model* con-

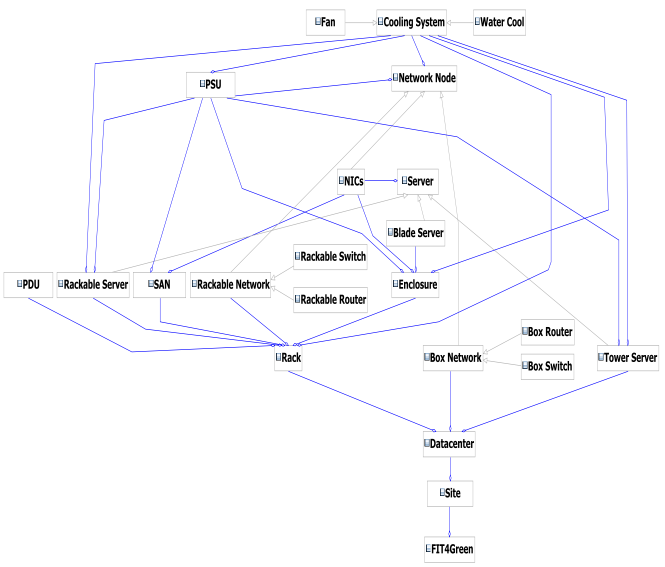

Figure 2: UML class diagram for ICT resources modelling.

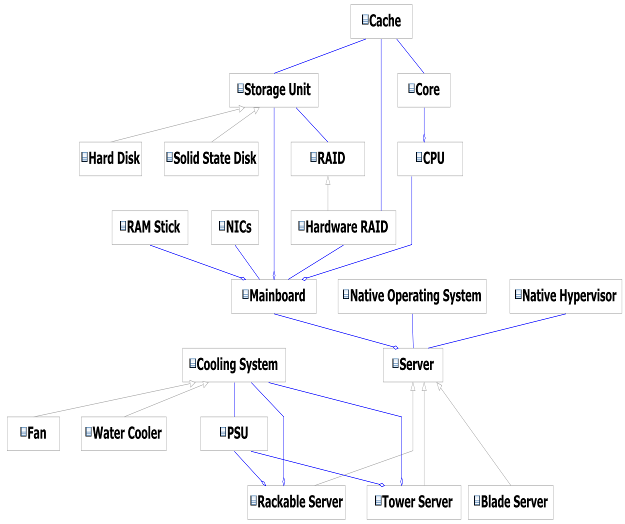

Figure 3: UML class diagram for Server modelling.

tains one object for each relevant physical entity. The main rationale for this principle is that the model should reflect exactly the ICT resources (mainly server, storage and networking systems), and these are managed by the data centre operators, who are able to identify them inside the site and will be responsible for the editing of the *model instance*[2] specific for their data centre (some forms of automated discovery of resources and export of model instances might be provided, but at the end the data centre operators are responsible for the validation of their data centre configuration, i.e. the model instance of the data centre). Next, we describe the design of our generic *data centre model* by decomposing it into four modelling phases: ICT Resources (3.1), Server (3.2), Storage (3.3) and Services (3.4), and give the relevant energy-related attributes necessary for power consumption predictions. It is worthwhile to note that the *model* also includes networking elements (e.g. routers, switches, NICs, etc) whose power consumption prediction models are not described in this paper, but they appear in the figures.

3.1 ICT Resources Modelling

Since typically it's possible to federate a set of data centres scattered geographically, then the root of the class hierarchy (*FIT4Green* class, see Figure 2) contains a set of sites (*Site* class), whose main attribute is the Power Usage Effectiveness "PUE". The "PUE" is related to the efficiency of the data centre facility, and it's determined as the ratio between the amount of power consumed by the site over the power consumed by its ICT resources (in this model it's considered as a constant value, each site having a different figure based on its efficiency). This attribute is important when evaluating the impact on global power consumption of migrating software load from one data centre to a federated one (possibly a more efficient one).

In general one site can, depending on its layout and internal organization, contain a set of data centres (*Datacenter* class); in this model each one of them is dedicated to a specific computing style represented in the "computingSyle" attribute. At the current state, *Traditional Computing*, *Supercomputing* and *Cloud Computing* are the supported categories. It is worthwhile to note that different optimization policies are applied depending on the value of this attribute.

Inside each data centre, there are ICT equipments which can be placed either inside racks or in independent cases (single box stands, generally with tower form factor), in addition to network devices. Therefore, inside the model, the *Datacenter* class contains potentially a set of racks (*Rack* class), a set of servers with tower case (*TowerServer* class) and a set of box-like network devices such as routers and switches (*BoxNetwork* class whose details are not covered in this paper). Inside a rack there are many different elements: rack-mountable servers are represented by the *RackableServer* class, enclosures for blade form factor servers are represented by the *Enclosure* class, and the blade servers inside them are modelled through the *BladeServer* class. All these three classes representing servers are specializations of a single parent *Server* class, which contains the common attributes and acts as the container for the internal server components. Typically each *TowerServer* and each *RackableServer* has independent power supply units (*PSU* class) and cooling system (*CoolingSystem* class), while *BladeServer* shares the *PSU* and *cooling system* from its *Enclosure*. Racks contain typically a set of power distribution units (*PDU* class): in most cases they're passive devices simply used to connect the different power plugs of the rack elements; in some cases they can also be active and perform power measurements and switch on/off functions. Storage Area Network devices (*SAN* class) are generally mounted inside racks and get power from the internal *PDUs*, like any other element of a correctly installed rack. Finally, network devices such as routers and switches can also be mounted inside racks (*RackableNetwork* class whose specifics are out of the scope of this paper).

3.2 Server Modelling

[2]Represents the exact configuration of the data centre along with its corresponding ICT resources.

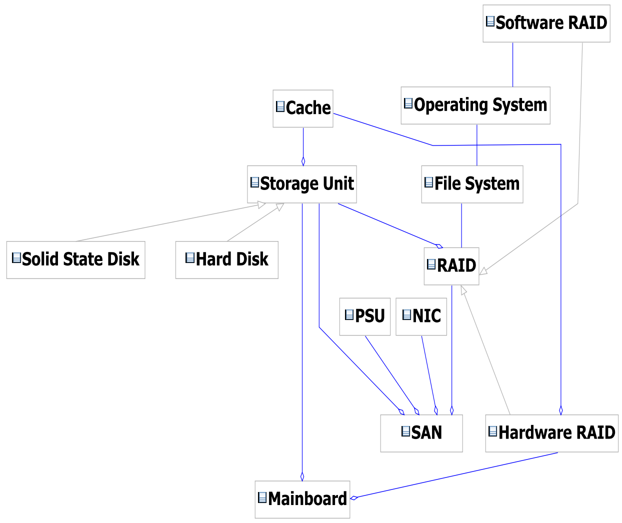

Figure 4: UML class diagram for Storage modelling.

In this section, we introduce the *Server modelling* where we only present the most relevant attributes of the classes due to space considerations. The UML class diagram of Figure 3 illustrates the Server modelling where *Server* class represents an abstraction for a generic server computer, such that the different specializations used in data centre model (*TowerServer*, *RackableServer*, and *BladeServer* classes) are distinguished by their physical form factor, which can have effects on the power consumption. Among other attributes of the *Server* class, "computedPower" and "measuredPower" indicate respectively the power consumption of the server computed through our power computation module (see Section 4.1) and the power consumption of the server measured by means of a possible power meter. Note that these two parameters serve for the model refinement as they are useful to compare measured and computed values to check and refine the power consumption models.

In general, the *Server* class is composed of at least single *Mainboard* class and runs several software applications (see Section 3.4). The *Mainboard* holds the following crucial components of the system: Central Processing Units (*CPU* class), Random Access Memories (*RAMStick* class), Network Interface Cards (*NICs* class), hardware RAIDs (*HardwareRAID* class) and Storage Units (*StorageUnit* class). Among other attributes of *Mainboard*, "computedPower" has the same definition as the one in the *Server* class, whereas "memoryUsage" denotes the overall usage of the attached memories whose value is updated constantly through the monitoring module.

A *CPU* is composed of at least one core (*Core* class) where each core has its own cache (*Cache* class). Among other attributes of *CPU*, "architecture" indicates the processor's manufacturer (e.g. INTEL, AMD, etc), "cpuUsage" denotes the utilization of the processor whose value is updated through the monitoring module, whereas "computedPower" has the same definition as *Server* class. Each *Core* operates on a specific "frequency" and "voltage". Furthermore, "coreLoad" represents the utilization of the corresponding core whose value is updated through the monitoring module, whereas "computedPower" has the same definition as *Server* class.

The *RAMStick* class has several attributes relevant to power consumption estimation: "size" denotes the size of the memory whereas "voltage" reflects the supply voltage under which the memory operates which are highly dependent on its "type" (e.g. DDR_1, DDR_2, DDR_3, etc). "frequency" denotes the frequency of the memory, "vendor" indicates the manufacturer (e.g. KINGSTON, HYNIX, etc), "isBuffered" shows whether the memory is fully buffered or not. The values of all these attributes can be populated inside the data centre model based on the manufacturer's data sheets. Finally, "computedPower" has the same definition as the one in the *Server* class.

A *Server* can be connected to a storage unit (*StorageUnit* class) either directly through the *Mainboard* or through a *Hardware RAID* device. Additional information regarding Storage modelling is provided in Section 3.3. Finally, *Tower* and *Rackable* servers have their own power supply units (*PSU* class) and cooling systems (*Cooling System* class) which can be either a *Water Cooler* or a *Fan*. The most relevant energy-related attributes of *PSU* are the followings: "efficiency" indicates the power efficiency usage which is highly related to the "load", whereas "computedPower" and "measuredPower" denote respectively the power consumption of the PSU computed through our power computation module and the power consumption of the PSU measured by means of a possible power meter. Among the other attributes of *Fan*, "width" and "depth" indicate respectively the width and the depth of the Fan, whereas "actualRPM" denotes its current rotation speed.

3.3 Storage Modelling

The UML class diagram of Figure 4 illustrates the Storage modeling where the *Storage Unit* class represents the abstraction for all kinds of disk-like devices providing the physical storage for data. *Storage Unit* devices can be directly connected to *Servers* through the *Mainboard* or to *SAN* devices; optionally a *Hardware RAID* controller can be used to provide the different levels of RAID support to servers or SAN devices. Note that SAN devices have additionally their network interface cards (*NIC* class) as well as power supply units (*PSU* class). We consider both traditional disks with revolving platters (*Hard Disk* class) and solid state disk controllers (*Solid State Disk* class) as possible *Storage Unit* devices.

The most relevant energy-related attributes of the *Storage Unit* are the followings: "maxReadRate" and "maxWriteRate" denote respectively the maximum number of read and write operations that can be performed on the disk. The values for both of these attributes can be populated inside the data centre model based on the manufacturer's data sheets. "readRate" and "writeRate" indicate respectively the actual number of read and write operations performed on the disk. The values for both of these attributes are updated constantly through the monitoring module. Finally, "computedPower" represents the power consumption of the storage unit computed through our power computation module.

Each *Hard Disk* has the following different energy-related attributes: "rpm" indicates the round per minute of the hard disk, "platters" denotes the number of platters, whereas "AAM" presents whether the hard disk is equipped with Automatic Acoustic Adjustment feature. For the *Solid State Disk*, "powerByRead" and "powerByWrite" denote respectively the power consumed by read and write operations.

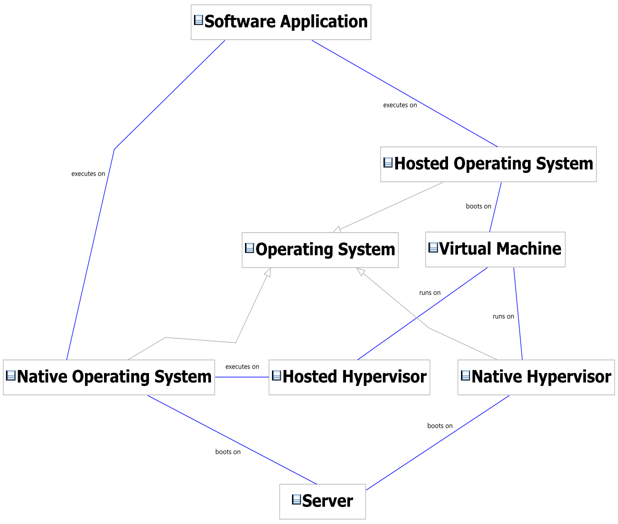

Figure 5: UML class diagram for Services modelling.

Here the distinction is made due to the fact that read and write operations in solid state disks have different power consumption behaviour.

3.4 Services Modelling

The power optimization policies of [3] seek to distribute (and also potentially move) *application load* to the computing element(s) which consume less power and still satisfy the Service Level Agreements for the software application. In this perspective, all software components, either at application or system level, independently from the computing style of the data centre, can be modeled as load generator for the IT resources. In the title of this section, the term *service* is used to represent any kind of software which can directly or indirectly affect the load of an IT resource. Inside the "plug-in", several system parameters are collected through existing data centre automation and monitoring frameworks (like for instance "cpuUsage" and "memoryUsage" mentioned in the previous sections), together with high level information about the software running on the data centre infrastructure, which is then mapped into the model depicted in Figure 5.

Physical servers (of any specialization of the *Server* class) running a native operating system have a 'boots' relation with a *NativeOperatingSystem* class objects, which on its turn is further connected to *SoftwareApplication* class objects (representing regular applications or system services) or to *HostedHypervisor* class objects (for running virtual machines) through 'executes' relation. In contrast, physical servers natively hosting virtual machines have an association with a *NativeHypervisor* class object. Both *HostedHypervisor* and *NativeHypervisor* objects can have a set of 'runs' relation with objects representing the Virtual Machines they're hosting (*VirtualMachine* class). Virtual machines are functionally almost equivalent to physical servers; in fact they have a 'boots' relation with a *HostedOperatingSystem* class, which *executes* software applications, in the same way as its *native* counterpart. *NativeOperatingSystem*, *NativeHypervisor* and *HostedOperatingSystem* are specialization of their parent *OperatingSystem* class: in fact, in reality, they're all special versions of operating systems - including native hypervisors, which are typically deployed inside embedded distributions of native operating systems.

4. POWER CONSUMPTION MODELS

In this section, we introduce the power consumption models for servers and storage systems of a data centre. Note that these models are used by the *power calculator module* of the "plug-in" in order to estimate the power consumption of these ICT resources of a data centre.

4.1 Power Consumption of Servers

It was shown in [7] that the power consumption of a server with a local disk is distributed among its components in the following manner: Processor: 37%, Memory: 17%, Mainboard: 12%, Hard disk: 6%, Fan: 5% and PCI slots: 23%. Based on these studies and results, we can notice that the main contributors in the power consumption of a server are the processor (CPU), memory (RAM), fans and storage devices. Next, we provide the power consumption prediction models for each of the above-mentioned components and then give the generic model of a server.

4.1.1 Processor

In general, processors are divided into two categories: single-core and multi-core. The difference between the two classes is mainly the technology with which they are devised that has an impact on the power consumption of the processors.

It was shown in [7] that the power consumption of single-core processors is linearly increasing with its utilization and is given by the following equation:

$$P_{CPU} = P_{idle} + (P_{max} - P_{idle})\frac{L}{100}, \quad (1)$$

where P_{max} and P_{idle} denote respectively the maximum (100% utilization) and the idle (no activity) power consumptions of a processor, whereas L denotes the utilization of the processor ("cpuUsage" in our case).

Based on our observations[3], we noticed that individual core of multi-core processors has the same power consumption behavior as that of the single-core ones: the power consumption increases linearly with the utilization. As a matter of fact, inspired by the linear utilization-based power consumption model of Equation (1), we give the power consumption of individual cores by the following equation:

$$P_C = P_{max}\frac{L_C}{100}, \quad (2)$$

where P_{max} denotes the maximum (100% utilization) power consumption of a core, whereas L_C indicates the utilization of the core ("coreLoad" in our case). Based on Equation (2), then the power consumption of a processor consisting of at least one core is given by the following equation:

$$P_{CPU} = P_{idle} + \sum_{i=1}^{n} P_{C_i}, \quad (3)$$

where n denotes the total number of cores for a given processor, P_{C_i} indicates the power consumption of an individual core, whereas P_{idle} represents the idle (no activity) power consumption of a processor. It is worthwhile to note that

[3] A custom benchmark that loads each core of a multi-core processor separately with ranges between 1-99%.

the idle power consumption P_{idle} of a server is considered constant in this paper.

In order to compute the maximum power consumption P_{max}, we adopted the following well known CMOS circuits [5] power consumption equation:

$$P_{max} = V_{max}^2 * f_{max} * C_{eff}, \tag{4}$$

where V_{max} and f_{max} denote respectively the voltage and frequency at maximum utilization, whereas C_{eff} indicates the effective capacitance which includes the capacitance C and switching activity factor $\alpha_{0\to1}$.

Based on our observations, we noticed that in certain cases, the power consumption of *multi-core* processors of Equation (3) is not always a simple summation of the power consumptions of their constituent cores. More precisely, unlike *AMD* [2] processors (where each core has its own cache), *INTEL* [10] ones consume less power due to the fact that certain cores share some information and hence induce less inter-core communications. As a matter of fact, we have introduced a power reduction factor for processors of *INTEL* architecture ("architecture" in our case) such that this power reduction factor changes significantly based on the number of cores (i.e. dual-core, quad-core, etc). Concerning the operating system's influence on the power consumption, we remarked that *Linux* operating systems have slightly higher consumption than *Windows* ones. As a consequence, further power reduction is applied to processors of servers running *Windows* as their native operating system.

It is worthwhile to note that the accuracy of our processor power consumption prediction of Equation (3) is based on the level of information (concerning the utilization) that the *monitoring module* provides to the *power calculator module*. In particular, if this module provides information concerning the utilization of individual cores ("coreLoad" in our case), then the estimations are not compromised. However, if the monitoring module provides information only concerning the utilization of a processor ("cpuUsage" in our case), then we evenly distribute this utilization among the cores of the processor, in order to provide the utilization of each core. As a consequence, the accuracy of Equation (3) is sacrificed.

4.1.2 Memory

In general, there exist several types of Random Access Memories which differ in terms of the technology (e.g. DRAM, SDRAM, etc) with which they are devised. In this paper, we focus on the *Synchronous Dynamic RAM* (SDRAM) technology because it is nowadays the most commonly used one in data centres.

Besides being of several types (e.g. DDR, DDR_2, DDR_3, etc), SDRAMs are divided into two categories: buffered and unbuffered. The difference between the two classes is that the former is used to increase reliability, speed and density of memory systems. Since DDR is a fairly old technology and is barely found in servers of today's data centres, then we give the power consumption models of DDR_2 and DDR_3 both for buffered and unbuffered SDRAMs.

Given an unbuffered SDRAM of type DDR_2[4] or DDR_3[5], then its power consumption at the idle state is given by:

$$P_{RAM\_idle} = \sum_{i=1}^{n} s_i * p, \tag{5}$$

[4]http://en.wikipedia.org/wiki/DDR2_SDRAM

[5]http://en.wikipedia.org/wiki/DDR3_SDRAM

Table 1: Values of p for different unbuffered DDR_2 SDRAMs.

Vendor	Value
Kingston[12]	$\frac{f}{1000}$
Samsung[15]	$0.95 * \frac{f}{1000}$
Hynix[9]	$1.9 * \frac{f}{1000}$
Generic	$1.45 * \frac{f}{1000}$

where n denotes the total number of installed memory modules and s indicates the size of each individual memory. The value of p varies based on the *type* and the *vendor* of the memory. For an unbuffered DDR_2 SDRAM, Table 1 gives the values of p for different vendors, where f denotes the *frequency* of the memory module. For a buffered DDR_2 SDRAM, Table 2 shows the power consumption at the idle state for different vendors. Note that the *Generic* vendor type provides a rough estimation of the idle power consumption for vendors other than those mentioned in Tables 1 and 2. Based on Equation (5), the power consumption of an unbuffered DDR_2 SDRAM is given by:

$$P_{RAM} = P_{RAM\_idle} + \gamma * \beta, \tag{6}$$

such that $\beta = 7.347$, whereas $\gamma \in [0, 1]$ is defined later. Based on our observations, we noticed that β is constant and independent of the size or number of memory modules. This is due to the fact that there is only 1 active operating rank per channel regardless of the number of modules or module ranks in the system. The remaining other ranks and other memory modules are in idle mode drawing less power. For a buffered DDR_2 SDRAM, the power consumption is given by:

$$P_{RAM} = P_{RAM\_idle} + \gamma * 2.3 * \beta, \tag{7}$$

such that $\beta = 7.347$, whereas $\gamma \in [0, 1]$ is defined later. For an unbuffered DDR_3 SDRAM, the value of p is given by the following equation:

$$p = \frac{f}{1000} + \alpha\sqrt{f}(f_c - f), \tag{8}$$

where $f_c = 1600$ indicates the reference frequency, f denotes the input frequency (MHz), whereas α is constant having a value of 0.000026. For a buffered DDR_3 SDRAM, the idle power consumption is twice the same as the idle power consumption of an unbuffered memory. Based on Equation (5), the power consumption of an unbuffered DDR_3 SDRAM is given by:

$$P_{RAM} = P_{RAM\_idle} + \gamma * 1.3 * \beta, \tag{9}$$

such that $\beta = 7.347$, whereas $\gamma \in [0, 1]$ is defined later. For a buffered DDR_3 SDRAM, the power consumption is given by:

$$P_{RAM} = P_{RAM\_idle} + \gamma * 1.9 * \beta, \tag{10}$$

such that $\beta = 7.347$, whereas $\gamma \in [0, 1]$ is defined next.

Since the only information that the *monitoring module* of the "plug-in" provides is the total used memory ("memoryUsage" in our case) and no information is provided concerning when each memory module changes its state from idle to accessing modes, then we adopted the following two techniques in order to derive values for γ:

Table 2: Idle power consumption for different buffered DDR_2 SDRAMs.

Vendor	Formula
Kingston[12]	$2.2 * P_{RAM\_idle}$ of unbuffered
Samsung[15]	$4.26 * P_{RAM\_idle}$ of unbuffered
Hynix[9]	$1.65 * P_{RAM\_idle}$ of unbuffered
Generic	$2.7 * P_{RAM\_idle}$ of unbuffered

1. If the processor is in idle state (no activity), then we also assume that the memory modules are in idle state ($\gamma = 0$).
2. If the processor is not in idle state, then we adopt a probabilistic approach in modeling γ, such that the more total memory is in use, the higher in probability that a memory access is performed.

4.1.3 Hard Disk

Typically, the hard disk's power consumption can be split up into three main parts: *idle*, *accessing*, and *startup* modes. The disk is in idle mode when no activity (read or write) is carried out, whereas it is in accessing mode when read or write operations are performed on the disk. The disk is in startup mode when all of its mechanical and electrical components are activated.

Based on our observations, we noticed that the idle mode power consumption can be further broken down into idle, standby and sleep states. Moreover, we observed that the power consumptions in standby and sleep states are quite the same and these are around 10% of the idle state power consumption. This is due to the fact that during standby and sleep states, the disk's mechanical parts are significantly stopped. Then, the idle mode power consumption of the hard disk is given by:

$$P_{HDD\_idle} = P_{idle}(\alpha + 0.2 * \beta), \quad (11)$$

such that $\alpha \in [0, 1]$ indicates the probability that the disk is in idle state, $\beta \in [0, 1]$ denotes the probability that the disk is in standby and sleep states (the values of α and β are given later), whereas P_{idle} is the idle state power consumption provided by the manufacturer's data sheet. Furthermore, we remarked that, the startup and accessing mode power consumptions are respectively 3.7 and 1.4 times (in average) more than that of the idle state power consumption. Then, the power consumption of the hard disk is given by:

$$P_{HDD} = a * 1.4 * P_{idle} + b * P_{HDD\_idle} + c * 3.7 * P_{idle}, \quad (12)$$

such that $a, b, c \in [0, 1]$ denote respectively the probability that the disk is in accessing, idle and startup modes, whereas P_{idle} is the idle state power consumption provided by the manufacturer's data sheet.

The monitoring module of the "plug-in" provides information regarding the average number of read ("readRate") and write ("writeRate") operations per second performed on the hard disk. Since no information is provided concerning when each disk changes from idle to accessing and to startup modes, then we adopted the following two techniques in order to derive values for a, b and c:

1. If the average number of read and write operations are zero ("readRate" = "writeRate" = 0), then we assume that the disk is in its idle mode ($a = c = 0$ and $b = 1$).
2. If the average number of read or write operations are not zero, then we adopt a probabilistic approach in modelling the mode changes such that:
   - If "readRate" > 0 and "writeRate" > 0, then $a = \frac{readRate+writeRate}{maxReadRate+maxWriteRate}$,
   - If "writeRate" = 0, then $a = \frac{readRate}{maxReadRate}$,
   - If "readRate" = 0, then $a = \frac{writeRate}{maxWriteRate}$,

whereas $b = 0.9 * (1 - a)$ and $c = 0.1 * (1 - a)$. Finally, in order to derive values of $\alpha, \beta \in [0, 1]$ for the idle mode power consumption, we adopted the following probabilistic approach:

1. If $0 < b \leq 0.3$, then we set $\alpha = 0.9$ and $\beta = 0.1$.
2. If $0.3 < b \leq 0.6$, then we set $\alpha = 0.5$ and $\beta = 0.5$.
3. If $0.6 < b \leq 1$, then we set $\alpha = 0.1$ and $\beta = 0.9$.

We can notice from the above equations that the more the hard disk is in idle mode ($b \simeq 1$), the higher is the probability that it will remain in standby and sleep states.

4.1.4 Mainboard

The mainboard is the central printed circuit board that holds many of the crucial components of the server. The power consumption of the mainboard is given by the following equation:

$$P_{Mainboard} = \sum_{i=1}^{l} P_{CPU} + P_{RAM} + \sum_{j=1}^{m} P_{NIC} + \sum_{k=1}^{n} P_{HDD} + c, \quad (13)$$

where l denotes the total number of processors whose power consumption is P_{CPU} of Equation (3), P_{RAM} is the power consumption of the memories (Section 4.1.2), m indicates the total number of network interface cards whose power consumption is P_{NIC} (out of the scope of this paper), n denotes the total number of attached hard disk drives (including those for the hardware RAID) whose power consumption is P_{HDD} of Equation (12), whereas c is constant having a value of 40 in case of tower and rackable servers, and a value of 55 in case of blade servers.

4.1.5 Fan

Given a fan of width w (in mm), depth d (in mm) and revolutions per minute a, then its power consumption is given by the following 4^{th} order polynomial:

$$\begin{aligned} P_{Fan} &= 8.33068 * 10^{-15} * a^4 + 8.51757 * w^4 - 2.9569 * d^4 \\ &- 1.10138 * 10^{-10} * a^3 + 54.6855 * w^3 - 76.4897 * d^3 \\ &+ 4.85429 * 10^{-7} * a^2 + 258.847 * w^2 - 1059.02 * d^2 \\ &- 6.06127 * 10^{-5} * a + 32.6862 * w + 67.3012 * d \\ &- 5.478 \end{aligned} \quad (14)$$

4.1.6 Power Supply Unit

The power supply unit is the only means of supplying power to the numerous components of the server. In order to compute the power consumed by a PSU having an *efficiency* of e, we adopted the following techniques:

1. If the monitoring module provides information at the PSU level ("measuredPower" of PSU), then the power consumption is given by the following equation:

$$P_{PSU} = \frac{measuredPower * (100 - e)}{100}.$$

2. If the monitoring module provides information only at the server level ("measuredPower" of server), then we assume that this "measuredPower" of the server is evenly distributed among the PSUs ("countPSU") providing power to the components, and compute the power consumption by the following equation:

$$P_{PSU} = \frac{(\frac{measuredPower}{countPSU}) * (100 - e)}{100}.$$

3. If the monitoring module provides no information neither at the server level nor at the PSU level, then we compute the power consumption by the following equation:

$$P_{PSU} = (\frac{serverPower}{countPSU * e}) * 100 - \frac{serverPower}{countPSU},$$

such that *serverPower* indicates the power consumption of all the components of the mainboard as well as fans of the server that have been calculated using previously introduced formulas (Sections 4.1.1 - 4.1.5), whereas *countPSU* denotes the number of PSUs providing power to the server.

4.1.7 Server Power

Given a server composed of several mainboards, fans and power supply units as illustrated in Section 3.2, then we compute its power consumption in the following manner:

1. If the server is of type *blade*[6], then its power consumption is given by the following equation:

$$P_{Server} = \sum_{i=1}^{l} P_{Mainboard}. \quad (15)$$

2. If the server is of type *Tower* or *Rackable*, then its power consumption is given by the following equation:

$$P_{Server} = \sum_{i=1}^{l} P_{Mainboard} + \sum_{j=1}^{m} P_{Fan} + \sum_{k=1}^{n} P_{PSU}, \quad (16)$$

such that l indicates the total number of mainboards whose power consumption is given by $P_{Mainboard}$ of Equation (13), m denotes the total number of fans whose power consumption is given by P_{Fan} of Equation (14), and n represents the total number of power supply units whose power consumption is given by P_{PSU} of Section 4.1.6.

4.2 Power Consumption of SANs

Given a Storage Area Network (SAN) device composed of several hard disks, network interface cards and power supply units as illustrated in Section 3.3, then its power consumption is given by the following equation:

$$P_{SAN} = \sum_{i=1}^{l} P_{HDD} + \sum_{j=1}^{m} P_{NIC} + \sum_{k=1}^{n} P_{PSU}, \quad (17)$$

where l denotes the total number of hard disk drives whose power consumption is P_{HDD} of Equation (12), m indicates the total number of network interface cards whose power consumption is P_{NIC} (out of the scope of this paper), and n denotes the total number of power supply units. It is worthwhile to note that the power consumption of SAN's PSU is computed using the same technique as in Section 4.1.6, however by replacing the attributes related to the servers with those concerning SAN devices.

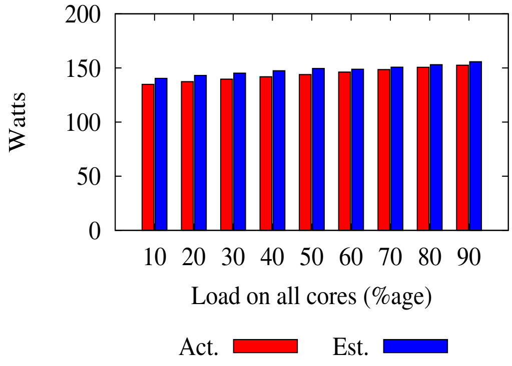

Figure 6: Actual and estimated power consumptions of a tower server with 2.0 GHz frequency.

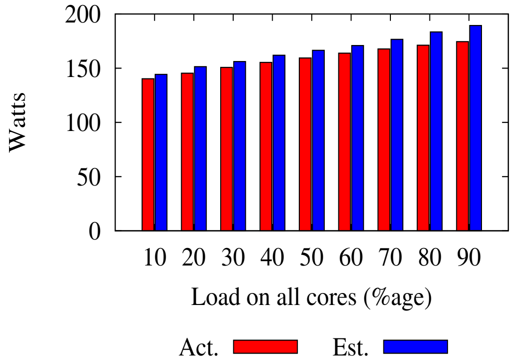

Figure 7: Actual and estimated power consumptions of a tower server with 2.5 GHz frequency.

5. EVALUATION

In this section, we provide a comparison between the computed power (using the power consumption models of Section 4) and the measured power (obtained from a power meter) on two simple lab-grade configurations, which can be considered as the foundation for larger scale results. Next, we present the configured environment as well as the generated workload, and then illustrate the obtained results.

5.1 Setup Configuration

The evaluation results are obtained by carrying out several observations on two different types of servers: *tower* and *blade*. The tower server is equipped with an Intel Xeon L5420 2.5GHz[7] processor, a Maxtor 120 GB hard disk, two

[6]http://en.wikipedia.org/wiki/Blade_server

[7]http://ark.intel.com/Product.aspx?id=33929

Table 3: **Workload of the tower server.**

Test	Test1	Test2	Test3	Test4
CPU load	20%	40%	60%	90%
Memory Usage	1GB	1GB	1GB	1GB
# of HD operations	52	42	32	20
Fan_1 RPM	1183	1183	1278	1249
$Fan_2 \rightarrow_5$ RPM	705	708	796	783

Table 4: Workload of the blade server.

Test	Test1	Test2	Test3	Test4
CPU load	20%	40%	60%	90%
Memory Usage	100MB	250MB	500MB	750MB
# of HD operations	80	80	80	80

DDR_2 modules of fully buffered Hynix[8] memories each of size 4GB, five fans[9] one of which has a width of 70 mm and a depth of 15 mm, whereas the others have a width of 120 mm and a depth of 25 mm, and a power supply unit of 80% efficiency. The HP BL460c G6 CTO blade server consists of two Intel Xeon E5520 2.27GHz[10] processors, two HP 300GB SAS 10K hard disks, six unbuffered DDR_3 modules of HP 2Rx4 PC3-10600R-9 memories each of size 4GB.

To generate corresponding workloads on the servers, the *lookbusy*[11] benchmark is used which induces load on the processor, hard disk and memory simultaneously. The performed observations were divided into three categories: low (between 20% and 40% of CPU usage), medium (between 40% and 65% of CPU usage) and high loads (90% of CPU usage). Tables 3 and 4 summarize the workloads for tower and blade servers respectively. Note that for tower and blade servers, the maximum number of operations per second carried out on the hard disk are respectively 55 and 400 operations.

5.2 Obtained Results

Figures 6 and 7 illustrate the computed (Est.) and measured (Act.) power consumptions for a tower server with 2.0 GHz and 2.5 GHz frequencies respectively. The observations are carried out based on different memory, hard disk as well as processor utilizations. We can notice from Figure 6 that the difference between the computed and measured power consumptions is at most 7 Watts for a CPU utilization between 10% and 50% (an error of 6%). However, such an error reduces gradually to 2% for a CPU utilization more than 60%. We can observe from Figure 7 that the difference between the computed and measured power consumptions is at most 6 Watts for a CPU utilization between 10% and 50% (an error of 6%). However, such an error increases and reaches to 8% for a CPU utilization of 90%.

Figure 8 demonstrates the computed (Est.) and measured (Act.) power consumptions for a blade server with different CPU utilizations. We can notice that the difference between the computed and measured power consumptions is at most 15 Watts for a CPU utilization between 10% and 50% (an error of 9%). However, such an error reduces gradually and reaches to 2 % for a CPU utilization of 90%. It is worthwhile to note that the tower and blade servers have significantly different power consumption for their mainboard (constant value c of Equation (13)).

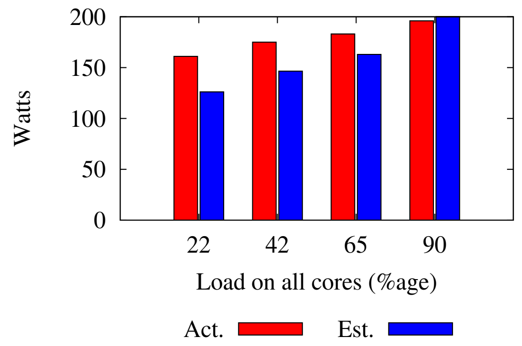

Figure 8: Actual and estimated power consumptions of a blade server with 2.27 GHz frequency.

6. CONCLUSIONS AND FUTURE WORK

Data centres are important contributors to the global energy consumption due to hosting myriad of ICT resources along with their complementary power and cooling equipments. In this paper, we introduced a methodology to derive a generic data centre model that represents the most relevant energy-related ICT resources with their corresponding attributes. Due to complexity, we broke down the design process of the model into four separate modelling phases: ICT resources, Server, Storage and Services modelling. Based on the generic data centre model, we presented models that carry out the power consumption at the server, storage, enclosure, rack, data centre and site levels. We showed that, for tower servers, our models provide an estimation with an error rate of 8% (in the worse case), whereas for the blade servers, the error rate of estimation is no worse than 9%.

Among future works, as mentioned in Section 5 further iterations have to be performed in order to analyze the trial results and refine both the generic data centre model and the power consumption models to achieve better accuracy.

7. ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Community's 7^{th} Framework Programme in the context of the FIT4Green project.

8. REFERENCES

[1] M. Allalouf, Y. Arbitman, M. Factor, R. I. Kat, K. Meth, and D. Naor. Storage modeling for power estimation. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, 2009.

[2] AMD Processors. *http://www.amd.com/us/products/Pages/processors.aspx.*

[3] R. Basmadjian, C. Bunse, V. Georgiadou, G. Giuliani, S. Klingert, G. Lovasz, and M. Majanen. Fit4green - energy aware ict optimization policies. In *Proceedings*

[8] http://www.hynix.com/inc/pdfDownload.jsp?path=/upload/products/gl/products/dram/down/DDR2MODULE.pdf

[9] http://www.delta.com.tw/product/cp/dcfans/dcfans_product.asp?pcid=1ptid=1

[10] http://ark.intel.com/Product.aspx?id=40200

[11] http://www.devin.com/lookbusy/

of the COST Action IC0804 on Energy Efficiency in Large Scale Distributed Systems - 1st Year, 2010.

[4] A. Berl, E. Gelenbe, M. D. Girolamo, G. Giuliani, H. D. Meer, M. Q. Dang, and K. Pentikousis. Energy-efficient cloud computing. In *The Computer Journal*, volume 53, pages 1045–1051, 2010.

[5] A. P. Chandrakasan and R. W. Brodersen. Minimi zing power consumption in cmos circuits. Technical report, University of California at Berkeley, 1995. http://bwrc.eecs.berkeley.edu/publications/1995/Min_pwr_consump_CMOS_crct/paper.fm.pdf.

[6] D. Economou, S. Rivoire, and C. Kozyrakis. Full-system power analysis and modeling for server environments. In *Workshop on Modeling Benchmarking and Simulation (MOBS)*, 2006.

[7] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In *Proceedings of the 34th annual international symposium on Computer architecture*, pages 13–23, 2007.

[8] T. Heath, B. Diniz, E. V. Carrera, J. Wagner Meira, and R. Bianchini. Energy conservation in heterogeneous server clusters. In *Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming*, 2005.

[9] Hynix Semiconductor. *http://www.hynix.com/products/consumer/consumer_sub.jsp?menuNo=1&m=2&s=1&menu3=01&RK=03&RAM_NAME=DDR2%20SDRAM&SUB_RAM=1Gb*.

[10] Intel corporation. *http://www.intel.com/products/processor/index.htm*.

[11] K. H. Kim, R. Buyya, and J. Kim. Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs-enabled clusters. In *Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid*, pages 541–548, 2007.

[12] Kingston. *http://www.kingston.com/hyperx/products/khx_ddr2.asp*.

[13] G. Newsroom. Gartner estimates ict accounts for 2 percent of global. Technical report, 2007. http://www.gartner.com/it/page.jsp?id=503867.

[14] S. Rivoire, P. Ranganathan, and C. Kozyrakis. A comparison of high-level full-system power models. In *HotPower*. USENIX Association, 2008.

[15] Samsung. *http://www.samsung.com/global/business/semiconductor/productList.do?fmly_id=696&xFmly_id=695*.

[16] J. Zedlewski, S. Sobti, N. Garg, F. Zheng, A. Krishnamurthy, and R. Wang. Modeling hard-disk power consumption. In *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, 2003.