Modeling power consumption in multicore CPUs with multithreading and frequency scaling

D. Cerotti, M. Gribaudo, P. Piazzolla, R. Pinciroli, and G. Serazzi

Abstract The rapid growth of energy requirements in large data-center has motivated several research projects focusing on the reduction of power consumption. Several techniques have been studied to tackle this problem, and most of them require simple power models to estimate the energy consumption starting from known system parameters. It has been proven that the CPU is the component of a server that is most responsible for its total power consumption: for this reason several power models focusing on this resource has been developed. However, only a few accounts for standard CPU features like dynamic frequency scaling and hyperthreading, which can have a significant impact on the estimation accuracy. In this paper, we present the results from a set of experiments focusing on these CPU features, and we propose a simple power model able to provide accurate power estimates by taking them into account.

1 Introduction

Power consumption reduction in datacenter is one of the most important research topics that is being addressed in different ways and with different techniques by many scientists both from the industry and the academia. However, most of the proposals require a suitable power model that can estimate the energy consumption starting from simple system parameters, like the utilization of its resources, and the time required to complete the considered tasks. Especially when the considered techniques are based on results that exhibit non-linear behavior, the accuracy of the power consumption model is of paramount importance to correctly identify the optimal system configurations that can achieve the target performance with the lowest possible energy budget. In different works, such as [9] and [17], the instantaneous

Dip. di Elettronica, Informazione e Bioingegneria, Politecnico di Milano,
via Ponzio 34/5, 20133 Milano, Italy,
e-mail: [davide.cerotti,marco.gribaudo,pietro.piazzolla,riccardo.pinciroli,giuseppe.serazzi]@polimi.it

power consumption of a server was observed to have a linear relation with the CPU utilization. Such works present also a non-linear estimation, which however requires the computation of a coefficient that must be extrapolated from a large set of measurements collected from the analyzed system. Both models consider the CPU as the main index of the server activity. Since the CPU is involved in managing any server components, there is a strong correlation between the utilization of CPU and the utilization of all other resources. For such reason, the CPU utilization alone is representative of the server activity and thus of the entire power consumption.

The typical use of analytic expressions for the power computation is the inclusion of energy characterization in models of the system defined using suitable formalisms, such as queuing networks, Petri nets, and so on. Quite often the system models are designed to evaluate techniques or policies for energy saving aiming to analyze the benefits obtained on a single server. In most of the cases, the system models can provide estimates not only for the utilization, but also for other configuration settings such as the number of cores used and the frequency at which the system is running. Thus, taking into account widespread commercial power reduction techniques, like dynamic and voltage frequency scaling (DVFS) and hyperthreading, we can provide a more detailed power consumption expressions that can compute more accurate estimates.

In this paper we will focus on multicore CPUs that exploit hyperthreading, i.e., each physical core can execute two or more threads simultaneously. This number of thread is usually referred to as *Simultaneous MultiThreading* (SMT) *level*. The operating system (OS) can thus concurrently execute as many threads as the product of the number of cores and the SMT level. Moreover, the OS can control the so called *clock governor* by setting the maximum and minimum frequency that the DVSF mechanism can use. Initially, we will present a set of experiments designed to assess the impact on the power consumption of the number of cores used, of the level of SMT exploited and of the maximum clock frequency set. Then, we will show how the models presented in [9] may be affected by large relative errors if the above described features are not considered correctly. We finally propose a new power consumption model that can effectively take into account the considered CPU settings to provide more accurate energy estimations.

The paper is organized as follows. The main related work of the literature is described in Section 2. In Section 3, we describe the setup of the performed experiments. Section 4 presents and analyses the experimental results. Section 5 proposes various power models of increasing complexity able to capture the impact of hyperthreading and dynamic frequency scaling. The conclusion of the paper is provided in Section 6.

2 Related Work

As data-centers continue to consume an ever-greater share of the world's electricity, estimated to be 91 billion kilowatt-hours of electricity in 2013 only [13], a huge amount of researches are dedicated to improve the energy efficiency and develop effective energy-preserving strategies. These researches has produced a wide spectrum of techniques that exploit, albeit in different ways, two basic mechanisms: the dynamic scaling of components performances (Dynamic Speed Scaling) and the dynamic hibernation of components (Dynamic Resource Sleeping), see e.g.[2], [12], [16].

In the first studies that applied power control techniques on specific components of data-centers [9], the power consumption of a whole server was assumed proportional to the CPU utilization. The evidence of such a behavior was investigated and confirmed in further studies, see e.g. [17]. Recently, in order to provide more accurate estimates, authors started to isolate also the contribution of other components to the whole energy consumption [7]. Some of these newer works focus on CPU, see e.g., [11], other focus on memory, e.g. [10],while others on disks, [6].

To the best of our knowledge, only a few works propose power models taking into account hyperthreading and dynamic voltage and frequency scaling (DVFS) and none considers both of them together. For instance, [15] proposes an operating system facility to profile the power requirement of server requests, but without considering hyperthreading nor DVFS. Instead, [19] provides a detailed hyperthread-aware model to profile the power consumption of individual application, but ignore the DVFS impact. Such effect may be significant, as shown in [3] where the authors combine CPU indexes with instantaneous voltage demand information to improve the DVFS setting and achieve an average speedup of 7.3% over Windows Vista default DVFS algorithm.

In the literature, energy consumption behavior in data-centers has been modeled using different evaluation formalisms or metric models. Petri Nets are used in works like [8] where the authors, using a non-Markovian Stochastic Reward Networks modeling approach, investigate power-performance efficiency. Queueing theory has been widely applied for investigating optimal power allocation and load distribution in clouds, like e.g.[5]. In works like e.g.: [14] or [1], queueing theory is exploited for proposing tools to manage energy efficiency in datacenters. However, the effectiveness of such model may be hindered without a proper estimate of the power consumption of an individual server.

3 Experimental setup

We perform the power evaluation experiments on two different servers. The first called *Server1* is a PC with an i3-2120 CPU@3.3GHz. with two CPU cores and 6 GB of RAM, an hard disk drive and an integrated graphic card. *Server2* is a PC with an i7-3770 CPU@3.4GHz with four cores and 16 GB including two hard disk drives and a solid state disk with a dedicated GeForce GTX 560 graphic card. Both machines have an SMT level of two, meaning, for instance, that the i3-2120 CPU architecture provides two cores to the user, but is capable to concurrently run a total of four threads.

In all experiments the servers run CPU-bound applications, in particular instances of the Sunflow benchmark from the daCapo [4] suite. It renders a set of images using ray tracing, splitting the load into several concurrent threads. The number of threads used by the benchmark is set equal to the maximum number of threads that the server is capable to run concurrently. The server OS is Linux Ubuntu, but with different version: Ubuntu 12.04 for *Server1*, 14.04 for *Server2*. In both versions the maximum CPU frequency can be directly set editing the "scaling_max_freq" file. Moreover, it is possible to manually turn on and off each available threads editing the "online" file in "/sys/devices/system/cpu". It is also allowed to separately set the maximum frequency at which each thread run and switching off the multithreading on a specific CPU core.

The power consumption during each experiment is periodically measured by the the Yokogawa WT210 digital power meter [18] directly attached to the analysed server. The CPU utilization is monitored by the `io_stat` command which periodically produces CPU statistics calculated as the average among all threads. Finally, the CPU frequency is monitored by inspecting the "/proc/cpu_info" file.

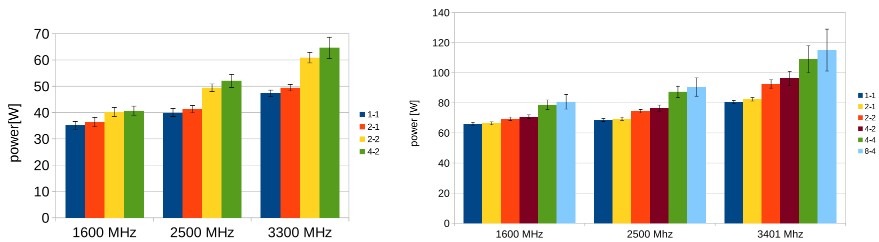
4 DVFS and Hyperthreading

In this Section we analyse the power consumption of a CPU-bound application taking in consideration two main technologies integrated in the CPUs: DVFS and hyperthreading. We expect a significant effect of such technologies on the power consumption. In fact, in a CMOS circuit the dynamic power [2] can be computed as $P_{dyn}(U) = \alpha \cdot V^2 fr$, where α is a positive constant. Thus, changing the clock frequency fr or the voltage V will clearly have an energetic impact. Furthermore, in hyperthreading, emulating an additional CPU core by concurrently executing two hardware threads should consume less power than using two physical cores. However, in both cases such effects are not captured by considering the CPU utilization only.

To investigate such phenomena, we measure through the power meter the server power consumption of the Sunflow benchmark fixing in each execution both the maximum CPU frequency and the number of threads used. We perform a full factorial experiment on the machine *Server*1 with a maximum CPU frequency ranging on the set $\{1600, 2500, 3300\}$ MHz. By selecting as maximum 1600 $MHz.$ the frequency scaling is disabled; with 2500 $MHz.$, during the execution of the benchmark, the CPU frequency may span from 1600 and 2500 $MHz.$, and so on. The number of concurrent threads ranges from 1 to 4. For each configuration 20 executions have been performed.

The histograms in Fig. 1(a) show the resulting power consumption with 95% confidence intervals. As expected, the power linearly increases with respect to the maximum CPU clock frequency. Such increment grows according to the number of parallel threads used; for instance, using four of them the increment from 1600 to 3300 MHz is slightly more than 50% (40 to 63 Watt). Instead, the power does not increases linearly according to the number of threads used, but it follows a step behavior with a gap of nearly 10 W from the second to third configurations. Indeed, the configurations with 1 or 2 threads involve just a single CPU core, while in the remaining ones both CPU cores work together, thus consuming more power.

An analogous set of experiments was performed on the *Server*2 machine with a maximum CPU frequency ranging on the set $\{1600, 2500, 3401\}$ MHz. and the number of parallel threads ranging from 1 to 8. As shown in Fig. 1(b), a similar behavior is obtained. In this case, a further gap is present using two or four CPU cores.



(a) (b)

Fig. 1 CPU-bound application power varying CPU frequency with several CPU cores and hyper-threading : (a) dual-core; (b) quad-core CPU.In the legend, a configuration with a number x of threads running on y CPU cores is denoted by the label $x-y$.

5 Power modeling

The first power model proposed in [9] for the estimate of warehouse-sized data-center consumption was the following:

$$P(U_{CPU}) = P_{Idle} + U_{CPU} \cdot (P_{Max} - P_{Idle}) \tag{1}$$

where U_{CPU} is the CPU utilization, P_{Idle} is the power consumed when no user applications are running and P_{Max} is the maximum power consumption when the server is 100% utilized. In the same work also a more accurate non-linear estimate was proposed:

$$P(U_{CPU}) = P_{Idle} + (2U_{CPU} - (U_{CPU})^r) \cdot (P_{Max} - P_{Idle}) \tag{2}$$

with r experimentally computed by data collected from the analyzed system.

We compute the values of the parameters P_{Max}, P_{Idle} and r by a fitting procedure from the collected experiments. The fitting procedure was performed by the DEPS Evolutionary Algorithm, a variant of the DEPSO algorithm proposed in [20], for non-linear optimization. It integrates together the Differential Evolution and Particle

Swarm Optimization techniques. To prevent over-fitting, we use half of the collected measurements to perform the fitting, thus computing the value of the parameters. The remaining half is used to validate the obtained model and compute the resulting mean absolute percentage error (MAPE)[1].

The straightforward application of such models to estimate the power consumption of the experiments in Section 4 provides results with a very low MAPE. Indeed, during all the experiments the CPU utilization is nearly 100%, thus according to Eq. 1 the energetic power consumption should be maximum, even if the server is not fully exploited since only a fraction of the available computational power (i.e. CPU frequency or number of threads and cores) is used.

As a first solution to avoid such problem, we weight the CPU utilization with the fraction of used threads:

$$U(C_{th}) = U_{CPU}\frac{C_{th}}{T_{th}} \tag{3}$$

where C_{th} is the number of used threads and T_{th} is the total number of threads that the CPU is capable to run concurrently. Using Eq. 3 to compute the models of Eq. 1 and 2 of the dual cores CPU, the values of MAPE obtained are 13.756% and 13.754%, respectively, as shown in Tab 1.

We can improve the results considering also the CPU frequency, thus defining the value of U_{CPU} as either one of the two following expressions:

$$U(fr,C_{th}) = U_{CPU}\tfrac{C_{th}}{T_{th}}\tfrac{fr}{fr_{Max}} \quad (a)\ , \quad U(fr,C_{cr}) = U_{CPU}\tfrac{C_{cr}}{T_{cr}}\tfrac{fr}{fr_{Max}} \quad (b) \tag{4}$$

where C_{th} (C_{cr}) is the number of used thread (cores), T_{th} (T_{cr}) is the total number of threads (cores). Moreover, fr and fr_{Max} are the average and maximum CPU frequency, respectively. We will call $O = \{C_{th}, C_{cr}, fr\}$ the operative parameters which may be set by software applications. Instead $M = \{T_{th}, T_{cr}, fr_{Max}\}$ is the set of machine parameters which are fixed and depend on the HW characteristics of the server.

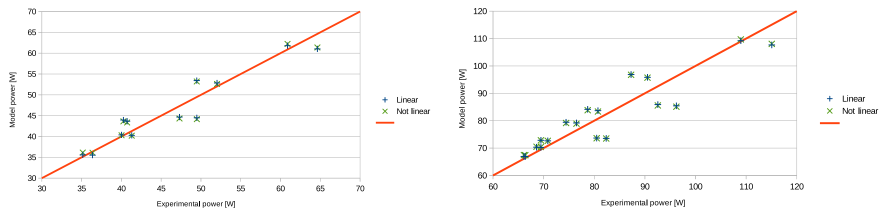
We performed the aforementioned fitting procedure between the power values collected in the experiments and the model of Eq. 1 (called Linear) and Eq. 2 (Not linear), for both the expressions in Eq. 4. Fig. 2 shows a comparison plot between the experimental and the estimated power consumption fitted using $U(fr,C_{cr})$, which achieves the best accuracy between the two expressions in Eq. 4. Both the linear and not linear models achieve nearly the same accuracy, in particular the linear model has a MAPE of 4.12% and 4.712% for the dual-core and quad-core CPU, respectively.

Our proposal to further improve the results is introducing the scaling factor:

$$\Delta(O,M) = \left(\frac{C_{th}}{T_{th}}\alpha_{log} + \frac{C_{cr}}{T_{cr}}\alpha_{cr}\right)\left(\frac{fr}{fr_{Max}}\right)^{\eta} \tag{5}$$

where α_{th}, α_{cr} and η are coefficients evaluated by the fitting procedure.

[1] The mean absolute percentage error is defined as: $MAPE = \frac{1}{N}\sum\left|\frac{A_t - F_t}{A_t}\right|$, where A_t is the actual value and F_t is the estimated one.



(a) (b)

Fig. 2 Comparison of simple linear and non linear models with experimental results:(a) dual-core; (b) quad-core CPU.

We define the maximum power consumption $P_{Max}(O,M)$, dependent on the current operative conditions and the server features, as:

$$P_{Max}(O,M) = P_{Idle} + \Delta(O,M) \cdot (P_{Max} - P_{idle}) \tag{6}$$

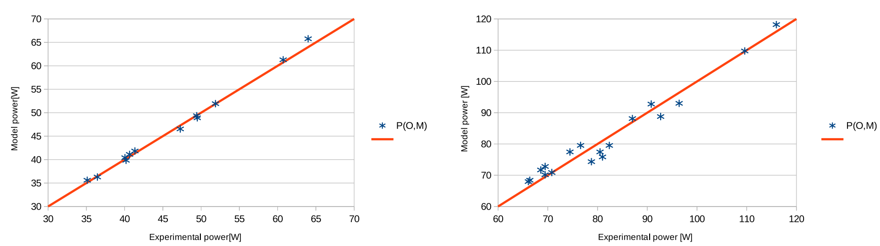
Finally, we define the power value $P(O,M)$ as:

$$P(O,M) = P_{Idle} + (P_{Max}(O,M) - P_{Idle})\, U_{CPU} \tag{7}$$

Substituting Eq. 6 in Eq. 7 we obtain:

$$P(O,M) = P_{idle} + \Delta(O,M) \cdot (P_{Max} - P_{Idle})\, U_{CPU} \tag{8}$$

As before, we perform a fitting procedure with the DEPS Algorithm to compute the coefficients α_{th}, α_{cr}, P_{Idle} and P_{Max}. The values obtained are shown in Table 1. The comparison plot between the resulting estimate and the experiment results is shown in Fig. 3. The accuracy is further improved with a MAPE of 1.14% for the dual core and 2.99% for the quad core.



(a) (b)

Fig. 3 Comparison of proposed model with experimental results: (a) dual-core; (b) quad-core CPU.

Table 1 Fitted parameters.

Cores	Power Model	U Def.	P_{Idle}	P_{Max}	α_{th}	α_{cr}	η	r	*error*
4	Eq. 1	Eq. 3	32.99	52.884	—	—	—	—	13.756%
4	Eq. 2	Eq. 3	32.99	52.931	—	—	—	0.934	13.754%
4	Eq. 1	Eq. 4(b)	28	61.363	—	—	—	—	4.12 %
4	Eq. 2	Eq. 4(b)	28	62.855	—	—	—	0.938	4.273%
4	Eq. 8	—	30.256	67.143	0.277	0.71	1.533	—	1.14%
8	Eq. 1	Eq. 3	65.147	92.761	—	—	—	—	9.943%
8	Eq. 2	Eq. 3	64.804	92.675	—	—	—	1.045	9.928%
8	Eq. 1	Eq. 4(b)	60.905	105.293	—	—	—	—	4.712 %
8	Eq. 2	Eq. 4(b)	65.225	11.216	—	—	—	0.731	4.456%
8	Eq. 8	—	63.708	115.514	0.364	0.723	2.101	—	2.99%

6 Conclusion

In this work we have considered the power consumption of CPUs with multiple cores, hyperthreading and dynamic frequency scaling. The next step will be including different types of resources such as disks, memory, and network: even if their contribution to the power budget of a server is much smaller than the CPU it is still important to include them for having a complete picture of the energy consumption of a server. The Graphic Processing Unit (GPU) can require even more energy than the CPU: this device however is needed only for very specific tasks like video-transcoding, and currently it is only marginally used in most of servers in a data-center. Finally, the impact of virtualization should be considered, since the "trap-and-execute" technique employed by most of the virtual machine managers correlates the virtualized resources and the utilization of the CPU on the hosts, leading to different energy foot-prints.

References

1. Aljohani, A., Holton, D., Awan, I., Alanazi, J.: Performance evaluation of local and cloud deployment of web clusters. In: Network-Based Information Systems (NBiS), 2011 14th International Conference on, pp. 274–278 (2011). DOI 10.1109/NBiS.2011.47
2. Beloglazov, A., Buyya, R., Lee, Y.C., Zomaya, A., et al.: A taxonomy and survey of energy-efficient data centers and cloud computing systems. Advances in computers **82**(2), 47–111 (2011)
3. Bircher, W.L., John, L.: Predictive power management for multi-core processors. In: Proceedings of the 2010 International Conference on Computer Architecture, ISCA'10, pp. 243–255. Springer-Verlag, Berlin, Heidelberg (2012). DOI 10.1007/978-3-642-24322-6_21. URL http://dx.doi.org/10.1007/978-3-642-24322-6_21
4. Blackburn, S.M., Garner, R., Hoffmann, C., Khang, A.M., McKinley, K.S., Bentzur, R., Diwan, A., Feinberg, D., Frampton, D., Guyer, S.Z., Hirzel, M., Hosking, A., Jump, M., Lee, H., Moss, J.E.B., Phansalkar, A., Stefanović, D., VanDrunen, T., von Dincklage, D., Wiedermann, B.: The dacapo benchmarks: Java benchmarking development and analysis. SIGPLAN Not. **41**(10), 169–190 (2006). DOI 10.1145/1167515.1167488

5. Cao, J., Li, K., Stojmenovic, I.: Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers. Computers, IEEE Transactions on **63**(1), 45–58 (2014). DOI 10.1109/TC.2013.122
6. Chen, D., Goldberg, G., Kahn, R., Kat, R., Meth, K.: Leveraging disk drive acoustic modes for power management. In: Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on, pp. 1–9 (2010). DOI 10.1109/MSST.2010.5496993
7. Economou, D., Rivoire, S., Kozyrakis, C.: Full-system power analysis and modeling for server environments. In: In Workshop on Modeling Benchmarking and Simulation (MOBS (2006)
8. Escheikh, M., Jouini, H., Barkaoui, K.: A versatile traffic and power aware performability analysis of server virtualized systems. In: Modelling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2014 IEEE 22nd International Symposium on, pp. 207–212 (2014). DOI 10.1109/MASCOTS.2014.34
9. Fan, X., Weber, W.D., Barroso, L.A.: Power provisioning for a warehouse-sized computer. SIGARCH Comput. Archit. News **35**(2), 13–23 (2007). DOI 10.1145/1273440.1250665. URL http://doi.acm.org/10.1145/1273440.1250665
10. Kant, K.: A control scheme for batching dram requests to improve power efficiency. In: Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '11, pp. 139–140. ACM (2011). DOI 10.1145/1993744.1993795
11. von Laszewski, G., Wang, L., Younge, A., He, X.: Power-aware scheduling of virtual machines in dvfs-enabled clusters. In: Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on, pp. 1–10 (2009). DOI 10.1109/CLUSTR.2009.5289182
12. Liu, Y., Zhu, H.: A survey of the research on power management techniques for high-performance systems. Software: Practice and Experience **40**(11), 943–964 (2010). DOI 10.1002/spe.952. URL http://dx.doi.org/10.1002/spe.952
13. Ltd., D.D.: Dcd industry census 2013: Data center power. Tech. rep. (2014). Available from: http://www.datacenterdynamics.com/critical-environment
14. Mitrani, I.: Trading power consumption against performance by reserving blocks of servers. In: M. Tribastone, S. Gilmore (eds.) Computer Performance Engineering, *Lecture Notes in Computer Science*, vol. 7587, pp. 1–15. Springer Berlin Heidelberg (2013). DOI 10.1007/978-3-642-36781-6_1. URL http://dx.doi.org/10.1007/978-3-642-36781-6_1
15. Shen, K., Shriraman, A., Dwarkadas, S., Zhang, X., Chen, Z.: Power containers: An os facility for fine-grained power and energy management on multicore servers. SIGPLAN Not. **48**(4), 65–76 (2013). DOI 10.1145/2499368.2451124. URL http://doi.acm.org/10.1145/2499368.2451124
16. Valentini, G., Lassonde, W., Khan, S., Min-Allah, N., Madani, S., Li, J., Zhang, L., Wang, L., Ghani, N., Kolodziej, J., Li, H., Zomaya, A., Xu, C.Z., Balaji, P., Vishnu, A., Pinel, F., Pecero, J., Kliazovich, D., Bouvry, P.: An overview of energy efficiency techniques in cluster computing systems. Cluster Computing **16**(1), 3–15 (2013). URL http://dx.doi.org/10.1007/s10586-011-0171-x
17. Vasan, A., Sivasubramaniam, A., Shimpi, V., Sivabalan, T., Subbiah, R.: Worth their watts? - an empirical study of datacenter servers. In: High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on, pp. 1–10 (2010). DOI 10.1109/HPCA.2010.5463056
18. Yokogawa Electric Corporation: URL http://tmi.yokogawa.com/discontinued-products/digital-power-analyzers/digital-power-analyzers/wt210wt230-digital-power-meters/
19. Zhai, Y., Zhang, X., Eranian, S., Tang, L., Mars, J.: Happy: Hyperthread-aware power profiling dynamically. In: Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference, USENIX ATC'14, pp. 211–218. USENIX Association (2014)
20. Zhang, W.J., Xie, X.F., et al.: Depso: hybrid particle swarm with differential evolution operator. In: IEEE International Conference on Systems Man and Cybernetics, vol. 4, pp. 3816–3821 (2003)