Using Power Measurements as a Basis for Workload Placement in Heterogeneous Multi-Cloud Environments

Mascha Kurpicz
University of Neuchâtel
Rue Emile-Argand 11
2000 Neuchâtel, Switzerland
mascha.kurpicz@unine.ch

Anita Sobe
University of Neuchâtel
Rue Emile-Argand 11
2000 Neuchâtel, Switzerland
anita.sobe@unine.ch

Pascal Felber
University of Neuchâtel
Rue Emile-Argand 11
2000 Neuchâtel, Switzerland
pascal.felber@unine.ch

ABSTRACT

Distributed data centers for multi-cloud environments usually do not consist of homogeneous hardware as they are not built at the same time by the same owner. Assigning workloads to the most appropriate processing units is therefore a challenging task. In this paper we show how in the context of heterogeneous data centers power consumption can be used as a metric to drive scheduling.

We study the performance and energy efficiency of a set of heterogeneous architectures for multiple micro-benchmarks (stressing CPU, memory and disk) and for a real-world cloud application. We observe from our results that some architectures are more energy efficient for disk-intense workloads, whereas others are better for CPU-intense workloads. This study provides the basis for workload characterization and cross-cloud scheduling under constraints of energy efficiency.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Systems

Keywords

Energy efficiency, heterogeneous architectures, cloud computing, scheduling

1. INTRODUCTION

Due to the environmental concerns of different energy resources and the massive power consumption of large data centers, energy efficiency becomes more and more important. Cloud computing as a whole consumes more energy than countries like Germany or India [2]. Besides the environmental reason, reducing the energy consumption within data centers reduces their total cost.

A first step to reduce energy consumption is making hardware more energy efficient. One of the hardware components responsible for the highest power consumption is the CPU. The Thermal Design Power (TDP) specifies the maximum amount of heat generated by the CPU that must be dissipated by the cooling system. For example, an Intel i7 Bloomfield processor with 4 cores from 2008 has a TDP of 130W[1]. In contrast, an Intel i7 Haswell-DT processor with 4 cores from 2013 has a TDP as low as 35W[2]. An approach to improve energy-efficiency is to reduce energy and power consumption by energy-aware workload-scheduling. For example, Mashayekhy et al. [5] propose a scheduler where MapReduce jobs are scheduled to different machines under constraints of energy consumption. The authors developed a greedy algorithm that improves energy efficiency while still satisfying the negotiated service level agreements (SLA). With their approach, the energy consumption can be reduced by 40 %, but they only consider a single data center and assume homogeneous hardware.

The concept of *sky computing* was introduced by Keahey et al. [4]. The authors show that with virtualization and overlays it is possible to build a multi-cloud environment in a trusted environment. However, this brings also problems of heterogeneity (not only on the hardware level) and application scheduling becomes more challenging. One example that tackles multi-cloud scheduling is shown by Tordsson et al. [7], who optimize the placement of virtual machines among different cloud providers to achieve higher performance, lower costs and better load balancing. A *cloud broker* containing a scheduler implements the decision logic based on user-specified criteria, but does not in particular consider heterogeneous hardware or energy efficiency. In the Reservoir project [6] a key role is a scheduler that assigns a particular workload to the *best* fitting cloud. The placement is based on a *load balancing policy* or a *power preservation policy.* With the power preservation policy the virtual machines are aggregated on a minimal number of physical machines and the other machines are switched off. These two policies show that there is a tradeoff between performance and power consumption. While the reduced number of machines will reduce the overall power consumption, the challenge is to not over-commit the system to avoid SLA violations during peak phases. Besides tackling challenges

Copyright © ACM, 2014. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in CrossCloud Brokers'14, December 8, 2014, Bordeaux, France, http://dx.doi.org/10.1145/2676662.2676678.

[1] http://ark.intel.com/products/37147/Intel-Core-i7-920-Processor-8M-Cache-2_66-GHz-4_80-GTs-Intel-QPI

[2] http://ark.intel.com/products/75121

towards performance and power consumption, most of the previously described research assumes that data centers consist of homogeneous hardware. However, this assumption is not realistic in real-world data centers, as shown in [3]. In this paper, we show that it is important that the scheduler is aware of the existing hardware as well as of the type of workload. As a basis, the scheduler needs information on the current power consumption as well as a notion of current performance. We show that performance per Watt is sufficient for categorizing different resources. We categorize workload types that are either CPU-intense or disk-intense. Then we run them on different setups and show that for being energy efficient, workloads have to run on the right type of hardware. This experimental study is the first step towards energy efficient scheduling decisions in a heterogeneous multi-cloud environment.

This paper is organized as follows. In Section 2 we explain the experimental setup with the hardware and software specification. We further go into details on the metrics used. Section 3 contains an extensive discussion of the gathered results and finally we conclude the paper.

2. EXPERIMENTAL SETUP

For our experiments we selected a number of hardware setups covering recent and older architectures as well as a number of representative workloads, which we describe in the following sections.

2.1 Hardware

Table 1 gives an overview of the architecture characteristics of a variety of systems that comprise either AMD CPUs or Intel CPUs. Even though they are typically not used in data centers, we also consider mobile CPUs (i7 and VIA) as they are currently among the most energy efficient architectures, typically running on battery power.

The selected systems include multicore CPUs or multiprocessors with frequencies between 1.6GHz and 3.4GHz. The Xeon (from 2007) has two processors with an overall of TDP of 300 W. The AMD (from 2008) with four processors and a total TDP of 460 W can also be power hungry. While the AMD, the AMD-TC and the Xeon have a high number of cores (8 to 16 cores), some of the Intel machines (i3, i5, i7, Haswell) have a lower core number but support hyperthreading.

The selected machines do not only differ by their CPU, but also by their hard disks. Most of the hard disks rotate 7200 times per minute (RPM), only the i5 and the mobile devices (i7 and VIA) have 5900 RPM and 5400 RPM respectively. The cache size generally depends on the size of the hard disk, i.e, larger disks comprise a larger cache.

Besides the TDP, the idle power is an important characteristic to determine energy efficiency. The idle power is the power required to run just the operating system on the hardware, hence it is typically constant and has to be considered for every workload. We therefore compare the idle power of the selected systems.

For all of our measurements, we use a highly-sensitive physical power meter (PowerSpy[3]). The driver for Linux is available as open source software[4]. In Figure 1 we see that the AMD with 387 W and the Xeon with 208 W consume already much more idle power than the other systems. The reason is that these are server architectures that do not comprise the most recent hardware parts. In comparison, the mobile devices (i7 and VIA) have a very low idle value (11W and 19W respectively) as their main target is to enhance battery life.

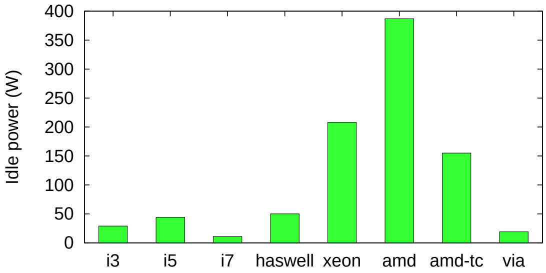

Figure 1: Idle power consumption (only OS, no workload) of the different systems.

2.2 Workloads

We selected workloads that can be categorized as CPU-intense or disk-intense to understand the behavior of the different systems under different loads.

For the CPU-intense tasks we decided to go for a variety of microbenchmarks, and one real-world application as power consumption cannot be linearly linked to CPU-utilization. For the disk-intense workload, we investigate performance and power consumption during different, but controlled operations on the hard disk.

Stress. The *stress*[5] utility is a workload generator, which we use as microbenchmark to stress CPU and memory. The CPU stress loops on the *sqrt()* operation for a given time period (30 seconds). We allocate 1 to N workers, with N being the number of threads (or cores without hyperthreading) available in the system. To stress the memory, we create a stress worker that spins on *malloc()* and *free()* operations. It allocates 1 to M GB, where M is the number of GB of RAM available minus one. By keeping 1 GB for the system, we avoid swapping of memory during the experiments, which would influence the system behavior and hence the power measurements.

Factorial. The stress workload runs for a given time period. Thus, depending on the capabilities of the different architectures it will perform a different amount of work. In contrast, each machine will perform the same number of operations when performing a complex computation. In our second benchmark we calculate the factorial of a large number (299,999).

SPECjbb2013. As a real-world benchmark, we use SPECjbb2013 [1]. This benchmark is implemented in Java and represents a typical software for a supermarket company, including distributed warehouses, online purchases and high level management operations such as data mining. During the execution, it covers different levels of CPU utilization. To ensure the benchmark runs smoothly, the JVM gets 3 GB memory per run. We exclude the VIA architecture from these experiments as it does not fulfill the minimum

[3] http://www.alciom.com/fr/produits/powerspy2.html

[4] https://github.com/patrickmarlier/powerspy.py

[5] http://manpages.ubuntu.com/manpages/trusty/man1 /stress.1.html

Hardware	Model	Cores/Threads	RAM (GB)	TDP (W)	HDD	Size	RPM	Cache
Intel i3	i3-2100 (3.1GHz)	2/4	6	65	WDC (WD2500AAKX-7)	250GB	7200	16MB
Intel i5	i5-650 (3.2GHz)	2/4	4	73	WDC (WD10EADS-22M)	1TB	5900	64MB
Intel i7	i7-2620M (2.7GHz)	2/4	4	35	WDC (WD7500BPVX-6)	750GB	5400	8MB
Intel Xeon	2x X5365 (3.0GHz)	2x 4/4	5	2x 150	Seagate (ST3250820AS)	250GB	7200	8MB
Intel Haswell	i7-4770 (3.4Ghz)	4/8	12	84	Seagate (ST2000DM001)	2TB	7200	64MB
AMD	4x Opteron 8354 (2.2GHz)	4x 4/4	8	4x 115	Hitachi (HDP72505)	500GB	7200	16MB
AMD-TC	FX-8120 (3.1GHz)	8/8	8	125	Seagate (ST1000DM005)	1TB	7200	32MB
VIA	C7-M ULV (1.6GHz)	1/1	2	8	TOSHIBA (MK1252GS)	120GB	5400	8MB

Table 1: Hardware characteristics of the selected systems.

requirements of the benchmark in terms of memory.

Bonnie++. We used *Bonnie++*[6] to stress the disk. Bonnie++ has different phases including sequential output, sequential input and random seeks. In the phase of sequential output it writes single characters and entire blocks, and it modifies blocks. For the sequential input it reads character by character and entire blocks. In the end it comprises a phase of random seeks where the benchmark measures the physical movements of the head of the hard disk.

2.3 Metrics

Every second, the PowerSpy power meter reports its measurements in Watt. If the power consumption of a workload is steady, we can take the median power consumption P and multiply it by the execution time T of the workload, which results in the energy consumption E reported in Joule:

$$E = P * T$$

We can use this metric for the stress experiment, as its resource usage is steady. Whereas the median power consumption makes sense for the constant load in the stress experiment, it cannot be used for the other experiments. In the evaluation using Bonnie++, SPECjbb and the factorial *performance per Watt* as a metric is more informative. Performance per Watt is a performance metric (usually throughput) divided by the power consumption:

$$\text{Perf}/W = \frac{\text{Throughput}}{P}$$

For the factorial experiments, the throughput is defined as the number of iterations divided by the execution time. The throughput for the disk experiments with Bonnie++ is the write or read rate per second. For the SPECjbb benchmark, we use the maximum number of jOPS, which is a throughput metric provided by the SPECjbb benchmark [1]. It represents the overall maximum throughput capacity of the system in terms of response-throughput.

The power consumption includes also the idle power as we want to observe the total cost in terms of power when running the workload on a specific hardware.

3. RESULTS

In this section we will show that two types of workloads, CPU-intense or disk-intense, have different power characteristics and can hence best take advantage of different hardware configurations.

3.1 CPU and memory power consumption

To classify the different CPUs, we show in Figure 2 the median power consumption if we apply the stress command on all cores for 30 seconds. In the case of hyperthreading, we stress up to the number of available threads. We observe similar power consumption curves for the i3, the i5, the i7 and the Haswell systems. The power consumption increases from one to two cores (or from one to four cores in the case of Haswell), and is then constant. Hence, the power consumption only increases up to the number of physical cores. The power consumption for the AMD is almost linear to the number of cores. We assume that this is caused by the rather simple architecture of the CPU with fewer hardware features (i.e., no hyper-threading, no turbo, etc.).

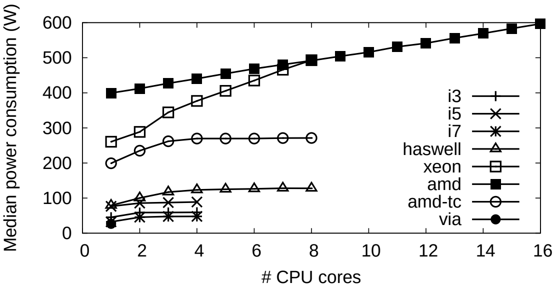

Figure 2: Median power consumption for CPU stress on all available cores.

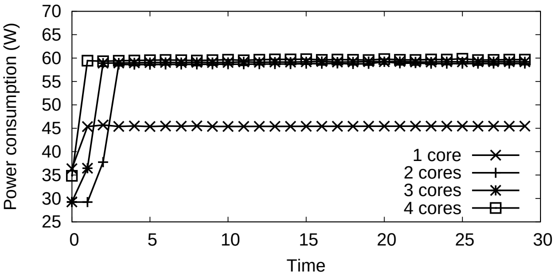

Figure 3: Power values for the CPU stress on 1-4 cores on the Intel i3 machine.

To provide more details about stressing single cores, Figure 3 depicts the power consumption for the i3. We stress one to four cores and we see that for stress on one core, the power is significantly lower than for the multiple core stress. If we stress two cores, we see that the scheduler tries to allocate two separate physical cores for load balancing. Therefore, the power consumption is doubled. However, this behavior could be influenced by pinning the stress processes to specific cores or hyperthreads, but this is out of scope of this paper. The difference between two and four cores is very small because of hyper-threading.

To compare the performance per Watt of the CPU work-

[6] http://sourceforge.net/projects/bonnie/

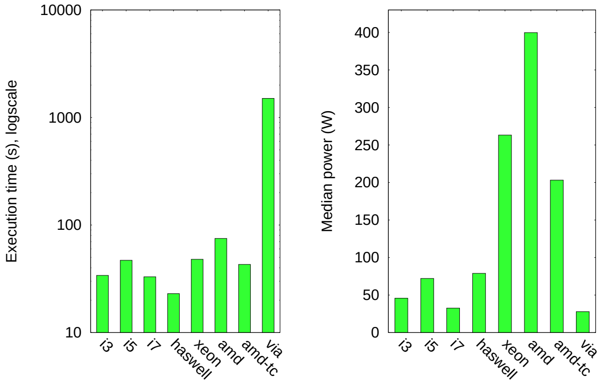

Figure 4: Execution time and median power of a factorial computation on different machines.

load, we measure the power of the computation of the factorial for a six-digit number (299,999). The number of iterations is constant, however, the execution time varies depending on the capabilities of the given system. Figure 4 depicts the execution time and the median power for the different systems. The VIA has a very long execution time for the factorial computation, but a very low median power consumption. When comparing the Haswell (4th generation i7) and the i7 (2nd generation i7), we notice that unsurprisingly the older processor is slower. When considering the median power, the older i7 consumes around 30 W whereas the Haswell machine consumes almost 80 W. The results of the Xeon and the AMD show a high impact of the high idle power in their final results. Hence, considering power consumption, it is expensive to run CPU-intense applications on these machines. Furthermore, the AMD has a lower CPU frequency (2.2 GHz) and thus a longer execution time for the factorial computation.

In order to achieve a good performance per Watt, machines need to provide a good tradeoff between execution time and power consumption. Figure 5 shows the performance per Watt for the factorial computation. Due to their long execution times, the VIA and the AMD have a very poor throughput per Watt, which is less than 20 iterations/s for a Watt. The i3, the i7 and the Haswell provide a good tradeoff between execution time and power and thus have a good performance per Watt. The i7 of 2nd generation is more energy efficient than the Haswell (4th generation i7), even though the execution time of the i7 is higher. We can thus conclude that the power consumption can be more important than the processor performance for energy efficiency.

Most workloads do not only stress the CPU, but also allocate memory. Hence, in Figure 6 we present the median power consumption for a memory-intensive workload.

We can observe that the power consumption is rather constant when a process is writing to memory. For the i5, we see that the power consumption decreases for a memory allocation of 3 GB. This machine started swapping and thus was less efficient.

To validate these results with real-world applications, we run the SPECjbb2013 benchmark [1]. Figure 7 shows the power consumption of the SPECjbb2013 workload on the i3 during a benchmark run. In this run, it is possible to identify the different phases of the workload just by plotting their power consumption. SPECjbb2013 consists of the following phases: (1) search HBIR (High Bound Injection Rate), (2) RT curve building (Response Throughput), (3) validation (run checks), (4) profiling (statistical data), and (5) reporting.

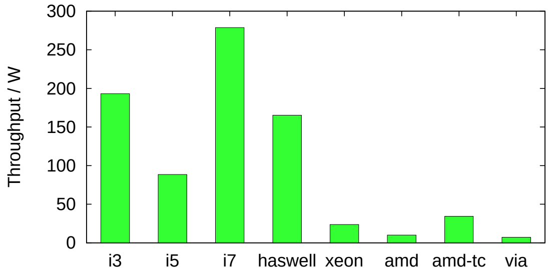

Figure 5: Throughput per Watt for the factorial computation (based on median power consumption).

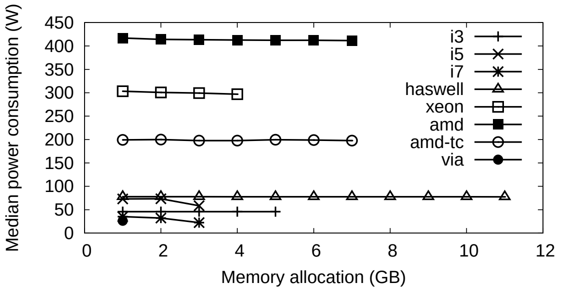

Figure 6: Median power consumption for memory stress from 1 GB to the number of GB of RAM available minus one.

The *search HBIR* phase approximates the maximum injection rate the system can handle. The most important phase for our evaluation is the *RT curve building* phase. Starting from 0% of HBIR, it increases the injection rate step by step until the maximum capacity is reached. This phase provides the data that is used to determine the maximum jOPS, which we use as throughput metric for the evaluation. The RT curve building phase can be observed in Figure 7 between second 600 and second 1300.

Based on the maximum jOPS and the maximal power values reached during the workload, we evaluate the performance per Watt. Figure 8 shows the maximum jOPS reached on the different architectures. The Haswell processor achieves the highest number of jOPS (almost 13,000). Other architectures like the i3 achieve less than 4,000 jOPS. Since the different architectures reach different jOPS, we cannot compare the execution time or the energy. With these two metrics, the machines reaching a higher throughput would be penalized.

Figure 9 depicts the maximum power value reached by each architecture during the execution of the SPECjbb2013 workload. The Xeon, even though not reaching a high number of jOPS, has a very high maximum power. Therefore, as shown in Figure 10, it has a very low performance per Watt. The Haswell machine with very high jOPS and a relatively low power consumption has the highest performance per Watt for the real-world workload.

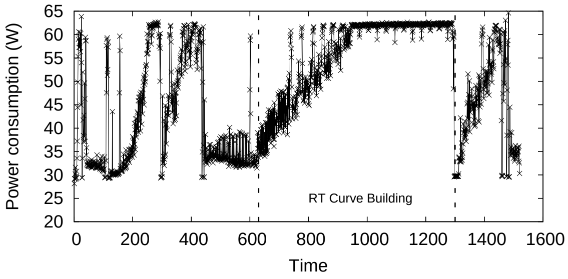

Figure 7: Power consumption of the SPECjbb workload on the i3 machine.

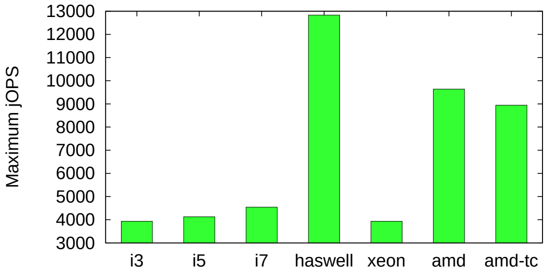

Figure 8: Maximum jOPS for the SPECjbb workload.

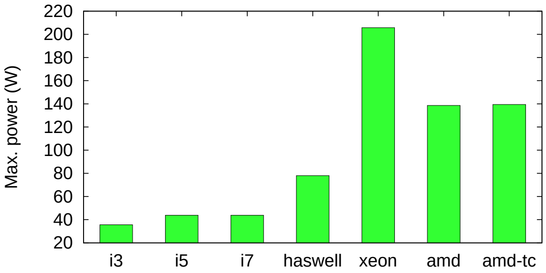

Figure 9: Maximum power value in main workload phase for SPECjbb workload.

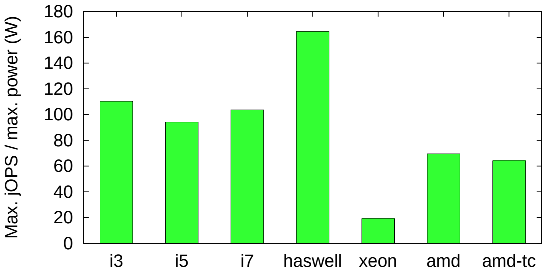

Figure 10: Performance per Watt for SPECjbb workload.

3.2 Disk power consumption

Some workloads, like for example webservers and databases, do not only utilize CPU and memory but write also to disk. We need to include such workloads in the power evaluation, too.

Figure 11 shows the power consumption for the Bonnie++ workload. This disk workload consists of the following phases: (1) sequential writing by character, (2) sequential writing by block, (3) modifying blocks, (4) sequential reading by character, (5) sequential reading by block, and (6) random seeks. The phases are separated by sleeps of 15 seconds.

When writing sequentially to the disk, we notice peaks in the beginning and in the end of the write operation. To the best of our knowledge, the initial peak is caused by the disk changing from the standby state to the active state. We verify our hypothesis by using the *hdparm* utility to put the hard disk in standby mode (see Figure 12).

The figure shows that when putting the disk in standby, we save 4 W. To change the state from standby to active, we require for a short time 8 W.

To compare the different architectures, we study the performance per Watt for block reads and writes, which we depict in Figure 13. The executions of writes and reads by character are too short and the results of the block modifications were not comparable as they are not stable enough.

We observe that the VIA netbook achieves a quite good performance per Watt, in particular for reads from the disk. This can partially be explained by the small disk size of 120 GB and a lower RPM. As seen in the previous experiments, systems with a high power have a lower performance per Watt.

3.3 Discussion

We notice that the same architecture can be more efficient for disk workloads and less efficient for CPU or memory intensive workload. In terms of throughput per Watt for the factorial computation, the i7 is the most energy efficient architecture (throughput per Watt 279), 44.5% better than the i3 with 193. For the disk workload, the i3 and the i7 have a throughput per Watt of 4388 and 2185 respectively, thus in this case the i3 is twice better than the i7. An energy efficient scheduler needs thus to make the right decisions based on the characteristics of the workloads. To give a feeling about scheduling impact, we consider the scheduler proposed by the Reservoir project [6] and its power preservation policy as explained in Section 2.2. Virtual machines are aggregated on physical hosts, and unused machines are turned off. We imagine a fictive data center with machines from our configuration: 10 times AMD, 5 times i3, 5 times Haswell. Based on how the scheduler assigns the workloads, different power scenarios are possible. We consider a very simple scenario where we place one virtual machine on each physical host and each virtual machine contains a single workload. There are ten workloads to be scheduled, half of them are writes of 30 seconds to the disk and the other half are CPU/memory-intense factorial computations of a large number. The factorial computation consumes 1,555 J on the i3, 1,810 J on Haswell and 30,018 J on AMD. Writing for 30 seconds to the disk consumes 1,064 J for the i3, 1,667 J on the Haswell and 12,008 J on the AMD.

Table 2 shows possible placements by a scheduler and the costs in terms of energy. We see that total energy drastically changes based on the scheduler decision: in the best case, the total cost is 14,370 J, whereas in the worst case it is 210,130 J, which is 14 times more. In a heterogeneous data center it is essential to choose the right hardware for a given workload. In particular, a scheduler needs to focus

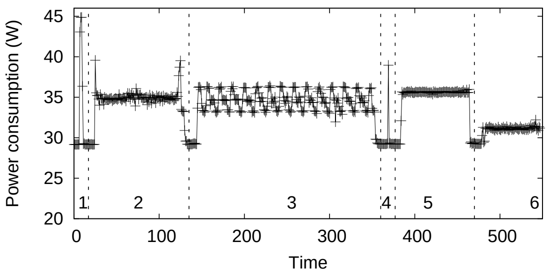

Figure 11: Bonnie++ workload for 1) sequential write by character, 2) sequential write by block, 3) modifying of blocks, 4) sequential read by character, 5) sequential read by block and 6) random seeks on the i3.

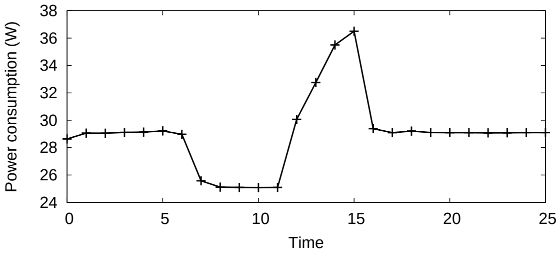

Figure 12: Disk on i3 put to standby and turning back to active state.

on aggregating workloads on energy efficient machines and possibly turn off servers with high energy consumption.

The cross-cloud broker could also include the facility to aggregate power estimation values from the different data centers and different physical machines. These values would be stored and processed in the broker and used as a basis for scheduling decisions.

4. CONCLUSIONS

In this paper we studied the energy efficiency of different workloads on heterogeneous hardware. This information is relevant for schedulers in both heterogeneous data centers and multi-cloud environments.

Our results show that the characteristics of the workload and the machine are important for cost reduction in data centers. For example, a machine that has a high throughput per Watt for disk access does not necessarily have energy efficient results for a CPU-intense workload.

In an example of a fictive data center with a very simple scheduler, we showed that we consume 14 times less energy in the best case than in the worst case. Thus, energy efficiency is clearly an important decision metric for schedulers in a heterogeneous context.

In future work, such a scheduler can take advantage of different workload characteristics and the availability of heterogeneous resources. Workloads can then be classified based on parameters such as runtime, priority and usage of resources (e.g., CPU, memory or disk). The scheduler would then choose for each category the most energy efficient hardware available for deployment. Even in a federated multi-cloud environment it would be advantageous for each cloud provider to have its data center filled with the most energy efficient workloads. When workloads are distributed in an energy efficient manner on the available hardware resources, the total cost of the data center can be reduced.

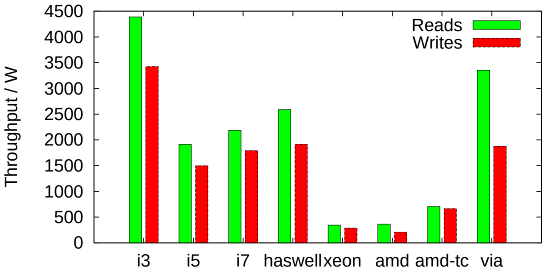

Figure 13: Write and read rate respectively divided by median power during execution of Bonnie++ during phases of block writing and block reading.

AMD	i3	Haswell	Total(J)
0	5xdisk	5xCPU	14370
0	5xCPU	5xdisk	16110
5xdisk, 5xCPU	0	0	210130

Table 2: Different workload placements and total energy costs in a fictive data center.

5. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under the ParaDIME Project (www.paradime-project.eu), grant agreement no. 318693.

6. REFERENCES

[1] Specjbb2013 design document. *Standard Performance Evaluation Corporation (SPEC)*, 2013.

[2] G. Cook. How clean is your cloud? Report, Greenpeace International, April 2012.

[3] T. Heath, B. Diniz, E. V. Carrera, W. Meira Jr, and R. Bianchini. Energy conservation in heterogeneous server clusters. In *Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming*, pages 186–195. ACM, 2005.

[4] K. Keahey, M. Tsugawa, A. Matsunaga, and J. A. Fortes. Sky computing. *Internet Computing, IEEE*, 13(5):43–51, 2009.

[5] L. Mashayekhy, M. M. Nejad, D. Grosu, D. Lu, and W. Shi. Energy-aware scheduling of mapreduce jobs. In *Proc. of the 3rd IEEE International Congress on Big Data*, pages 32–39, 2014.

[6] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, et al. Reservoir - when one cloud is not enough. *IEEE computer*, 44(3):44–51, 2011.

[7] J. Tordsson, R. S. Montero, R. Moreno-Vozmediano, and I. M. Llorente. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. *Future Generation Computer Systems*, 28(2):358–367, 2012.