Computer Science
Literature Study and Seminar

Energy Efficiency of Virtualizers

Student Name	Student Number (VU)	E-mail
René Aparicio Saez	2558007	2558007@student.vu.nl

Data centers have become one of the largest energy consumers in the world, with about 1.5% of global energy consumption. With the amount of data centers still rapidly increasing, energy reduction seems to become more and more necesarry. This paper reviews the current status (2013 releases onward) in the field of energy efficiency of hypervisors. This is done by looking at a number of aspects. Firstly, new energy estimation models are discussed that are used to estimate power consumptions by virtual machines. Secondly, energy efficiency methods regarding virtual machines in a number of subjects is discussed. These subjects being 'migration methods' and 'resource allocation methods' and 'Frequency scaling related methods'. Finally, a comparison between different hypervisors is made. By doing so, it should become easier to find the best hypervisor for a certain need.

1 Introduction

1.1 Subject motivation

Recently Cloud computing gained a lot of popularity. Both companies and people are seeing the benefits of being able to access their data from any location. This has caused an immense up rise in the amount of data centers across the globe, which comes at a cost. According to a 2007 study performed by Brown et al. [8], roughly 1.5% of the total U.S. electricity consumption was due to data centers, with over 61 billion kilowatt-hours consumed. The study warns for power consumption doubling within the next five years to over 100 billion kilowatt-hours.

A report [16] conducted in 2011 shows that the predictions on the increase of energy consumptions were exaggerated. However, in 2010 the energy consumption of data centers worldwide account for in between 1.1% and 1.5% of total electricity. In the US, data centers even account for 1.7% to 2.2% of total energy consumption. Even though the current growth is not as rapid as it was estimated to be, the numbers are still troublesome. They indicate that data centers are one of the mayor causes of greenhouse gasses and energy consumption in the world. Therefore, something has to be done in order to bring down the energy consumption of data centers.

The 2011 report [16] states that "Growth in the installed base of servers in data centers had already begun to slow by early 2007 because of virtualization and other factors". Virtual Machines (VMs) have become a much used method by data centers. Using VMs allows physical servers to be used to their full extend. However, there is still much to gain in the aspect of virtualization. Two mayor methods exist. The first being Dynamic voltage and frequency scaling (DVFS), which makes use of changing the voltage of a specific component depending on the circumstances it is used for. The second being Vary-On/Vary-Off (VOVO) power management mechanisms, which turns servers on or off to adjust the number of active servers.

Besides these two methods, a large number of other possibilities are still possible. The main issue that rises when searching for other methods is the fact that data centers need to have a certain Quality of Service (QoS) and need to adhere to the Service-level agreement (SLA). Since data centers usually provide a service to their users at any given time of day at any given location, the accessibility has to be high. Therefore, not all energy efficient solutions turn out to be applicable, because they do not keep up with the needed agreements between provider and user.

This paper will provide an overview of other, novel, energy efficient methods to be used by data centers. Since most data centers make use of virtualizers (or hypervisors) in order to manage multiple VMs, research is mainly focused on them. By looking at all the different methods, it will also become possible to provide a comparison between different hypervisors, indicating the advantages and disadvantages regarding energy efficiency for each of them. However, before going into these methods, we will look at new ways to estimate energy consumption of Virtual Machines. Since without proper energy estimation models, found results for new energy efficient methods could become less reliable.

1.2 Literature search

Two methods were used in order to find relevant papers for this literature study. First, two queries were constructed in order to find a large number of possibly relevant papers within Google Scholar. The results from these queries were all put together in a list, containing all their titles. This was done by making use of the FireFox plugin Zotero[5], which automatically takes all paper titles from the current Scholar webpage and puts them in a library. The two queries applied are the following:

Q1 '((intitle:energy OR intitle:power) AND (intitle:efficiency OR intitle:efficient))

AND (virtual machine OR cloud OR hypervisor) AND (-placement -migration)'

Q2 '(intitle:energy OR intitle:power OR intitle:efficiency OR intitle:efficient)

AND (Libvirt OR Xen OR Hyper-v OR Ironic)'

The first query returned 134 results, the second returned 168 results (for papers from 2013 onward, without patents or citations). Any overlap is not taken into account with these numbers. To narrow down the total amount of titles, each title was evaluated manually. If a title showed promise, it was added to a list with plausible titles for the literature studies. After all titles were evaluated, the abstracts of the remaining papers were read thoroughly. If the relevance of the paper was clear, it remained in the list of papers to be used within the studies, otherwise it would be discarded. It should also be noted that due to the addition of '-placement -migration' within the first query, the results did not provide all the necessary papers. This was removed in the second query, thereby providing papers on these topics as well. Additionally to these two queries, some additional papers regarding comparisons between hypervisors had to be found. This was done by picking titles manually using just the query 'comparison KVM Xen energy efficiency'. The following papers came from Q1: [17], [18], [22], [23], [24]. The following papers came from Q2: [12], [19], [25], [26], [27], [28], [29], [30] All the other papers or citations came either from the final query (e.g. [24]), were provided for this research (e.g. [2]) or were added manually as additional means to this literature study (e.g. [1]).

1.3 Thesis structure

The structure of the rest of the paper is as follows. First new methods regarding power estimation are given. Followed by a number of different energy efficient techniques regarding VMs in a number of subjects. Then a number of comparisons between different hypervisors will be discussed, resulting in an overview of all the comparison research that was examined. Finally, a conclusion and discussion section will be provided.

2 Energy Estimation

2.1 Current situation

Most energy estimation models that are used by physical machines in order to estimate the power consumption of VMs are outdated or unreliable. Most models make use of some form of linear estimation. These models consist of a static and a dynamic part. Usually they are of the form $P(t) = P_{static} + \sum_{i=1}^{m} P_i^{vm}(t)$, where P_{static} is the power consumption of the machine when no work is done on the machine, and $P_i^{vm}(t)$ is the power model of individual VM instances and components used by the VM. Each individual VM power consumption is then calculated by looking at its workload per component in some way, mostly in some form of ratio.

This can however be done in an improper way quite easily due to a number of reasons. Not all physical machines cohere to a linear model, and not all VMs run on just one single physical machine. Additionally, the ratios for each given component might not be correct, or might fluctuate a lot causing unreliable estimations. Additionally, models might only be applicable to certain types of hardware, e.g. uniprocessor systems only. Because of this, new models have to be found in order to estimate power consumption of VMs in a more reliable manner. Preferably, new models disregard any difference in hardware architecture, in order to be applicable in any situation.

2.2 Novel Estimation models

Wassmann et al.[26] give a more sophisticated energy estimation model. In order to do so, a Xen hypervisor is used per physical machine. Three components of each physical machine are monitored in terms of energy usage. Namely the resources processor (CPU), hard disk drive (HDD), and network interface controller (NIC), since these use the most power of the physical machine. All power consumptions of these components can be collected by the hypervisor fairly easy. Additional to this, each machine has an idle power consumption that is taken into account as well. Instead of looking at all components and creating a linear model from this, the authors of this work state "the consumption of components is not always a simple function of the load, our energy model will approximate the energy consumption by polynomial regression methods with varying degrees". This approach is mainly credited to the fact that many current processors make use of some integrated overclocking mechanism, therefore becoming non-linear.

By using polynomial methods with varying degrees, these model should theoretically fit any type of energy consumption model. The parameters of the model are estimated in two phases. The first phase is called the *init phase.* This phase is started before running any VM during the startup of the machine. During this phase synthetically varied load is used to find the power consumption in different situations. From this the initial values for the parameters of the model are set. The second phase, called the *runtime phase*, measures real VM loads in order to acquire VM specific parameters. This is done, because loads vary over time and could be non-linear. In order for the physical machine to stay up to date with its model, the model is changed whenever a new VM is added. This is done by looking at "the resource utilization of each VM, as well as the corresponding total power consumption of the system, will be tracked for a certain time".

When testing this new model on a variety of hardware, the model estimated energy consumption very well. Even integrated overclocking mechanisms were estimated in a fairly reliable way by the model, changing from a first order polynomial (linear model) to a sixth order polynomial for a AMD Phenom II system with Turbo Core technology off and on respectively. This can be seen in figure 1, showing the results after running the initial phase compared to the actual power that is being consumed.

As mentioned, most research makes use of linear models in order to estimate energy consumption. However, as can be seen from the results when using this novel non-linear approach, some hardware might even need up to a sixth order polynomial model in order to correctly estimate power consumption, due to the fact that current hardware makes use of things like integrated overclocking mechanisms. This new method therefore appears to be a more reliable way of estimating energy consumptions, disregarding any hardware specific parameters.

While this research was conducted using the Xen hypervisor, it only makes use of collecting status information of certain main resources. Therefore, it should be possible to apply this estimation model to other hypervisors than Xen.

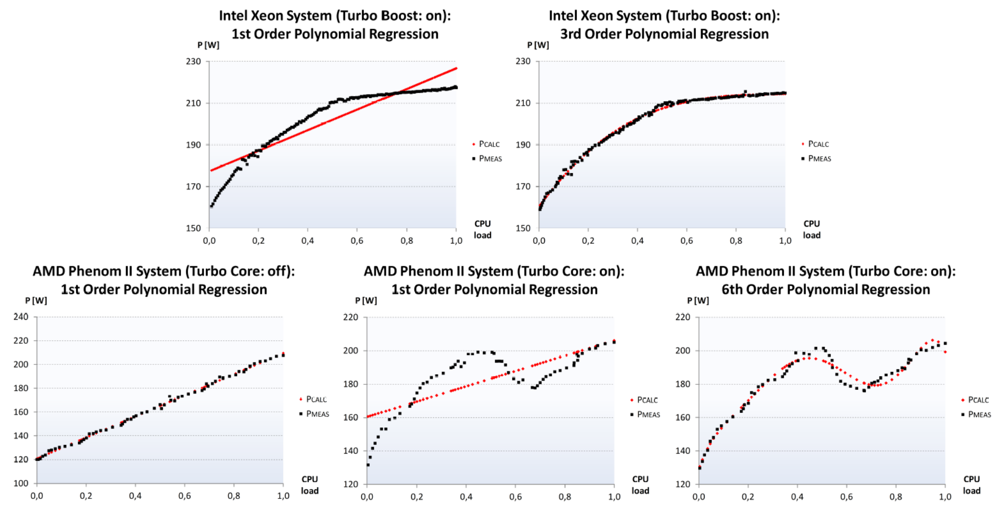

Figure 1: Init phase estimation results

In another study, Xiao et al.[28] also take recursive power consumptions into account. For example, most power consumption from I/O requests occur when encrypting or decrypting. This power estimation is not always taken into account in a correct manner. Models therefore show more error than they need to. They call this relative performance monitoring counter credit scheduling (rPMC-CS) which is based on the modelling technique used by relative performance monitoring counter (rPMC). A number of candidates to estimate the power are chosen, in this study these are: uOps, Halt, LLC, TLB, DMA, FSB, Interrupt. These candidates will be used to estimate power consumption of four components, namely: processor, memory, disk and I/O controller. By running benchmarks, a relation can be found between these candidates and power consumption. For each component, candidates are found that have a relation to the power consumption.

The novel idea comes from taking into account the recursive energy consumption that happens after certain events. This is done by using a method called credit scheduling (CS). This method provides fair sharing of CPU time among virtual machines by a credit-based mechanism. When all credits are spent, access to a CPU can only be achieved by recharging credits. If a recursive operation is made, its future processor utilization ratio will be increased. Thus allowing the hypervisor to know more energy is going to be used by the VM.

Experiments are run making use of a Xen hypervisor. When running a benchmark in order to test the energy estimation, the TPC-W benchmark [4] shows a power estimation error reduction from 11.9% to 5.4%. This is quite a huge error reduction and shows that this method could be used to estimate power consumption more accurately.

While both these methods provide great additions to energy estimation models, they are fairly new. Therefore, the newly defined methods do not appear throughout other literature in this study. Future work however should most definitely take the observations that were done into account, in order to provide better and more reliable results.

3 Energy Efficiency

When looking at the literature, there are a few main subjects that researchers focus on regarding energy efficiency using virtual machines. These being 'migration methods' and 'resource allocation methods' and 'Frequency scaling related methods'. Therefore, these different subjects will be discussed separately.

3.1 Migration methods

When VMs are running for a long period of time, it could become a necessity to move the workload of these VMs to another one. This is done by a process called migration. While some literature regarding migration is mainly intended on enhancing performance, migration may also help to achieve a better energy efficiency. When less time is needed to migrate, less energy has to be consumed doing so, therefore helping in lowering energy consumption.

Migration is mostly done by using the pre-copy algorithm. This algorithm copies over files or memory pages in an iterative way. When a certain threshold is met, the part that is not yet copied from the old VM's memory is copied in its entirety. However, as stated by [19] "for write-intensive workloads, the pre-copys straightforward iteration strategy will become inefficient". This is due to the fact that copied pages could be 'dirtied' because they are overwritten. The algorithm then tries to retransmit the page, which could cause it to take a lot of time before it is properly transmitted without becoming dirtied during the process.

Ruan et al.[19] proposes a new method in order to compensate for this migration problem. This is done by creating a dynamic forecasting model based on the analysis of frequently modified pages. The implementation makes use of a transition state model that analyses the memory write pattern in a short period of time. By using this model, the pre-copy algorithm is enhanced in order to compensate for problems with write-intensive workloads. This state transition model has the Markov property, and therefore does not know any states it was in beforehand. The algorithm makes use of the notion of temporary and spacial locality. Temporary locality states that if a page is currently being accessed it will also be accessed in the near future. Spacial locality states that if a page is accessed, pages nearby will likely soon be accessed as well. Over a longer period of time, this algorithm can then find hotspots, which are considered the states of the transition model. From these hotspots, rarely used pages have to be subtracted, otherwise too many pages will be filtered. This is done by looking at the Local Writable Working Set, which only looks at accessed pages for a short amount of time. If a page is infrequently accessed, it will not become part of this set, and will thus not be filtered.

Experiments using the Xen hypervisor show that using 256 states turn out to be the optimal amount. Experiments look at three aspects, namely: downtime, migration time and amount of migrated data. The latter looks at the total amount of copied pages, including all the retransmitted dirty pages. Experiments compare the 'old' pre-copy algorithm with the enhanced 256 state pre-copy algorithm. While the downtime of the enhanced algorithm is somewhat higher for five out of six benchmarks, with roughly 5.5% on average increase in downtime, the migration time is always lower. The increase in time is not as high as expected due to the filter algorithm slowing it down. Additionally the amount of pages being transmitted using the new algorithm is much lower than with the old one. Thus, the new algorithm works as intended when filtering out dirty pages and migrating pages quicker than before. The results can be seen in figure 2

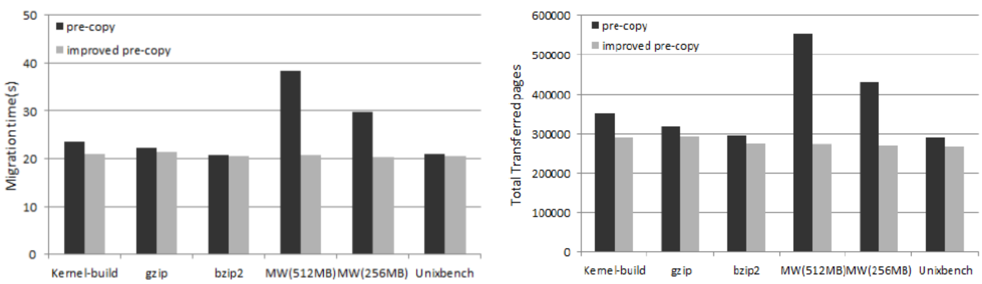

Figure 2: Comparison between a number of benchmarks for the pre-copy and improved algorithm, (left) migration time, (right) number of total transferred pages

Another aspect of migration is locating the optimal placement in a fast way. When selecting a new host, multiple possibilities of placement exist. However, one option might not be as energy efficient as the other. Therefore, choosing the most energy efficient placement is beneficial for a data center. Placement is mostly done by some heuristic. Zhao et al.[30] propose a new heuristic, called PS-ES. This heuristic makes use of a combination of particle swarm optimization (PSO) and simulated annealing (SA).

By looking at a time window, SA is put into a PSO algorithm in order to find out the better global optimal solution. However, in longterm view, this solution might turn out to be a local optimum. But since no prediction about the future can be given correctly, this suboptimal solution is chosen. It should be noted that when choosing the location to migrate too, it is not allowed to be migrated to the host it is currently being moved out of.

This method was compared to random migration policy, the optimal migration policies based on PSO and the optimal migration policy PS-ABC. This was done using the CloudSim platform [9]. CloudSim is an event driven simulator, that can be used to calculate energy consumption by a data center during some simulation period. This is done by calling the getPower() function.

When comparing it to the other three approaches, PS-ES consumes less energy over a four week period of time. This is mainly due to the fact that PS-ES enhances itself overtime. The results indicate that power consumption is lowered over time using this policy, whereas all other policies perform at the same rate at every moment. The simulated failure rate of the policy performs about the same as PS-ABC, and they both perform much better than the two other policies. Also when applying different workloads over time to each different policy, PS-ES has less incremental energy consumption than any other policy.

While this paper does not focus on meeting SLA specifications, it does state that "the Multiple-SS power management policy leads to the better trade-off between energy cost and penalty cost as well as have the relatively better total cost of both them".

In another approach, Zhao et al.[29] proposes MOGA-LS, a heuristic and self-adaptive multi-objective optimization algorithm based on the improved genetic algorithm (GA). The approach makes use of Pareto non-dominated sorting, individual density estimation, and mathematical statistics of Pareto optimal solutions to achieve a long-term excellent power saving and load balancing optimization. Pareto efficiency states that, no resource allocation can be made in order for one subject to be better off while another is worse off than before. In GA, possible candidates evolve using genetic operators. Solutions contain certain properties that can be combined, mutated or altered in order to create a new set of solutions named generation.

Pareto solutions can be obtained, but instead of choosing one of these solutions at random, mathematical statistics theory is used to find the optimal location selection. This is done by using a probability wheel for each VM. Two solutions are compared using the improved GA-based approach. The tournament selection operator is used. This operator chooses two groups from the possible Pareto solution set at random. Each group consists of two individuals. From these four individuals, two will 'win'. The winners will undergo the crossover operation, after which it undergoes some mutations. This can be seen as a set of chromosomes that are combined to build a new combination. The idea here is that performing these steps will allow for better better global search ability in the early iterations and better local search ability in a later phase. The mutation operators can be modified, making this more dynamic and self adaptive.

Experiments are again conducted using the CloudSim platform. Comparisons are made against random migration policy, dynamic load balancing (DLB) and the StdPSO migration policy. Results indicate that MOGA-LS and StdPSO migration policies consume relatively less power, this is mainly due to the fact that these approaches focus on lowering energy consumption. While StdPSO consumes slightly less energy over time, MOGA-LS also takes into account load balancing. When comparing the standard deviation for load balancing, MOGA-LS has a much lower standard deviation. Especially when compared to StdPSO. When looking at the amount of migration failures versus the amount of migration requests, it becomes obvious that MOGA-LS has fewer failures compared to other approaches. StdPSO having almost double the amount of failures compared to MOGA-LS in the results.

A downside to this algorithm is the trade-off between taking into account SLA versus power consumption costs. A decision has to be made about what penalty is allowed to happen regarding the SLA compared to the additional energy consumption costs this might add. As stated by the author "the multiple-SS policy gives the optimal cost trade-off, relatively".

In another study, Valliyammai et al.[23] optimize workload distribution in order to have as few machines running as possible. Distribution optimization is done by a fuzzy controller, making use of a vector packing algorithm. The fuzzy controller analyses energy consumption per VM, and puts them in one of three energy clusters. These three clusters being high medium or low energy clusters. The membership matrix is constructed by applying the fuzzy C-means algorithm, providing a value between zero and one. One meaning that the particular energy value belongs to that cluster.

In order to adhere to the SLA, jobs are shifted from one energy cluster to another if SLA degradation is too high. By moving VMs from low to a medium energy cluster, or medium to high energy cluster SLA degradation is minimized.

The vector packing algorithm is used to allocate jobs per host. Jobs with the highest resource needs are placed on the hosts with the lowest maximum power consumption. Suspended VMs on these hosts are separated into two categories, namely hot and cold. Hot VMs have only been suspended for a short amount of time, while cold ones have been suspended for a longer period of time. When a job is being allocated, cold VMs are preferably chosen. This is done in order to overcome overheating of the hot VMs, thus lowering carbon emission. When no cold VMs are available, hot VMs are chosen.

Results using the CloudSim platform indicate that the vector packing algorithm reduces energy consumption. The energy consumed by the proposed system is 16.04% less when compared to the normal scenario.

In another study, Xiao et al.[27] proposes a heuristic in order to provide energy aware scheduling for data intensive workloads. Energy Consumption in Deployment and Scheduling (MECDS) algorithm, two phases are performed. The first phase configures VM instances. Secondly it schedules activities to VM instances based on combining execution-dependency and the per-VM power model together.

The main part of the algorithm, called Minimal Data-Accessing Energy Path (MDEP), selects a storage node which can be allocated to a VM instance aiming to obtain minimal data accessing energy consumption for the current activities. It is defined as the minimal energy consumption from the moment of initialization to the current activity. If a node satisfies this minimum, it should be chosen as storage node.

Experiments are conducted on the CloudSim platform and on a real cloud platform using the Xen hypervisor. The CloudSim results mainly focus on energy consumption for data intensive workloads and energy performance. The test run makes use of a parameter that sets the data work flow between VMs. Energy efficiency of the MECDS algorithm are not that great when the data workflow parameter is low. However, when the parameter is increased and the data workflow increases, energy consumption of the MECDS algorithm improves drastically compared to other algorithms. When performing tests on an actual cloud platform, results remain the same. Showing that the algorithm works well for the subject it was designed for.

A combination of the novel migration filter combined with one of the new heuristics for choosing a new host to migrate too, might proof to be an even more energy efficient approach. However, most of the proposed heuristics have only been tested on the CloudSim platform, which might proof to have different results than really is the case. Therefore, these heuristics should first be studied in a real environment. Finally, as can be seen from the different heuristics studies, finding a good approach for migration that coheres to the SLA is not easy. Performance remains the main aspect to be taken into account. Since in the end, clients still prefer performance over energy consumption. Some studies do not even specify any information regarding cohering to SLA agreements.

3.2 Resource allocation methods

As an extension to the migration problem, some research is done in the field of resource allocation. Resources can be allocated between different hosts in a smart way, reducing energy consumption in the process. In order to do so, migration might be needed in order to find the optimal solution. The research in this field is mainly focused on the allocation of the resources, but subsequently lowers the amount of migrations.

Wang et al.[25] propose a two dimensional greedy type heuristic algorithm based on CPU and memory utilization to implement the mapping from the VMs to hosts. The study is carried out with the Amazon EC2 in mind, thus having Xen as virtualizer. The focus is set on centralized job scheduling strategies. A two-level control is applied, namely: "application workloads to resource requirements through a global controller at the resource-pool level and the mapping from virtual resources to physical resources through local controllers at the virtual-container level". A job arrives at the global controller and is queued. When enough resources are available to be allocated the job is dispatched to a server.

Since the energy minimum problem is a NP-hard problem, it will be hard to find the optimal solution. Instead a suboptimal solution through the heuristic is found. Since not all servers have to be utilized at all time, a quadratic exponential smoothing method to predict the workloads is used, allowing for servers to be turned on or off or set to idle mode. VMs are picked by looking at the ratio of available CPU and memory resource. If the ratio is lower than 1, a VM with max memory is chosen. The other way around, with a ratio large than 1, a VM is chosen with max CPU demand.

Migration takes place if certain thresholds are passed. If for example the CPU utilization becomes too high, some VMs have to be migrated in order to lower the utilization. On the other hand if there is too little CPU utilization, all VMs have to be moved to another host so it can be put into a sleeping state.

When comparing results to other work, namely MBFD[6], results indicate that this new algorithm consumes less energy for any amount of CPU utilization, with about 12% less energy consumption on average. The new algorithm also has less SLA violations compared to the MBFD algorithm, with around 5% less violations. Additional to this, the proposed algorithm also utilizes more active resources than the MDFB algorithm.

It should be noted that in this research a linear energy model was used. This in comparison to the arguments as stated before that current day processors might not follow a linear energy model. Results could therefore be less reliable. It is also curious to see that results are only compared to one specific other study, of which this work is not even based on (beside being in the same field of research).

In another study, Portaluri et al.[17] proposes a solution based on genetic algorithms. The allocation approach performs joint allocation of computational and network resources, in order to find a trade-off between task completion time and energy consumption. The study is considered homogeneous single-core resources. The algorithm was implemented using a specific java based open source framework called jMetal [11].

As before, the combination of Pareto solutions and GA is used. By using the genetic algorithm, a set of Pareto solutions can be found. By using the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [10] heuristic, the solutions are ranked. 'Chromosomes' from the best ranking solutions are taken in order for the next iteration to find better solutions.

Results are not compared to other research or approaches. However, from the results it can be noted that this approach is very scalable. Results show that the optimal Pareto solutions are found over time (for the same task). Increasing the amount of servers to perform a task increases the amount of consumed energy, but lowers the time needed to perform a task, as can be expected. From the Pareto solutions a trade-off between performance and energy consumption can be chosen.

3.3 Frequency Scaling related methods

A much used method for energy efficiency is Dynamic Voltage and Frequency Scaling (DVFS). This method changes the operating frequency and voltage of a given computing resource. By doing so it consequently reduces power consumption. Tesfatsion et al.[22] combine this method together with changing the number of virtual machines and the number of cores in order to achieve even higher energy efficiency.

The combination of these three methods are described as follows. Horizontal scaling is used to add or remove VMs for a running service. Vertical scaling changes the amount of resources being used, like cores or memory. Hard power scaling utilizes DVFS to reduce power consumption. These methods are being applied during runtime.

Experiments are conducted on KVM version 1.0.0. KVM has the downside of not being able to change the number of cores assigned to a VM dynamically, thus impacting vertical scaling. The experiments are run by combining from 1 to 10 VMs, a number of cores that range from 1 to 32, and server CPU frequencies running at 1.4, 1.5, 1.7, 1.9, and 2.1 GHz, resulting in 3885 results.

Results indicate that the major impact on consumption is the amount of cores that is being used. The total energy saving achieved peaked on 34%, compared to constituent approaches, while meeting performance targets.

In another study, Ren et al.[18] evaluate a multi-objective evolutionary game theoretic framework for adaptive and stable application deployment in clouds that support DVFS. They call their approach Cielo. It maintains a set (or a population) of deployment strategies, each of which indicates the location of and resource allocation for that application. Through a series of evolutionary games, a stable equilibrium is reached regarding resource allocation.

Because of its evolutionary approach, Cielo learns over time and becomes more and more effective until it finds an optimal solution for a given objective in aspects of: CPU allocation, bandwidth allocation, response time, power consumption. By starting its evolution game considering the entire population of allocation options, the algorithm randomly selects pairs form this population. The loser is removed and the winner gets replicated into the population, thus increasing its population share. The winner gets mutated with some probability P_m, in order to look for better solutions. By repeating the game, an optimal solution should remain in the end.

Experiments were conducted using JavaVM 1.7. When comparing Cielo to the known algorithm NSGA-II, Cielo yields the best performance in response time and bandwidth allocation. Cielo's average stability is 29.32% higher than NSGA-II's. Most importantly, Cielo consumes less energy than NSGA-II both when using DVFS and not using DVFS.

Results from this study indicate that the [17] study could benefit from this algorithm, since it made use of the NSGA-II algorithm and Cielo appears to be better. When using Cielo instead, better results might appear faster, resulting in even more efficient resource allocation.

4 Hypervisor Comparison

A variety of hypervisors are available on the market, both open source and commercial products. In the open source section Xen(server) and KVM are the most popular. In scientific literature, Xen and KVM are usually used, mainly due to the fact that they are open source and can thus be tinkered with. Commercially a number of hypervisors are available, of which VMWare ESX/ESXi and Microsoft Hyper-V are the two most commonly used. While Xen, KVM and VMWare run on Linux, Microsoft's Hyper-V (for obvious reasons) runs on Windows. Hyper-V can run on Linux, but it is more optimized to run on Windows.

All four of these hypervisors are so called type-1 or bare metal hypervisors. This means that they run directly on the hardware, instead of on top of an operating system. It is disputed whether or not KVM is a type-1 or a type-2 hypervisors. While some literature just accepts KVM as being a type-2 hypervisor, others state that, against popular believe, KVM is also a type-1 hypervisor.

When looking at the literature, the Xen and KVM hypervisors appear to be the hypervisor of choice. While most research mostly focus on just one hypervisor, sometimes comparisons will be made between performance for different hypervisors. Comparisons are mostly made between the two open source hypervisors and either VMWare or Hyper-V as a third. As stated by [24] "it is worth mentioning the difficulty to find in the literature fair comparisons of all these hypervisors".

The open source hypervisors are more popular because of different reasons. Xen has been around for a fair amount of time, and was the only big open source hypervisor for a long time. Additionally Xen is the driving force behind Amazons EC2 platform. Therefore, most researchers preferred to use Xen. However, as of 2007 KVM has become available within the Linux Kernel. Because of this, KVM gained a lot of popularity within the community, though not as much in enterprise customers. Enterprises tend to use VMWare ESXi instead. Windows Hyper-V remains the least favorite of these four and is not recommended by [2] commercially.

Varrette et al. [24] provide a study comparing Xen 4.0, KVM 0.12, and VMWare ESXi 5.1 from a High Performance Computing (HPC) perspective. This is done on Grid'5000, which is a scientific instrument for the study of large scale parallel and distributed systems. By testing the High Performance Linpack (HPL) reference benchmark, a comparison is made between the different hypervisors.

When comparing the benchmark run per hypervisor against a baseline as was obtained by using a non virtualized Grid'5000 cluster indicates that, for each hypervisor, the performance efficiency is not as good as the baseline. This is due to the performance being bounded by the maximal capacity of the physical host running the hypervisors.

When just looking at the performance of the hypervisors, VMware ESXi performs better than Xen and KVM. However, VMWare is bounded to a limited amount of VMs, namely four, after which it became unstable. Both Xen and KVM on the other hand remain stable throughout the entire benchmark, and are thus more scalable. Results also indicate that performance is lost when more than 2 VMs per physical host are used for VMWare ESXi or more than 4 VMs per physical host for Xen and KVM. When just comparing Xen and KVM, Xen has a higher performance rate throughout the benchmark.

When comparing energy efficiency, the hypervisors turn out to be better than the found baseline measurement. All hypervisors use less energy per Floating Point Operation (Flop) than the baseline measurement. The measurements are done in a metric called 'Performance per Watt' or MFlops/Watt. KVM performs worst, always having less than 300 MFlops/Watt throughout the benchmark. Xen roughly achieves 400 MFlops/Watt, peaking once at 650 MFlops/Watt in the benchmark. VMWare, when results were gathered during its stable moments, performed the best, always having over 500 MFlops/Watt. While HPC may not benefit much from virtualization at this point, the results do indicate that virtualization is more energy efficient.

A number of studies perform the same performance benchmarks in order to provide a comparison between hypervisors. However, due to the fact that all studies are carried out using different hardware, each study is assessed separately. However, from the results a general conclusion on the different hypervisors can be made.

Guzek et al.[12], the same authors that performed the previous study regarding hypervisor efficiency in a HPC environment, test a number of other benchmarks on two grid'5000 clusters, each containing different hardware. Using the same hypervisors as before, they run three other benchmarks. Two of which return in other literature, namely Bonnie++[1] and IOZone[3]. Bonnie++ is a benchmark that is aimed at performing a number of simple tests of hard drive and file system performance. IOZone is a more complete cross-platform suite that generates and measures a variety of file operations. I/O is the most problematic aspect for virtualization. Therefore, this benchmark is very important to take into account.

Results for each benchmark vary for each cluster. One cluster with Intel hardware shows that from the hypervisors for Bonnie++, VMWare ESXi is the fastest and most energy efficient, followed by KVM and Xen in the last place. However, on another cluster with AMD hardware flips the results around, with Xen the fastest and most energy efficient followed by KVM and VMWare ESXi in the last place.

For IOzone the Intel cluster is also the fastest using the VMWare ESXi hypervisor, while Xen and KVM perform roughly the same. On the AMD on the other hand, results are not flipped around. VMWare ESXi remains the best performing hypervisor, followed by Xen and KVM in the last place. These results indicate that hypervisors are very architecture and application dependent, meaning that a hypervisor should be chosen depending on what it is going to be used for.

Another study by Soriga et al.[21] also runs the IOzone benchmark, this time not making use of a HPC cluster. This study however only compared the Xen with the KVM hypervisor. Results form this study also vary depending on how the benchmark is set up.

When using a single VM running on the hypervisor, Xen slightly outperforms KVM on most of the benchmark. However, when increasing the number of VMs to 11, KVM outperforms Xen in the entire benchmark. The authors do mention that KVM used to perform worse regarding IO operations in previous versions of the hypervisor. However, this version (0.12.1) has overcome most of these problems.

Jaikar et al.[14] examine another performance comparison between Xen and KVM. This time not only looking at the IOZone benchmark, but also Bonnie++. The benchmarks are tested on two different setups, where the guest OS is changed around.

When looking at the results from IOZone, for all cases KVM is outperformed by Xen.

This happens for both setups. For Bonnie++ KVM also performed worse than Xen. This might be due to the old version of KVM that was used, namely KVM 83 which dates back to 2009. The results from the previous studies all mention newer version of KVM where IO operations have been improved. However, this study also makes use of an older version of Xen, namely version 3.03. It is however also unclear what the differences in IO throughput are with different versions of Xen.

Another Bonnie++ benchmark test is performed by Hwang et al.[13]. This study takes a look at four hypervisors. Namely Hyper-V, KVM, vSphere and Xen. vSphere will however not be discussed in the scope of this paper. The most recent versions of the hypervisors were used at that time. Namely KVM 2.6.32, Xen 4.1.2 and Hyper-V 2008 R2.

The results indicate that KVM performs the best throughout different tested operations, in a steady way. Hyper-V also performs steadily throughout the tests but at a lower rate than KVM. Xen does not perform in a steady way. Output operations are done very poorly. For input operations and random seeking, results come close to that of KVM.

Apart from testing virtualizers for general or HPC problems, Shirinbab et al.[20] compare the performance of three hypervisors regarding a telecommunication application. KVM, VMware and XenServer are compared in regard of CPU utilization, disk utilization and response time of a large industrial real-time application. This is done on two servers connected to one another with a 1 Gbit fibre cable.

Performance is tested by varying the amount of cores within the setup, testing for 6, 12 and 16 cores. Disk utilization is tested by varying loads, testing for 500, 1500, 3000, 4300, 5300 req/s. It should be noted that Xen could not cope with 5300 req/s.

Results indicate that Xen has the highest CPU utilization of approximately 80% when using any amount of cores. Xen's CPU usage increases faster compared to the load it has to handle, compared to KVM and VMWare. For 16 cores, VMWare and KVM perform roughly the same. However, when them amount of cores gets lower, KVM performs better with higher loads, where VMWare performs better with smaller loads.

Disk utilization is the highest for VMWare for any amount of cores. Xen comes second when only 6 cores are used. KVM on the other hand is better than Xen for 12 or 16 cores.

While most papers comparing hypervisors do not take energy efficiency into account that much, Jin et al. [15] provide an empirical investigation of virtualization on energy efficiency. This is done by looking at the energy usage in servers under different task loads. The hypervisors that are compared are Xen 3.0.3 and KVM 83 which, as stated before, are older versions of these particular hypervisors.

In order to get measurements, the task is given to calculate the value for π up to 100.000 digits after the decimal point. When looking at the idle amount of energy that is being consumed, the power consumption in Watts is lower for the Xen hypervisor, both for using 2 VMs as 3 VMs. The consumption does not seem to increase much when an additional VM is used. KVM on the other hand consumes more power and does show a visible increase in power consumption when 3 instead of 2 VMs are used.

When performing the task of calculating π, the average power consumption in Watt is slightly higher for Xen than the KVM hypervisor. Average power consumption when using 2 VMs is also higher than 3 VMs, which is curious. Xen performs the task the fastest. Because of this, the Xen hypervisor uses less overall power compared to KVM that needs much more time to complete the task. Xen takes 10% less time than KVM on average and as a result uses roughly 11% less energy.

Table 1 holds information on what is tested per paper and can be used to compare each literature somewhat easier with one another.

Table 1: Hypervisor Comparison Table

Literature	**Hypervisor(s) tested**	**Focus of paper**	**Benchmarks tested**	**Main results**	**Taken SLA violations into account**
[2]	Xen, KVM, VMWare ESXi and Windows Hyper-v	Comparing hypervisors in general	none	For enterprise purposes Xen and KVM are most recommended since both hypervisors are supported by a large number of users.	SLA is not mentioned throughout the entire report.
[24]	Xen 4.0, KVM 0.12, and VMWare ESXi 5.1	Comparing hypervisors in a HPC environment (Grid'5000), to see if they can be used for HPC purposes	HPL benchmark	VMWare performs best (650 MFlops/Watt), but becomes unstable with many VMs. Xen has the best performance rate throughout the entire benchmark (500 MFlops/Watt). KVM performs the worst (300 MFlops/Watt). Overall Virtualization is more energy efficient but has a drop in performance and should therefore not be used for HPC purposes at this point.	SLA is not mentioned throughout the entire paper.

[12]	Xen 4.0, KVM 0.12, and VMWare ESXi 5.1	Comparing hypervisors in different HPC environments (Grid'5000)	Bonnie++ and IOzone (other benchmarks were tested but not discussed in this paper)	An Intel hardware based cluster resulted in VMWare performing best for Bonnie++ followed by KVM and Xen in the last place. In an AMD based cluster the results were flipped around for Bonnie++. For IOZone, VMWare performs best in both clusters. In the Intel cluster KVM and Xen perform roughly the same while in the AMD cluster Xen ourperforms KVM.	SLA is not mentioned throughout the entire paper.
[21]	Xen 4.2.2 and KVM 0.12.1	Comparing performance between hypervisors	IOzone	Xen outperforms KVM with less than 11 VMs, after which KVM starts outperfoming Xen.	SLA is not mentioned throughout the entire paper.
[14]	Xen 3.03 and KVM release 83	Comparing performance of hypervisors on different setups	IOzone, Bonnie++	Xen outperforms KVM throughout all benchmarks. This might be due to the old version of KVM that was used.	SLA is not mentioned throughout the entire paper.
[13]	Xen 4.1.2, KVM 2.6.32 and Windows Hyper-V 2008 R2	Comparing performance between hypervisors	Bonnie++	KVM performs best in this study while being stable throughout the benchmark. Hyper-V performs in a steady way troughout the benchmark as well. Xen performs well for Input operations but performs poorly on the output part of the benchmark.	SLA is not mentioned throughout the entire paper.
[20]	Xen 3.0.13, KVM (version unclear) and VMWare 5.0	Comparing hypervisors regarding a telecommunication application	none	Xen has the highest CPU Utilization, followed by KVM and VMWare which perform roughly the same. Disk Utilization is the highest for VMWare followed by Xen and KVM respectively.	SLA is not mentioned throughout the entire paper.

[15]	Xen 3.0.3 and KVM release 83	Comparing energy efficiency of hypervisors	Calculating 100.000 digits of π	Average power consumption is higher for Xen than for KVM. However, Xen performs the task faster, resulting in less overall power consumption compared to KVM. Xen is 10% faster and uses 11% less energy.	SLA is not mentioned throughout the entire paper.

5 Discussion

While the novel energy estimation methods as described pose a more sophisticated and reliable way of estimating power consumption, they are not widely adopted. Therefore, results as given in other literature, that make use of older, linear power models, might not be as reliable as suggested by their authors. This also goes for researched energy efficiency methods that could be used in combination with the work provided in some studies. The main example being the Cielo algorithm outperfoming NSGA-II, the algorithm that was used by [17].

One major point of discussion is the use of older hypervisors that were used in research. While some could be related to the fact that research was most likely conducted on Amazons EC2 platform, that utilizes the older Xen 3.x version, other hypervisors also appeared to be outdated in some literature.

Some literature only researched their proposed algorithms using simulation software. CloudSim being the platform mainly utilized for this. While simulations could indicate how a certain algorithm might run, it does not provide real-life confirmation that the algorithm will work out in the field, running actual workloads on actual machines.

Another aspect that was missing in most literature was information about cohering to SLA agreements and QoS agreements. These agreements between the provider and clients are either not discussed or only slightly discussed in literature. This makes it difficult to see if the provided energy efficient methods can actually be applied in actual data centers. The fact of the matter is that data centers want to provide the best performance, since this results in more customers being satisfied with the service. Energy efficiency is seen as important but should not result in much loss of performance. It can be disputed if this way of thinking is good or not, since energy efficiency is and will remain a major problem in the future.

As can be seen when looking at the impact factors per journal in the Appendix 7, most literature that is relevant to this topic comes from conferences and symposiums. Reuters does not provide any impact factors regarding conferences and symposiums. Most conferences however appear to be held by trustworthy publishers like IEEE and ACM. It should also be noted that the journals: [14][19][29] could proof to be untrustworthy. They are not listed with a Reuters Impact Factor, or were removed from the Impact Factor list. Therefore, the results and research conducted by them should be considered with caution.

Interestingly, most research regarding this topic appears to come from Asian universities. The reason for this is unknown, since a multitude of universities participate in this research, compared to all research coming from a handful of universities.

As a final remark it should be noted that in hindsight, the queries used to find papers could have been improved. Instead of searching for ‘KVM’, the more general term ‘LibVirt’ was used, which is a management tool for KVM and other hypervisors. By using the more specific term ‘KVM’, more relevant results might have been found.

6 Conclusion

Since energy consumption by data centers is becoming larger every year, solutions have to be found regarding energy efficiency in data centers. By looking at some recent research, an overview was provided of the current state of possibilities regarding energy efficiency for data centers, virtual machines and hypervisors.

This paper started by looking at possibilities of enhancing power estimation models in section 2, since most research make use of linear-models to represent power consumption. As explained, non-linear power consumption is becoming more common, due to hardware making use of complex internal systems like integrated overclocking. [26] posed a new energy estimation model, disregarding hardware, allowing it to be applicable on any type of machine. In addition [28] showed that recursive energy usage could be taken into account within energy estimation models. Results show that the error in the models can be significantly reduced.

Current research regarding reducing energy consumption when using VMs, focuses on a number of subjects. These subject being ‘migration methods’ and ‘resource allocation methods’ and ‘Frequency scaling related methods’, which were discussed in section 3.

Five studies ([19],[30],[29],[23],[27]) regarding migration were discussed. One providing a way of enhancing migration speed for migrations having frequently modified pages. The others discussing new migration policies. These migration policies differ from a vector packing algorithm to genetic algorithms, indicating that a large variety of possibilities regarding migration policies is being studied.

Then two studies ([25],[17]) were evaluated regarding resource allocation. One providing a two dimensional greedy type heuristic solution for resource allocation regarding CPU and memory. The other provides a genetic algorithm approach in order to allocate resources.

Two studies ([22],[18]) regarding frequency scaling related methods were evaluated.The first utilizes DVFS together with changing the number of virtual machines and the number of cores in order to provide optimal energy efficiency. The number of cores being the main factor of consumption according to this research. The other, called Cielo, makes use of an evolutionary approach combined with DVFS to lower power consumption. Cielo also turns out to be more stable than NSGA-II, indicating that Cielo could be adopted in other studies instead of NSGA-II.

Evolutionary and genetic algorithms overall seem to have the main focus in research. These algorithms show good results for any of the three defined subjects regarding energy efficiency. This is mainly due to the evolutionary way these algorithms work. Because of this the algorithms return better results over time and will find good suboptimal results for the given energy efficiency problems posed to them.

Finally, in section 4, literature comparing hypervisors is discussed. From all the literature regarding comparisons between hypervisors, it becomes clear that it is very hard to choose one optimal hypervisor. Most research is being done regarding the Xen and KVM hypervisors. From the commercial hypervisors the VMWare seems more popular than the Hyper-V, this is mainly made clear due to the fact only one study discussed the Hyper-V compared to multiple studies discussing VMWare.

When looking at the actual results found in the studies, different versions of the hypervisors should also be taken into account. While older versions of KVM for example performed poorly with IO operations, newer versions of KVM have caught up with other hypervisors. Performance wise, both Xen and KVM seem to be the best, depending on the need of the user.

While no real choice regarding the best hypervisor can be made, it is clear however that virtualizers are more energy efficient than using just the hardware. On the other hand, virtualizing comes at a cost regarding IO throughput and overall performance. Therefore, virtualization is not useful for subjects like High Performance Computing. This may however change in the future, when hypervisors will become more sophisticated and optimized for HPC related tasks.

References

[1] Bonnie++ benchmark. http://www.coker.com.au/bonnie++/.

[2] Flexiant- common considerations when selecting your hypervisor. http://learn.flexiant.com/hs-fs/hub/154734/file-494587875-pdf/Common_considerations_when_selecting_your_hypervisor.pdf.

[3] Iozone benchmark. http://www.iozone.org/.

[4] Tpc-w benchmark. http://www.tpc.org/tpcw/default.asp/.

[5] Zotero. https://www.zotero.org/. Accessed 14-06-2015.

[6] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future generation computer systems, 28(5):755–768, 2012.

[7] Meenakshi Bist, Manoj Wariya, and Abhishek Agarwal. Comparing delta, open stack and xen cloud platforms: A survey on open source iaas. In Advance Computing Conference (IACC), 2013 IEEE 3rd International, pages 96–100. IEEE, 2013.

[8] Richard Brown et al. Report to congress on server and data center energy efficiency: Public law 109-431. Lawrence Berkeley National Laboratory, 2008.

[9] Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1):23–50, 2011.

[10] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. Evolutionary Computation, IEEE Transactions on, 6(2):182–197, 2002.

[11] Juan J Durillo and Antonio J Nebro. jmetal: A java framework for multi-objective optimization. Advances in Engineering Software, 42(10):760–771, 2011.

[12] Mateusz Guzek, Sébastien Varrette, Valentin Plugaru, Johnatan E Pecero, and Pascal Bouvry. A holistic model of the performance and the energy-efficiency

of hypervisors in an hpc environment. In Energy Efficiency in Large Scale Distributed Systems, pages 133–152. Springer, 2013.

[13] Jinho Hwang, Sai Zeng, Frederick Y Wu, and Tim Wood. A component-based performance comparison of four hypervisors. In Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on, pages 269–276. IEEE, 2013.

[14] Amol Jaikar, Gyeong-Ryoon Kim, and Seo-Young Noh. Performance trade-off between xen and kvm. Journal of Next Generation Information Technology, 4(8), 2013.

[15] Yichao Jin, Yonggang Wen, Qinghua Chen, and Zuqing Zhu. An empirical investigation of the impact of server virtualization on energy efficiency for green data center. The Computer Journal, pages 977–990, 2013.

[16] Jonathan Koomey. Growth in data center electricity use 2005 to 2010. A report by Analytical Press, completed at the request of The New York Times, page 9, 2011.

[17] Giuseppe Portaluri, Stefano Giordano, Dzmitry Kliazovich, and Bernabé Dorronsoro. A power efficient genetic algorithm for resource allocation in cloud computing data centers. In Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on, pages 58–63. IEEE, 2014.

[18] Yi Ren, Junichi Suzuki, Chonho Lee, Athanasios V Vasilakos, Shingo Omura, and Katsuya Oba. Balancing performance, resource efficiency and energy efficiency for virtual machine deployment in dvfs-enabled clouds: an evolutionary game theoretic approach. In Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion, pages 1205–1212. ACM, 2014.

[19] Yonghui Ruan, Zhongsheng Cao, and Yuanzhen Wang. Efficient live migration of virtual machines with a novel data filter. In Network and Parallel Computing, pages 369–382. Springer, 2014.

[20] Sogand Shirinbab, Lars Lundberg, and Dragos Ilie. Performance comparison of kvm, vmware and xenserver using a large telecommunication application. Published in the fifth International Conference on Cloud Computing, GRIDs, and Virtualization, 1(15):114–122, 2014.

[21] Stefan Gabriel Soriga and Mihai Barbulescu. A comparison of the performance and scalability of xen and kvm hypervisors. In Networking in Education and Research, 2013 RoEduNet International Conference 12th Edition, pages 1–6. IEEE, 2013.

[22] SK Tesfatsion, Eddie Wadbro, and Johan Tordsson. A combined frequency scaling and application elasticity approach for energy-efficient cloud computing. Sustainable Computing: Informatics and Systems, 4(4):205–214, 2014.

[23] C Valliyammai, S Uma, Bharathi K Dhivya, and P Surya. Efficient energy consumption in green cloud. In Recent Trends in Information Technology (ICRTIT), 2014 International Conference on, pages 1–4. IEEE, 2014.

[24] Sébastien Varrette, Mateusz Guzek, Valentin Plugaru, Xavier Besseron, and Pascal Bouvry. Hpc performance and energy-efficiency of xen, kvm and vmware hypervisors. In Computer Architecture and High Performance Computing (SBAC-PAD), 2013 25th International Symposium on, pages 89–96. IEEE, 2013.

[25] Jinhai Wang, Chuanhe Huang, Kai He, Xiaomao Wang, Xi Chen, and Kuangyu Qin. An energy-aware resource allocation heuristics for vm scheduling in cloud. In

High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on, pages 587–594. IEEE, 2013.

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

[27] Peng Xiao, Zhi-Gang Hu, and Yan-Ping Zhang. An energy-aware heuristic scheduling for data-intensive workflows in virtualized datacenters. Journal of computer science and technology, 28(6):948–961, 2013.

[28] Peng Xiao, Zhigang Hu, Dongbo Liu, Guofeng Yan, and Xilong Qu. Virtual machine power measuring technique with bounded error in cloud environments. Journal of Network and Computer Applications, 36(2):818–828, 2013.

[29] Jia Zhao, Yan Ding, Gaochao Xu, Liang Hu, Yushuang Dong, and Xiaodong Fu. A location selection policy of live virtual machine migration for power saving and load balancing. The Scientific World Journal, 2013, 2013.

[30] Jia Zhao, Liang Hu, Yan Ding, Gaochao Xu, and Ming Hu. A heuristic placement selection of live virtual machine migration for energy-saving in cloud computing environment. PLoS ONE, 2014.

7 Appendix

The following table indicates the 'Thomson Reuters Impact Factor' of the used journals.

Table 2: Thomson Reuters Impact Factor per Journal and citation

Journal	**Impact Factor**	**Citation number**
Future Generation Computer Systems	2.786	[6]
IEEE Transactions On Evolutionary Computation	3.654	[10]
Advances In Engineering Software	1.402	[11]
Concurrency And Computation-Practice & Experience	0.997	[12]
The Computer Journal	0.787	[15]
Journal Of Computer Science And Technology	0.672	[27]
Journal of Network and Computer Applications	2.229	[28]
PLoS One	3.234	[30]

The following table indicates the journals that were not found in the 'Thomson Reuters Impact Factor' list. This was mostly due to the literature being presented at conferences.

Table 3: Publications that have no known impact factor

Place of publication	**Possible reason**	**Citation number**
advance Computing Conference (IACC), 2013 IEEE 3rd International	Conferences and symposiums are not listed with an Impact Factor by Reuters	[7]
Lawrence Berkeley National Laboratory	Advice committee for the US government	[8]
integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium	Conferences and symposiums are not listed with an Impact Factor by Reuters	[13]
Journal of Next Generation Information Technology	Unknown	[14]
New York Times	Study performed on request of the newspaper and only published there	[16]
Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference	Conferences and symposiums are not listed with an Impact Factor by Reuters	[17]
Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion	Proceedings are not listed with an Impact Factor by Reuters	[18]
International Federation for Information Processing	Journal does not exist anymore	[19]
Cloud Computing 2014 : The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization	Conference	[20]
Networking in Education and Research, 2013 RoEduNet International Conference 12th Edition, IEEE	Conferences and symposiums are not listed with an Impact Factor by Reuters	[23]

Sustainable Computing: Informatics and Systems - Elsevier	Unknown	[22]
Computer Architecture and High Performance Computing (SBAC-PAD), 2013 25th International Symposium, IEEE	Conferences and symposiums are not listed with an Impact Factor by Reuters	[24]
High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC EUC), 2013 IEEE 10th International Conference	Conferences and symposiums are not listed with an Impact Factor by Reuters	[25]
Proceedings of the 28th Annual ACM Symposium on Applied Computing, ACM	Proceedings are not listed with an Impact Factor by Reuters	[26]
The Scientific World Journal	Reuters dropped its impact factor without specifying a reason	[29]