Dynamic Voltage and Frequency Scaling in Multimedia servers

Alaa Brihi and Waltenegus Dargie
Chair for Computer Networks
Faculty of Computer Science
Technical University of Dresden
01062 Dresden, Germany
Email:brihi.alaa@gmail.com, waltenegus.dargieg@tu-dresden.de

***Abstract*—In this paper, we experimentally investigate the scope and usefulness of Dynamic Voltage and Frequency Scaling (DVFS) in multimedia servers. For our experiment, we considered four scaling policies, two heterogeneous servers, and two different application scenarios. In the first scenario, we used an IO-Intensive multimedia downloading application while in the second scenario we used a predominately CPU-Intensive application, which transcodes video files. We will show that while the advantage of DVFS for IO-Intensive applications is apparent, it is not so with CPU-Intensive applications. The advantage in IO-Intensive applications depends on the selection of frequencies in the machine and the way the CPU speed scales. We observed that the choice of a particular DFVS technique became more consequential when the machine had a wide selection of operation frequencies while a gradual change in the operation frequency was more energy efficient. For CPU-Intensive applications, the use of DVFS was counter productive and the overhead of scaling was considerable.**

***Index Terms*—Dynamic power management, dynamic voltage and frequency scaling, power consumption analysis, multimedia servers, energy proportion computing**

I. Introduction

The power consumption and the energy-cost of Internet-based servers and data centres have been steadily increasing at a considerable pace. This will have a noticeable and long term impact on the economy, the environment, and the way Internet-based service will be provided in the future [5], [1], [11], [9].

The research community is trying to achieve energy-efficient computing at a considerable scale. Broadly speaking, the existing or proposed approaches can be categorised into two. The first category of approaches focuses on designing hardware and software systems that consume as little energy as possible. As far as hardware design is concerned, promising progress are emerging in processor and memory design [8]. The second category of approaches strive to embed self-managing capability in server clusters so that these can become "energy-proportional" [1]. This means that the energy consumption of a particular cluster is proportional to the work it accomplishes. Ideally, in this setting, a server consumes almost no power when it is idle. Some of the mechanisms that enable energy-proportional computing are "dynamic resource pool sizing" [10], [12], meaning the amount of computing resources required to handle a given workload is decided dynamically; "service consolidation" [14], [2], [7], [16] in which services running on underutilised servers are migrated at runtime to those which can accommodate them, so that the underutilised servers can be switched off; and dynamic voltage and frequency scaling (DVFS), in which the voltage and frequency of the processor(s) of a server (cluster) are dynamically adjusted according to the anticipated workload of the server (cluster) [6], [4].

The premises for DVFS are the following: (1) the power consumption of the processor in a server amounts to the largest portion of the overall power consumption of the server; (2) task schedulers in operating systems often over provision resources to tasks; and (3) most of the tasks are completed before their deadline leading to a significant amount of idle time between consecutive tasks. As a result, a large number of DVFS algorithms have been proposed to frugally allocate computing resources and minimise the idle power consumption of the processor by setting its operation frequency and voltage at the minimum level possible. The main focus of this paper is to experimentally examine the scope and usefulness of DVFS in multimedia servers. We will use heterogeneous server platforms and different applications with varying task arrival rates in a realistic server cluster.

The rest of this paper is organized as follows. In Section II, we give a brief introduction about dynamic voltage and frequency scaling. In Section III we outline our methodology to experimentally investigate the scope and usefulness of dynamic frequency scaling in multimedia servers. In Section IV, we present our experiment results and share our observations. Finally, in Section V, we give concluding remarks.

II. Dynamic Voltage and Frequency Scaling

The basic motivation of dynamic voltage and frequency scaling is the relationship between the power consumed by the processor and its operation voltage and frequency, which can be expressed as follows:

$$P = f\left(f, V^2, C\right) \tag{1}$$

where f is the operation (switching) frequency of the processor, V is processor core voltage and C is related to the transistors' capacitance inside the CMOS technology. As can be seen in Equation 1, reducing the switching frequency of the processor reduces its operation speed, but it reduces also its power consumption. Likewise, reducing the core voltage by one fold reduces the power consumption by two fold. This does not mean, however, that the operation frequency and voltage of the processors can be varied arbitrarily. Most existing processors have limited operation frequencies and there is a strong dependency between these frequencies and permissible processor voltages.

The scheduler inside the operating system is responsible for binding tasks (processes) to CPUs and for setting a deadline to each binding. In some cases, the processor runs idle between

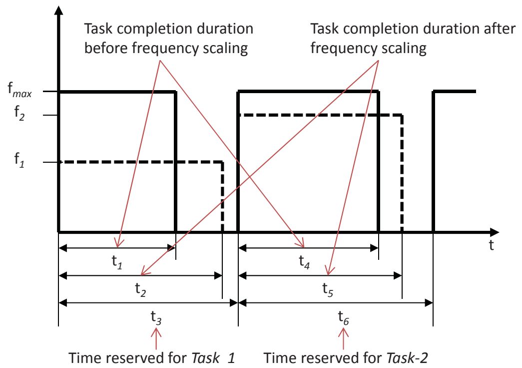

Fig. 1. Dynamic voltage and frequency scaling

the time the last task is executed and until the next batch of tasks arrive in the queue. In addition, not all tasks require the maximum capacity of the processor. In both cases, it is useful to reduce the voltage and frequency of the processor to reduce its idle-state power consumption. For example, in Figure 1 the scheduler sets deadlines for the two tasks, but both of them are completed before their deadlines. The dashed lines illustrate the extended execution durations when DVFS is applied. Ideally, these extension durations are still within the set deadlines, but in reality it is difficult to reduce the power consumption of the processor and avoid the violation of task deadlines at the same time, in which case, DVFS entails performance degradation.

DVFS is carried out in three steps: First, the workload of the processor (or, alternatively, the average duration of the idle time between consecutive tasks) is predicted. Secondly, the CPU cycles to handle this workload is estimated and the appropriate voltage and frequency are determined. Third, the operation frequency and voltage are adjusted. When the intertask arrival time is long, the overhead of prediction and estimation is justified, but when it is short, it is not. We shall experimentally examine the gain and the overhead cost of DVFS in the subsequent sections.

III. METHODOLOGY

In this section, we briefly explain our experiment setting and our methodology to measure and analyse the impact of dynamic voltage and frequency scaling on the power consumption characteristic of multimedia servers. In an earlier publication [3], we have described the experiment setting in more detail and report our initial result, though the report focused entirely on homogeneous servers, which is not the case here.

We set up a cluster consisting of two heterogeneous servers, a switch, and a client (workload generator). One of the servers is built on a D2581 Siemens-Fujitsu motherboard and integrates a 3.16 GHz Intel E8500 dual core processor and 4 GB PC-5300 DDR2 SDRAM memory chips. The second server is built on a D2461 Siemens-Fujitsu motherboard and integrates a 2 GHz AMD Athlon 64 dual core processor and 4 GB PC-4200 DDR2 SDRAM memory chips.

The AMD Athlon 64 X2 server can operate at four different frequencies: 1000 MHz, 1330 MHz, 1670 MHz, and 2000 MHz. Accordingly, the processor core voltage (VDD) can be adjusted between 0.8 V and 1.55 V in step of 0.25 V. Likewise, the Intel processor's core voltage can be adjusted between 0.85 and 1.3625 V and the operation frequency can be varied between 1999 MHz and 3165 MHz.

Predicting the future workload (utilisation) of the processor is essential to estimate the appropriate operation frequency. IN the AMD server, the operating system can sample the CPU utilisation at rate in the interval $[0.23 \times 10^3, 93.45]$ samples/s. For the Intel processor, it is in the interval $[0.23 \times 10^3, 100]$ samples/s. A high sampling rate is accurate but its overhead is high as well and interferes with the normal operation of the server. On the other hand, a low sampling rate is inaccurate but its overhead is negligible. In all our experiment, we used 10 samples/s for the AMD processor and 50 samples/s for the Intel server. Table I summarises the specification of the AMD and Intel processors.

Server	***AMD***	***Intel***
processor	Athlon 64 X2 3800+	Core2Duo E8500
Clock speed (GHz)	2.0	3.16
Cores / Threads	2/2	2/2
Frequency (GHz)	1.0 - 2.0	1.9 - 3.16
Voltage (V)	0.8 - 1.55	0.85 - 1.36
L2 cache	512KB	6 MB
Memory	4 GB DDR2 SDRAM 133 MHz	4 GB DDR2 SDRAM 667 MHz
Storage(GB)	160	160

TABLE I
THE SPECIFICATIONS OF THE PROCESSORS WE EMPLOYED IN THE DVFS EXPERIMENTS.

We run Ubuntu Server Edition (v 10.04) on all our machines[1] and install Apache[2] on both servers to process multimedia download requests from users. Both servers host a large database of video and music files of various sizes (between 3 MB and 100 MB). The servers process user requests in two different scenarios. In the first scenario, users request for videos of available formats. The servers search these videos and download them without further processing. This scenario is predominantly IO-intensive. In the second scenario, users request for videos of unavailable formats, in which case, the servers use a transcoder to generate the desired formats (MPG4 and FLV formats were converted to AVI format) and send the videos to the users. The second scenario is predominantly CPU-intensive. We use FFmpeg[3] for transcoding the video files.

In both scenarios, the servers accommodate up to 100 requests/s.

A. DC Power Measurement

In order to justify the results presented in this paper, we will explain next how we took measurements. The motherboard of each server is supplied with DC power through a 4-pole connector and a 24-pole power connector. The 4-pole connector supplies the motherboard with 12 V (we will refer to this voltage as 12V2) while the 24-pole power connector supplies the motherboard with 3.3 V, 5 V, and 12 V (we will refer to this voltage as 12V1).

[1]http://www.ubuntu.com
[2]http://www.apache.org/.
[3]http://ffmpeg.org/ (Last visited on February 21, 2012: 15:25 CET.

The main voltage regulator of the D2581 motherboard is a two-phase voltage regulator controlled by an ISL 6326 Pulse Width Modulator (PWM) controller. The output of each phase is supplied to the processor 50% of the time. The controller obtains its core voltage (VCC) from the 5 V rail of the 24-pole connector, but the two MOSFET drivers and their N-channel power transistors are powered by the 12V2 rail. In other words, the processor draws a significant portion of its power from the 12V2 rail.

The voltage regulator responsible for generating the various voltages of the memory unit (including the memory termination logic) is the TPS 51116 voltage regulator. This voltage regulator draws power exclusively from the 5 V rail of the 24-pole connector.

The Southbridge of the motherboard connects all the IO-controllers with the processor subsystem. It is supplied with power by a single-phase voltage regulator employing the ISL 6521 PWM controller. The controller draws power from the 12V1 and the 5 V rails.

Likewise, the main voltage regulator of the D2461 motherboard is a three-phase voltage regulator, employing an ISL 6312 PWM controller. Each phase uses three H9N03LA power transistors as its switching elements. The PWM controller obtains its core voltage from the 5V rail of the 24-pole connector while all the power transistors are connected to the 12V2 rail.

A single phase voltage regulator controlled by an ISL 6545 PWM controller generates all the voltages of the memory subsystem. The core voltage of the PWM controller and the biasing voltage of the power transistors are taken from the 5V rail of the 24-pole connector.

The motherboard provides the Southbridge with two voltage regulators. One of them is a single phase voltage regulator controlled by an ISL 6545 PWM controller while the other is an adjustable SPX1587 low power voltage regulator that can generate 1.5, 1.8, 2.5, 3.3, or 5 V. The PWM controller of the single phase voltage regulator gets its supply voltage (VDD) from the 12V1 rail. Likewise, the drain of one of the power transistors is connected to the 12V1 rail. The core voltage of the SPX1587 voltage regulator is taken from the 5V rail of the 24-pole connector.

IN summary, in both servers, the 12V2 rail supplies power to the processor and the predominant subsystem that draws current through the 5 V rails is the memory unit.

We employ Yokogawa WT210 digital power analysers to measure the energy and power consumptions of the servers. The devices can measure DC as well as AC power consumption at a rate of 10 Hz and a DC current between 15 μA and 26 A with an accuracy of 0.1%.

B. DVFS Tool

We integrated the **cpufrequtils** utilities[4] into the Ubuntu kernel infrastructure for supporting DVFS. The utilities provide us with three different types of scaling policies: power-save, on-demand, and conservative [13]. The power-consumption and performance of the servers under these policies will be compared to each other and to the condition in which no scaling policy is used, i.e., when the servers are running at maximum frequency. The latter state is called *performance state*. The power-save policy operates the processors at the lowest frequency while the on-demand and conservative policies adapt the clock frequency to the change in the workload of the servers. The last two policies estimate the utilisation of the processor using a moving average technique, predict its future workload (for the next time slot), and scale-down or scale-up the processor's speed accordingly. The essential difference between the two is that the on-demand policy scales up the CPU frequency to the maximum whenever an increment in the CPU utilisation is predicted whereas this is done gradually in the conservative policy. This essential difference is depicted in Figure 2.

[4]https://wiki.archlinux.org/index.php/Official_Repositories: Last accessed on November 14, 2011: 22:38 CET.

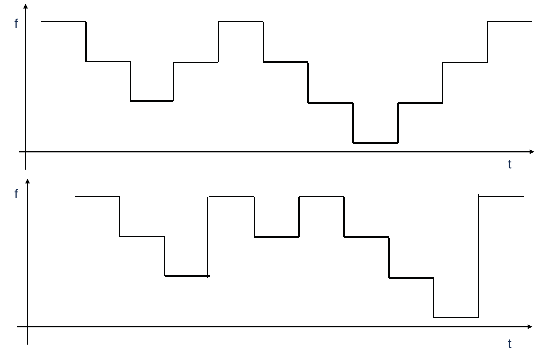

Fig. 2. The difference between the conservative (top) and on-demand (bottom) scaling policies.

All measurements were accomplished by running the experiments for a similar duration (one hour). Each experiment was conducted at least 5 times to collect enough statistics about the applied policies.

IV. Experimental Results

We use the Cumulative Distribution Function (CDF) to examine the power consumption of the two servers under the different scaling policies. The CDF approach considers the power consumption as a random variable **p**. This is a fitting consideration since it is not possible to give a complete account as to why the power consumption of a server or a component thereof behaves the way it does. The CDF, or simply the distribution function is defined as:

$$F_p(p) = P\{\mathbf{p} \leq p\} \tag{2}$$

where p is a real number. In other words, the distribution function expresses the probability that the random variable **p** has a value less than or equal to a certain real number p. Because it is a cumulative function, $F_p(p)$ is monotonic increasing, so that for any $\{p_1, p_2 | p_2 > p_1, F_p(p_2) \geq F_p(p_1) \quad \forall p_2, p_1\}$. Moreover, for our case, $F_p(0) = 0$ and $F_p(\infty) = 1$.

A. Overall (AC) Power Consumption

We measured the overall (AC) power consumption of the two servers before the power entered into the main power supply unit. The AC power consumption includes the actual power consumed by the server as a result of a work done by the server as well as the power dissipation at the power supply unit during the AC to DC conversion and at the voltage regulator during a DC to DC conversion. The DC to DC conversion is

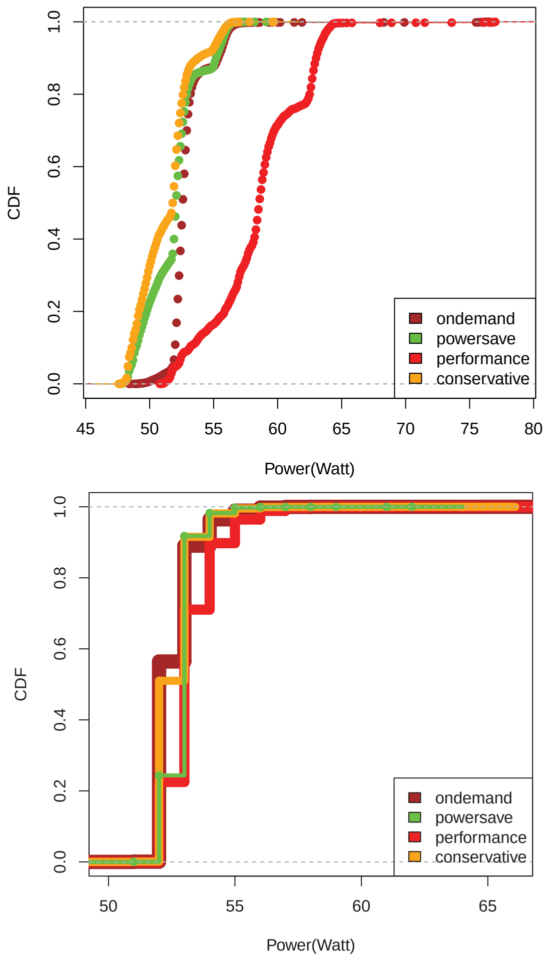

Fig. 3. The distribution functions of the overall power consumptions of the AMD (top) and Intel (bottom) servers when they run the IO-Intensive application.

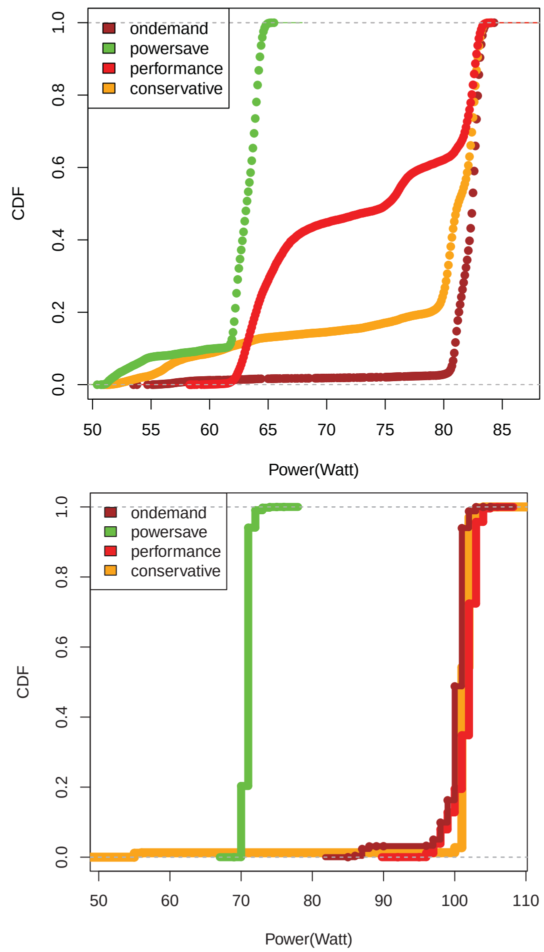

Fig. 4. The distribution functions of the overall power consumptions of the AMD (top) and Intel (bottom) servers when they run both applications.

required because most of the DC voltages generated by the power supply unit are not suitable to the various hardware components, including the processor and the memory unit.

Figures 3 displays the CDF of the overall power consumption of the AMD and Intel servers when Apache was the only application running. In each case, the user request rate was 100 requests/s to download a 3.4 MB music file. For the AMD server the graphs clearly display the influence of the DVFS policies. All of them produced almost comparable results with $F_p(57W) = 1.0$. In contrast, the performance policy resulted in the highest power consumption with $F_p(67W) = 1.0$. For the Intel server, the influence was not remarkable. True, a careful examination of the CDFs reveals that the performance policy resulted in the highest power consumption, for example, $F_p(52) = 0.9$ for all the DVFS policies whereas it is 0.7 for the performance policy. This difference is not considerable, nevertheless.

Figure 4 displays the CDF of **p** when the two servers run both Apache and the transcoder (the second scenario). To start with, none of the DVFS policies (except the power save mode) resulted in a better power saving than the performance policy. The power save policy had a visible gain in both servers, but as we will see shortly, this gain came at a price (a loss in throughput). Secondly, the performance policy is the one with the best performance (see Table III and V). The reason is straightforward: the transcoder stressed the processor much of the time and it made no sense to scale its operation frequency. However, the background computation to predict the future workload and to determine the appropriate frequency resulted in an extra power consumption and a reduction in throughput.

The CDFs of **p** for the AMD server were more diverse than those for the Intel. This is because the number of alternative frequencies in the AMD processor were larger than in the Intel server.

B. DC power measurement

To better understand the effect of DVFS on the DC power consumption of a server, it is useful to examine how much of the AC Power reaches at the processor.

Depending on the load on the power supply unit, between 25 and 35% of the AC power is dissipated during the AC to DC conversion[5]. We believe there is an additional 8 to 10% power loss on the DC power due to a further DC to DC conversion at the various voltage regulators. We were not able to measure this loss, as this meant essentially modifying the

[5]ATX Specification, version 2.2 (2003 – 2005), Intel Corporation.

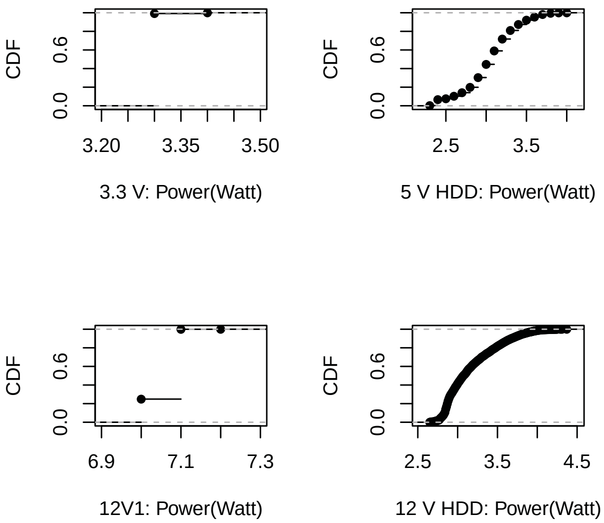

Fig. 5. The "static" component of the DC power consumption of the AMD server.

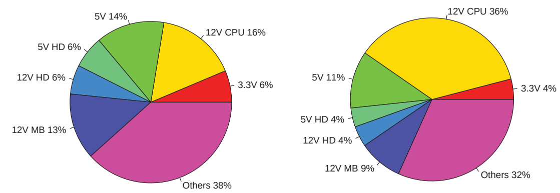

Fig. 6. The portion of the DC power consumed by the various subsystems in the AMD server under the on-demand policy when the IO-intensive application (left) and when both applications were running.

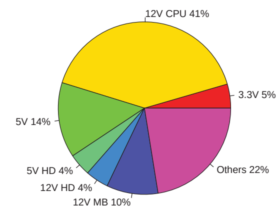

Fig. 7. The portion of the DC power consumed by the various subsystems in the AMD server under the performance policy when both applications were running.

motherboard structure, which was a difficult task. Therefore, our measurement was limited to the circuits between the output of the power supply unit and the voltage regulators.

The useful DC power has "static" and dynamic components. The DC power drawn through the 3.3 V and the 12V1 do not change appreciably under the various experiment scenarios or configurations. Likewise, the power consumption of the disk drive can be considered as a constant cost[6]. This is consistent with our analysis of the power distribution in the D2461 and D2581 motherboards (Section III-A). The 3.3 V line supplies power to the peripherals – including the Network Interface Card (NIC). The power consumption of the NIC was around 1.6 W in the AMD server and 2 W in the Intel server and varied feebly throughout the experiment. The remaining power consumption is on account of the graphic card, which is also appreciably small. The 12V1 is used, by and large, as a control signal by some of the voltage regulators and the memory termination logic. The CPU fan is supplied with power through the 12V1 line and it is both small and invariable. Figure 5 displays the CDF of the "static" power consumption of the AMD server for the performance policy.

Figure 6 displays the proportion of the overall power consumption of the various subsystems of the AMD server under the on-demand policy and when it run the IO-Intensive application (left in Figure 6) and when it run the CPU-Intensive application (right in Figure 6). Figure 7 displays the proportion of power consumption in the same server under the performance policy. In the on-demand policy, the largest portion of the AC power is lost in the form of dissipation when Apache was running alone and the portion of dissipation was comparable to the power consumed by the processor when both applications were running. For example, when Apache was the only application running, the average AC power consumption of the server was about 53 W. The average combined DC power consumption of the different subsystems was 33 W. Hence, 20 W (about 38%) was lost in form of dissepation due to the inefficiency of the power supply unit, which is consistent with the ATX specification. At this load, the ATX specification requires that the power supply unit should have an efficiency of 65%. The portion of power consumed by the processor became visibly dominant when the AMD server was running under the performance policy. It must be noted that this portion is a useful power because now the processor was doing some useful work. The power efficiency improved to 78% in this setting and this fact should be taken into account when speaking about the gain of DVFS.

The power drawn through the 5 V and 12V2 lines were dynamic, because it changed according to the workload of the server. Of these, the change in the 12V2 is dominant in both servers. Therefore, we will not concern ourselves with the power drawn through the 5V rail.

C. Processor's DC Power Consumption

Figure 8 displays the CDF of the DC power consumption of the AMD and Intel processors when the IO-Intensive application (Apache) was running. In the AMD processor, the advantage of DVFS for the IO-Intensive application (when Apache was the only application running) was conspicuous. Comparatively, the conservative policy (the gradual incremement of the processor frequency) performed better than all the others. It is also this policy which yielded the highest throughput. Remarkably, the performance policy consumed the highest average power but produced the lowest throughput (see Table II), clearly contradicting the thesis that there is a strong correlation between high power consumption and high performance. For the IO-Intensive application, the Intel processor performed well under all policies when the IO-intensive application was running alone, though high throughput was obtained under the performance policy (Table IV).

For the CPU-Intensive application (Figure 9), only the power save mode was distinct from the other. All the other policies were competitive. The gain of the power save policy was at a significant reduction in throughput, as can be seen in Table III and V.

[6]This is true as long as the request rate is below 100/s and the average file size is 3 MB. For larger request rates and larger video data, the power consumption of the disk drive fluctuates between 7 and 14 W.

Policy	Power (Watts)	CPU (%)	Throughput (GB)
Ondemand	53	5.79	51.80392
Power-save	51.9	5.27	50.17798
performance	58.5	4.44	49.87089
conservative	51.3	5.6	55.24799

TABLE II
A COMPARISON OF THE RELATIONSHIP BETWEEN THE POWER CONSUMPTION, CPU UTILIZATION, AND THROUGHPUT OF THE IO-INTENSIVE APPLICATION IN THE AMD SERVER.

Policy	Power (Watts)	CPU (%)	Throughput (GB)
Ondemand	81.8	99.97356	0.9463543
Power-save	62.4	99.97458	0.5701639
performance	73.3	99.52904	1.362283
conservative	78.1	99.65371	0.6737399

TABLE III
A COMPARISON OF THE RELATIONSHIP BETWEEN THE POWER CONSUMPTION, CPU UTILIZATION, AND THROUGHPUT OF BOTH APPLICATIONS IN THE AMD SERVER.

Policy	Power (Watts)	CPU (%)	Throughput (GB)
Ondemand	52.6	1.204606	50.45462
Power-save	52.86	1.572874	51.33617
performance	53.22	2.034471	52.74526
conservative	52.59	1.472948	47.26372

TABLE IV
A COMPARISON OF THE RELATIONSHIP BETWEEN THE POWER CONSUMPTION, CPU UTILIZATION, AND THROUGHPUT OF THE IO-INTENSIVE APPLICATION IN THE INTEL SERVER.

Policy	Power (Watts)	CPU (%)	Throughput (GB)
Ondemand	99.96	98.23628	1.86271
Power-save	70.86	99.97383	0.4185703
performance	101.52	98.17405	1.328156
conservative	100.89	98.71851	1.345913

TABLE V
A COMPARISON OF THE RELATIONSHIP BETWEEN THE POWER CONSUMPTION, CPU UTILIZATION, AND THROUGHPUT OF BOTH APPLICATIONS IN THE INTEL SERVER.

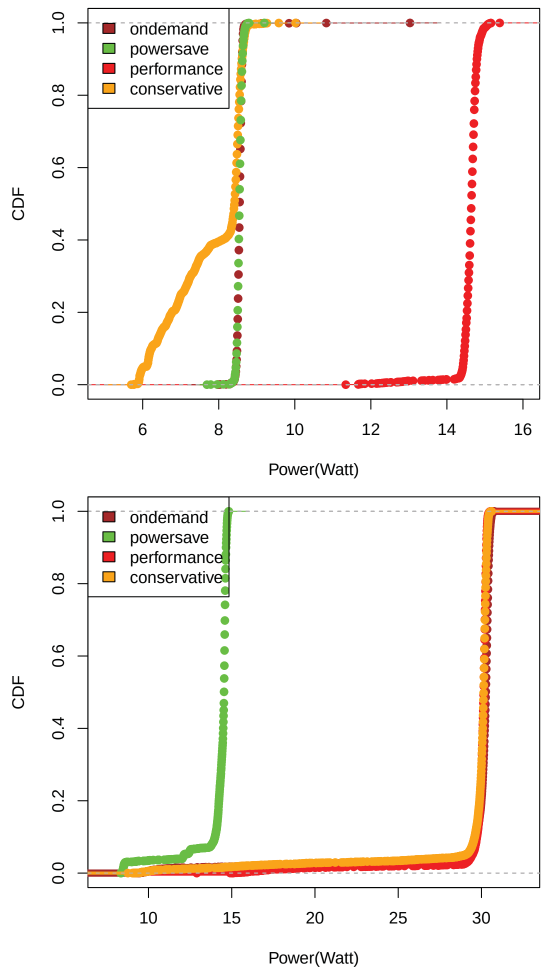

Fig. 8. The distribution functions of the power consumptions of the AMD (top) and Intel (down) processors under the different DVFS policies when the servers run the IO-Intensive application.

D. Energy Efficiency

In order to make comparison between the AMD and Intel platforms, we use the Energy-Efficiency expression defined in [15]: It is the ratio of the overall (the one hour) energy consumed to the work done by the server. For the first scenario, the work done can be expressed in terms of the performance of the server, which is the amount of data (GB) downloaded in one hours. For the second scenario, the performance is the amount of transcoded and transferred data (in GB) in one hour.

$$EE = \frac{Work}{Energy} = \frac{Work}{(Power \times Time)} = \frac{Performance}{Power} \tag{3}$$

The conservative policy in the AMD server for the IO-intensive workload resulted in the highest energy efficiency (EE). This is expected, since the conservative policy gradually adapts the clock frequency, avoiding a temporal unstable condition during DVFS. For the CPU-intensive workload, the power-save policy in the Intel server resulted in the highest EE.

The highest standard deviation of EE was observed in the AMD server for the IO-intensive workload, indicating that the choice of a scaling policy has a notable consequence on the energy efficiency of the server. Compared to the the standard deviations of EE in the IO-Intensive workload, the standard deviations of EE in the CPU-Intensive workload for both servers are very small, confirming our assertion that DVFS has little or no consequence when the CPU utilisation remained high and invariable.

Policy	IO-Intensive		CPU- and IO-Intensive	
	AMD	Intel	AMD	Intel
On-demand	0.97	0.95	0.011	0.018
Power-save	0.96	0.97	0.009	0.05
performance	0.85	0.99	0.018	0.013
conservative	1.05	0.89	0.008	0.013

TABLE VI
A COMPARISON OF THE ENERGY EFFICIENCY RATIO (EE IN GB/WATT) BETWEEN THE AMD AND INTEL SERVERS FOR THE IO-AND CPU INTENSIVE SCENARIOS.

The EE summary for the IO- and CPU-Intensive scenarios is given in Table VI.

V. CONCLUSION

In this paper we investigated the relationship between the power consumption and performance (throughput) of two heterogeneous servers when they operated under four different dynamic power management policies. The scaling strategies performed well for IO-Intensive applications. These are applications which leave the CPU idle much of the time. Moreover, the DVFS strategies resulted in different power consumption

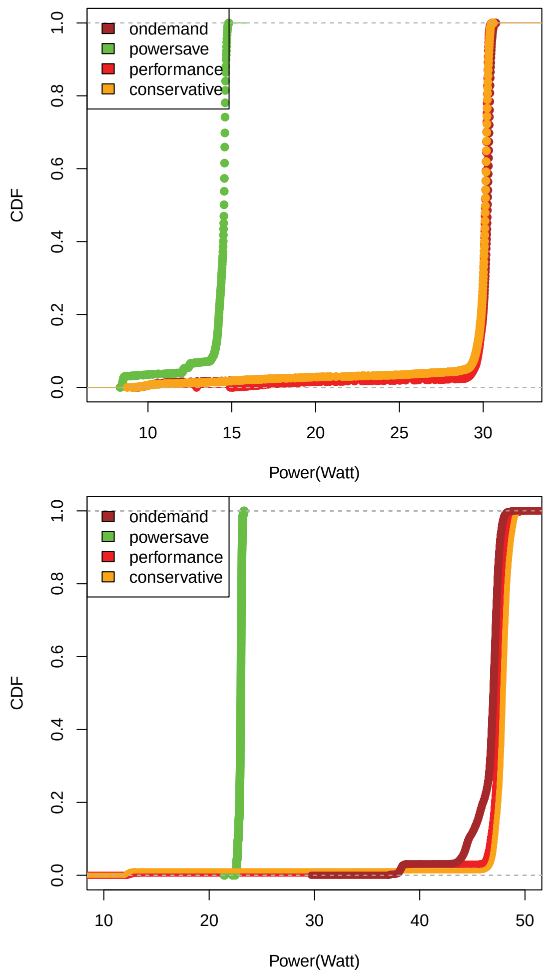

Fig. 9. The distribution functions of the the AMD (top) and Intel (down) processors under the different DVFS policies when the servers run both applications.

profiles, suggesting that the choice of a DVFS strategy for IO-Intensive application can be of a considerable consequence. The diversity in the power consumption characteristic in the AMD server was more visible than in the Intel server for the IO-Intensive application.

For a CPU-Intensive application, none of the DVFS strategies were advantageous. In fact, more power was consumed in both servers under these strategies without any gain in the throughput. Apparently, the background process to predict the workload of the CPU and determine the appropriate frequency as well as switch between frequencies resulted in a considerable overhead. This overhead was more significant in the AMD server, which offers a large alternative frequencies, than in the Intel server, which has very few alternative frequencies for scaling.

In general, from the experiments we learned that it was difficult to establish a meaningful relation between power consumption, CPU utilization, and performance when the server runs the CPU-Intensive application. We employed the energy-efficiency (EE) ratio to compare the energy consumption of the two heterogeneous servers. We discovered that the AMD server under the conservative policy performed the best when it run the IO-Intensive application. The two reasons for this are (1) the good frequency selection in the AMD server and (2) the gradual frequency increment in the conservative policy.

Acknowledgment

This work has been partially funded by the German Research Foundation (DFG) under project agreement SFB 912/1 2011.

References

[1] L. A. Barroso and U. Hölzle. The case for energy-proportional computing. *Computer*, 40:33–37, December 2007.
[2] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, pages 337–350, Berkeley, CA, USA, 2008. USENIX Association.
[3] W. Dargie. Analysis of the power consumption of a multimedia server under different dvfs policies. In *IEEE CLOUD*, pages 779–785, 2012.
[4] W. Dargie. Dynamic power management in wirless sensor network: State-of-the-art. *IEEE Sensor Journal*, 12(5):1518–1528, 2012.
[5] W. Dargie, A. Strunk, and A. Schill. Energy-aware service execution. In *The 36th Annual IEEE Conference on Local Computer Networks*, 2011.
[6] G. Dhiman and T. S. Rosing. Dynamic voltage frequency scaling for multi-tasking systems using online learning. In *Proceedings of the 2007 international symposium on Low power electronics and design*, ISLPED '07, pages 207–212, New York, NY, USA, 2007. ACM.
[7] E. N. Elnozahy, M. Kistler, and R. Rajamony. Energy-efficient server clusters. In *Proceedings of the 2nd international conference on Power-aware computer systems*, PACS'02, pages 179–197, Berlin, Heidelberg, 2003. Springer-Verlag.
[8] H. Esmaeilzadeh, T. Cao, Y. Xi, S. M. Blackburn, and K. S. McKinley. Looking back on the language and hardware revolutions: measured power, performance, and scaling. *SIGARCH Comput. Archit. News*, 39(1):319–332, Mar. 2011.
[9] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In *ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture*, pages 13–23, New York, NY, USA, 2007. ACM.
[10] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper. Resource pool management: Reactive versus proactive or let's be friends. *Comput. Netw.*, 53:2905–2922, December 2009.
[11] J. Hamilton. Internet-scale service infrastructure efficiency. *SIGARCH Comput. Archit. News*, 37:232–232, June 2009.
[12] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *INFOCOM*, pages 1098–1106, 2011.
[13] V. Pallipadi and A. Starikovisky. The ondemand governer. In *Proceedings of the Linux Symposium (volume two)*, 2006.
[14] S. Srikantaiah, A. Kansal, and F. Zhao. Energy aware consolidation for cloud computing. In *Proceedings of the 2008 conference on Power aware computing and systems*, HotPower'08, pages 10–10, Berkeley, CA, USA, 2008. USENIX Association.
[15] D. Tsirogiannis, S. Harizopoulos, and M. A. Shah. Analyzing the energy efficiency of a database server. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 231–242, New York, NY, USA, 2010. ACM.
[16] Q. Zhu, J. Zhu, and G. Agrawal. Power-aware consolidation of scientific workflows in virtualized environments. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '10, pages 1–12, Washington, DC, USA, 2010. IEEE Computer Society.