Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 870 – 877

Procedia Computer Science

The 6th International Conference on Sustainable Energy Information Technology (SEIT 2016)

EATS: Energy-Aware Tasks Scheduling in Cloud Computing Systems

Leila Ismail[a,*], Abbas Fardoun[b]

[a] *College of Information Technology, UAE University, Al-Ain, UAE*
[b] *College of Engineering, UAE University, Al-Ain, UAE*

Abstract

The increasing cost in power consumption in data centers, and the corresponding environmental threats have raised a growing demand in energy-efficient computing. Despite its importance, little work was done on introducing models to manage the consumption efficiently. With the growing use of Cloud Computing, this issue becomes very crucial. In a Cloud Computing, the services run in a data center on a set of clusters that are managed by the Cloud computing environment. The services are provided in the form of a Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). The amount of energy consumed by the underutilized and overloaded computing systems may be substantial. Therefore, there is a need for scheduling algorithms to take into account the power consumption of the Cloud for energy-efficient resource utilization. On the other hand, Cloud computing is seen as crucial for high performance computing; for instance for the purpose of Big Data processing, and that should not be much compromised for the sake of reducing energy consumption. In this work, we derive an energy-aware tasks scheduling (EATS) model, which divides and schedules a big data in the Cloud. The main goal of EATS is to increase the application efficiency and reduce the energy consumption of the underlying resources. The power consumption of a computing server was measured under different working load conditions. Experiments show that the ratio of energy consumption at peak performance compared to an idle state is 1.3. This shows that resources must be utilized correctly without scarifying performance. The results of the proposed approach are very promising and encouraging. Hence, the adoption of such strategies by the cloud providers result in energy saving for data centers.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Green Computing; Energy Efficiency; Energy Management; Scheduling; Cloud Computing; Performance

1. Introduction

Cloud computing [1,2,3] is an emerging technology for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. It is

* Corresponding author:
E-mail address: leila@uaeu.ac.ae

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.178

composed of three service models: Cloud Software as a Service (SaaS), Cloud Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). The Cloud SaaS provides the users with applications that they can run and get results. The Cloud PaaS provides the users with the possibility to deploy applications onto the cloud. The Cloud IaaS provides the users with the capability to provision processing, storage, and networks for running their applications. The services run in a data center on a set of clusters that are managed by the Cloud Computing environment. The Cloud computing is promising for high performance computing[4,5] of many scientific and engineering applications; for instance for the purpose of Big Data processing. Both performance and quality of services are essential. Many researchers were aiming at increasing the performance of applications by selecting adequate resources without paying attention to energy consumption. However, the energy consumption of the data center is equally a crucial issue for the both the entity hosting the data center as the corresponding bill increases in cost, and for the environment. For instance, Data centers in European Union are estimated to be about 1% in 2005 (including cooling) and it is estimated that it has reached 2.8% in the US. They are estimated to use 3% of the total electricity production in Europe, hence responsible for the same percentage of CO2 emission[6].

The data centers power consumption is expected to reach 100 TWh in year 2020[6]. The cost of the power consumed by a server during its lifetime could surpass its cost[7]. When a data center is built, a provision of power energy consumption is needed for building the infrastructure; i.e., the facilities, such as the cooling system and the electrical facilities. The facilities' cost is linearly dependent on the maximum utilization of the computing nodes; i.e., the servers operating at their peak performance. This cost is divided into a facility cost and a power consumption cost. Therefore, in order to amortize that cost, it is important to maximize the utilization of the computing machines. Underutilized resources become costly as a fraction of the total cost of ownership[8].

In Cloud computing, the hardware allocation is hidden to users. However, the distribution of users' applications, which is part of a Cloud Computing environment should take into consideration the energy efficiency of the cloud. Cloud server machines should not be overloaded, at risk of high power consumption, and execution inefficiency. They should not be underloaded, as the energy consumption of the computing facilities, designed for an efficiently-used data center, increases compared to the usage pattern[9]. This problem becomes crucial with the emergence of Big Data analysis in the Cloud[10], as infrastructure should be used in an energy-optimal way.

The goal of this work is to distribute a Big data for distributed processing in the Cloud in a way to decrease the overall energy consumption of the Cloud, without scarifying the application's performance. Therefore, we develop an energy-aware task scheduler (EATS) whose aim is to reduce both the power consumption of the utilized resources and the processing time of an application. We are considering applications of type divisible load applications[11]. Divisible load applications are a class of applications that can be divided into independent tasks and assigned to distributed resources with no synchronization and inter-tasks communication. Divisible load occurs in many scientific and engineering applications to analyze and process Big data, such as search for a pattern, compression, generic search applications, multimedia and video processing, image processing and data mining. Distribution of processing is meant to increase the performance of the distributed application compared to its sequential execution. In our previous works[11,12], we developed a scheduler to increase the performance of divisible load applications in a Cloud computing environment. The scheduler follows a linear-model approach and did not take into account the energy consumption of the cloud. In this work, we develop a non-linear-programming based scheduling model which aims at optimizing both the performance of the application and the energy consumption of the underlying resources. Our experiments show that the ratio of the energy consumption of a fully utilized server to its power consumption when it is idle is 1.3. This shows that idle, and underutilized resources consume a large amount of energy compared to that of full use. Our model takes into consideration this observation to optimize energy consumption during distributed computing.

The rest of the paper is structured as follows. Section 2 provides the system model of our scheduling approach. The scheduling algorithm is presented in section 3. Section 4 overviews related works and compare them to our approach. Our experiments and their evaluations are described in section 5. Section 6 concludes our work and highlights future works.

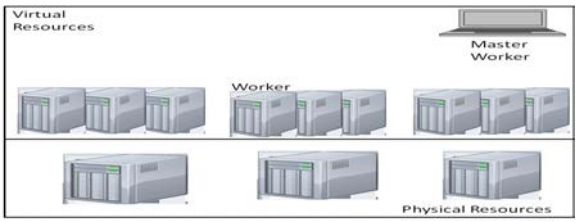

Fig. 1. Cloud Computing System Model.

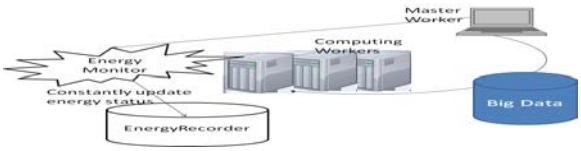

Fig. 2. Overall Architecture of EATS.

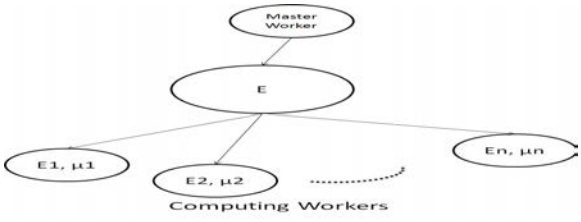

Fig. 3. Energy-Bounded Scheduling Algorithm.

2. EATS SYSTEM MODEL

Figure 1 shows a cloud computing system, where physical nodes are virtualized to users on a set of virtual machines. Each physical node can hold multiple virtual machines. To distribute a big data into chunks for distributed processing, among the virtual machines, one need a master worker and multiple computing workers. The computing workers are independently connected to the master worker via network connectivity. The virtual computing workers operate in an independent heterogeneous way. Each individual computer worker may consist of one or multiple virtual cores, which depends on the built-in capacity in terms of computing power, memory limitation and communication protocols of the shared physical resources. Upon receiving of an application or a users request, the master worker, or Broker, divides the whole application into a sequence of tasks, to be processed at the computing workers. Each task involves a part of the data of size *chunk*, and involves energy consumption for computing purpose. In our model, we do not account for network bandwidth. We assume that all the computing workers receive their corresponding chunks at the same time. To minimize the energy as well as to achieve an optimum performance efficiency, the master worker includes EATS. EATS takes into account the existing computing capacity of the computing worker, its available memory capacity, and its energy consumption for the sake of task partitioning, and scheduling.

Figure 2 shows the overall architecture of EATS. EATS includes an energy monitor which keeps information about the energy consumption of the computing workers on a physical node. The activities of each physical node is monitored and its energy consumption is recorded, as well as the energy consumption of the corresponding individual computing workers. The master worker, then consults the energy manager and takes distribution and scheduling decisions of the Big data to process.

As shown in Figure 3, EATS adopts the bounded-energy scheduler. Every virtual machine i, $i \in \{1, \ldots, N\}$, as an energy profile E_i. E_i is the maximum energy consumed by the computing worker i at peak performance. E is

the maximum energy consumed by all the computing workers simultaneously when they are all working at peak performance.

$$E = \sum_{i=1}^{i=N} E_i \tag{1}$$

When running the application, a computing worker i, registers a consumed energy value of, e_i, $i \in \{0, \ldots, E_i\}$

$$e_i \leq E_i \tag{2}$$

$$\sum_{i=1}^{i=N} e_i \leq E \tag{3}$$

A computing worker has a computing capacity, μ_i. An application consists of a total of W_{total} divisible computing loads. A portion of the total load, of size $chunk_i \leq W_{total}$, is processed by the computing worker worker i. We model the time required for a computing worker to perform the computation of $chunk_i$ units of load, TP_i, as follows. A unit of load is a unit of data input for an application. It could be one byte or several bytes and it is an application-dependent. For instance, one unit of load for the Ensemble Clustering data mining application[11], that we used for our experiments, is one record of raw data.

$$TP_i = \frac{chunk_i}{\mu_i} \tag{4}$$

where μ_i is the computational speed of the computing worker i in units of load per second. $chunk_i$ is the size of the data that is allocated to the computing worker i.

3. NON LINEAR PROGRAMMING SCHEDULING MODEL

As mentioned earlier, our main goal is to increase the speedup of an application by distributing its data and processing and to optimize energy consumption. EATS distributes a Big data of size W_{total} into N independent chunks, each of size $chunk_i$ $i \in \{1, \ldots, N\}$. The total execution time T of the distributed application is optimized. The execution time of the application is the time needed by the computing worker, which is executing last, to terminate its processing.

$$TP = \max TP_i \tag{5}$$

The above formula is then used to compare the performance of different distribution strategies over the set of computing workers. Also the current energy consumption e_i must be optimized but in a way to be equal or closer to the energy profile E_i.

4. BACKGROUND AND RELATED WORKS

The number of data center installations has been growing at a high rate of 16.7%, between 2000-2005, which corresponds to an increase of power consumption by 76%[13]. This tremendous increase in power consumption has called for new laws and regulations. In the USA, a new metric standard like the European code of conduct for data centers, the Energy star program for new metrics for data centers efficiency measurements and the 80 PLUS initiative[14,15,16], were introduced.

Works on energy efficiency can be divided into 2 main categories: 1) hardware-based solutions for energy efficiency and 2) software-based solutions for energy efficiency. Lately, there was a lot of work to optimize the energy consumption of computing components by designing energy-efficient circuits[17]. However, with the introduction of

large-scale cloud systems, energy consumption becomes a crucial issue. Then, software solutions were proposed, such as Power Nap [18]. Reference [9] proposes directions for correct power provisioning in warehouses. However, the proposed solutions do not consider optimizing energy consumption and application performance. [19] proposes algorithm which saves energy without compromising performance, based on task consolidation. However, it did not account for the problem of heterogeneity of computing powers. In our work, we propose an energy-aware scheduler which distributes a Big data on a heterogeneous platform, considering both performance and energy optimization.

5. PERFORMANCE ANALYSIS

In this section, we evaluate the impact of execution on energy consumption. We examine two types of applications, a computational-based application and another one which includes I/O operations.

5.1. Experimental Environment

In order to assess the impact of computing on the energy consumption, we conducted our experiments on a desktop with 2.13GHz of CPU and 1GB of memory. The desktop includes 2 single-core CPUs, running Windows Enterprise (SP1) operating system. Figure 4 shows the components of our experimental environment that were used to conduct the experiments. We measure the energy consumption, we connected the desktop to an Oscilloscope of type Tektronix TDS2012B. Current and voltage sensors were setup. The Oscilloscope is connected to the desktop by USB-to-USB cable. The Oscilloscope collects data samples at the rate of 1 Giga sample/second and the record length is 2500 sample/second. It has 100MHz of bandwidth. The Oscilloscope collects the Current and the Voltage of the desktop. The Oscilloscope uses Tektronix-based version of the Signal Express Software. The software reads the current and the voltage. One of the customization done to the software is changing the scale of the voltage and current, depending on the scale of the measuring probes used; this is for the sake of consistency between the value measured, and the value recorded on the software. Figure 5 shows a example of a measurement value of 529 mA (Milli-Ampere) which gives a power of 127 Watt. Oscilloscope uses Tektronix-based version of the Signal Express Software. The software reads the current and the voltage. One of the customization done to the software is changing the scale of the voltage and current, depending on the scale of the measuring probes used; this is for the sake of consistency between the value measured, and the value recorded on the software. Figure 5 shows a example of a measurement value of 529 mA (Milli-Ampere) which gives a power of 127 Watt. The power is measured in real time according to the following formula:

$$P(t) = v(t) * i(t) \tag{6}$$

where $v(t)$ is the voltage problem, and $i(t)$ is the current probe.

In addition, very accurate current and voltage probes are used to calibrate the oscilloscope power measurements. It shall be noted that the current measurement is not a pure sinusoidal waveform since the load acts as a nonlinear load. The desktop power at boot-up condition is shown in Figure 6. As shown in Figure 6, the real time power is not steady due to the harmonics in the current waveform. Note that at boot up, the power consumption reached a peak of about 138W and then it had an average of about 88W at steady state conditions. Matlab applications as well as data mining are used to load the desktop.

To measure the power consumption on computing, we wrote a simple program which includes a loop on a multiplication operation. The loop-based application runs through a Matlab, version R2010a. To know the impact of a Big Data processing application on the power consumption, we run a data mining application for big data analysis. The application, $K-$means, uses clustering algorithm [10] and was implemented in Java. We use the weka API from the Weka machine learning open source repository. We use the weka API from the Weka machine learning open source repository (http://www.cs.waikato.ac.nz/ml/weka/) for the core K-means clustering. To measure the energy consumption, 15 clusters are used.

We used the *Forest cover dataset* from the UCI repository. The dataset contains geospatial descriptions of different types of forests. It contains 7 classes and 54 attributes and around 581,000 instances (around 130MB size). We normalize the dataset to give same weight to each attribute. Also, for the clustering, we removed the class label to

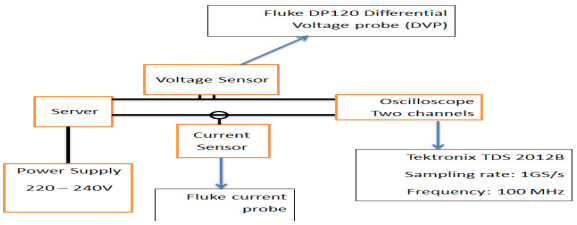

Fig. 4. Components of the Experimental Environment for Real Time Measurement of Energy Consumption.

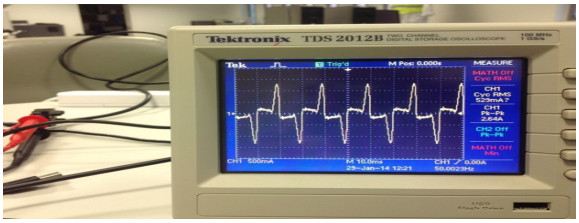

Fig. 5. Line Current Oscilloscope: Real Time Data Collection of Energy Consumption.

rule out any impact of unintended supervision during the clustering. For simulating experiments with large data, we produced a set of the forest cover data of 600,000 instances by replicating randomly chosen instances.

5.2. Experiments

The Oscilloscope measure the current RMS, in milliAmpere. We calculate the energy consumption, in Watt, by multiplying the current RMS and the RMS voltage, which is 240 in our experimental environment. In order to assess the impact of the different status of a computing worker on its energy consumption, we measure the energy consumption in different scenario. We measure the energy consumption of the computing worker while booting, and then reaching a steady/idle state. We also measure the energy consumption when running an interactive application such as Matlab. In order to assess the impact of the CPU computing on the energy consumption, we measure the energy consumption for a series of programs with increasing number of loops for a multiplication operation. To measure the impact of a Big data analysis application including I/O operations, we measure the energy consumption the K-means data mining application. To assess the impact of the data size on the energy consumption, we measure the energy consumption with increasing size of K-means data set. Each of our experiments was conducted 100 times and the average is computed.

5.3. Experimental Results Analysis

Figures 6, 7, 8, 9 and 10 show real time measurement of power consumption at different scenarios of execution of a computing worker. Figure 6 shows that, at boot, there is a peak of energy consumption for very few seconds. This suggests that frequent bootup of the machines might be a problem in a very large-scale cloud computing system. As shown in Figure 9, the power consumption increases with the use of the computing node. However, the ratio of the maximum energy consumption at 100% of the CPU usage, in our experimental environment, to the energy consumption of the computer worker at steady state is 1.3. That means, to amortize the cost of the Cloud data center and its facilities, it is important to run at 100% usage of the computing node. That suggests a bigger chunk of data sent to a computing worker is always better, but this should not compromise performance, keeping slow machines busy could become on the sake of faster ones, and therefore the performance would be compromised. EATS proposes a non-linear programming model to optimize both energy consumption and performance.

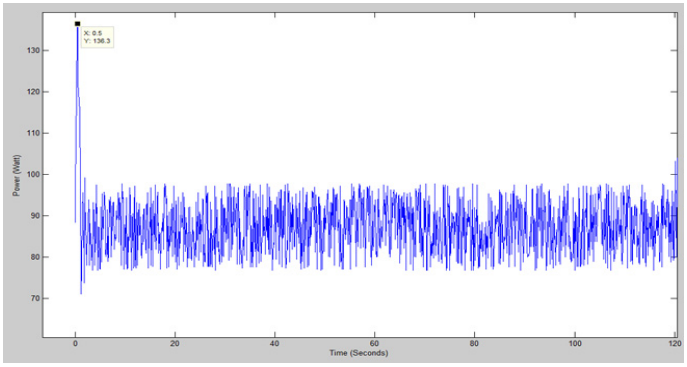

Fig. 6. Real Time Energy Consumption at Bootup and at Steady/Idle States.

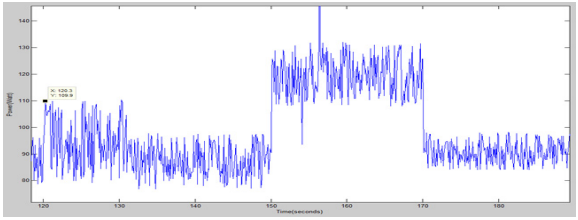

Fig. 7. Real Time Energy Consumption at Login.

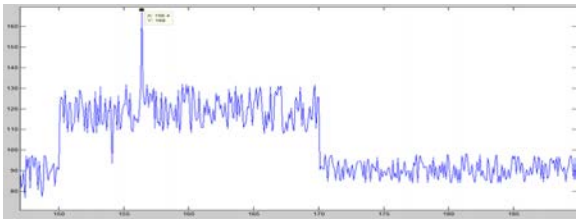

Fig. 8. Real Time Energy Consumption when Running Matlab.

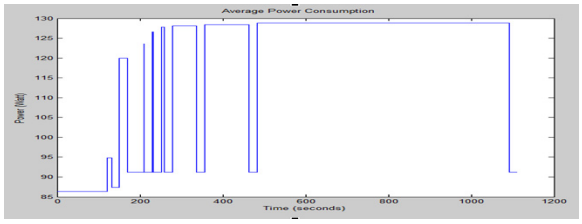

Fig. 9. Filtered Real Time Energy Consumption with Different Load Conditions.

6. Conclusion and Future Works

With the emergence of Clouds and Cloud computing, energy consumption of the underlying resources become crucial. In this work, we devise an energy-aware distribution and scheduling algorithm whose aim is to distribute a Big data for distributed processing taking into consideration bot performance and energy optimization. EATS solves a non-linear programming model take scheduling decisions. In this work, we conducted real power measurements on

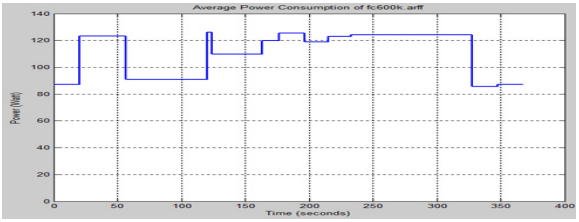

Fig. 10. Average Energy Consumption with KMeans Execution with 600k Instances of Data.

a computing desktop under different loads conditions. The experiments reveal an important issue is that the ratio of the energy measurement at peak performance to the energy measurement at idle time is 1.3, which is a call for servers utilization without scarifying performance which results in our non-linear programming scheduler. Future works of the proposed approach incorporate developing and implementing EATS for deployment in a Cloud computing environment to assess its performance.

References

1. R. Buyya, C.S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities", Keynote Paper, in Proc. 10th IEEE International Conference on High Performance Computing and Communications (HPCC 2008), IEEE CS Press, Sept. 257, 2008, Dalian, China
2. Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems". Volume 25, Issue 6, June 2009
3. M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, "Above the Clouds: A Berkeley View of Cloud computing", Technical Report No. UCB/EECS- 2009-28, University of California at Berkley, USA, February 10, 2009
4. Christian Vecchiola, Suraj Pandey, and Rajkumar Buyya, "High-Performance Cloud Computing: A View of Scientific Applicatications", Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN 2009, IEEE CS Press, USA), Kaohsiung, Taiwan, December 14-16, 2009
5. Leila Ismail, and Rajeev Barua, "Implementation and performance evaluation of a distributed conjugate gradient method in a cloud computing environment", Software: Practice and Experience, Wiley, Vol.43, Issue 3, Pages 281-304, March 2013
6. Paolo Bertoldi, "The European Programme for Energy Efficiency in Data Centres: The Code of Conduct ",http://re.jrc.ec.europa.eu/energyefficiency/html/standby_initiative_data_centers.htm, 2012
7. L. A. Barasso, "The price of performance: An economic case for chip multiprocessing", ACM Queue, 3(7), September 2005
8. W. P. Turner, J. H. Seader, and K. G. Brill, "Tier classifications define site infrastructure performance", The uptime Institute, White Paper, 2006
9. Xiaobo Fan, Wolf-Dietrich Weber, Luiz Andre Barasso, "Power Provisioning for a Warehouse-sized Computer", In Proceedings of the ACM International Symposium on Computer Architecture, 2007
10. Leila Ismail, M. Mehdy, and Latifur Khan, "SBD: A Framework for Scheduling of Big Data Mining in Cloud Computing", IEEE Cloud 2014, pp. 514-521, 2014
11. Leila Ismail, and Latifur Khan, "Implementation and performance evaluation of a scheduling algorithm for divisible load parallel applications in a cloud computing environment", Software: Practice and Experience. doi: 10.1002/spe.2258, Wiley, 2014
12. Leila Ismail, L. Zhang, K. Shuaib, S. Bataineh, "Performance Evaluation of Dynamic Single Round Scheduling Algorithm for Divisible Load Applications, the 18th International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA12: July 16-19, 2012, USA)
13. Jonathan G Koomey, "Worldwide electricity used in data centers", Environmental Research Letters, July 2008, ISSN 1748-9326, doi:10.1088/1748-9326/3/3/034008, http://stacks.iop.org/1748-9326/3/i=3/a=034008?key=crossref.976165ab72937d3bd0a21f91e350c756
14. http://www.energystar.gov/ia/partners/prod_development/downloads/DataCenters_AgreementGuidingPrinciples.pdf?6107-55e3
15. https://www.energystar.gov/sites/default/files/buildings/tools/Measurement
16. http://www.plugloadsolutions.com/80PlusPowerSupplies.aspx
17. Bart R. Zeydel and Vojin G. Oklobdzija, "Design of Energy-Efficient Digital Circuit", Springer, 2006
18. Meisner D, Gold BT, Wenisch TF, "Power Nap: eliminating server idle power", In Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'09), pp. 205-216, 2009
19. Young Choon Lee, Albert Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems", Journal of Supercomputer, 60:268-280, DOI 10.1007/s11227-010-00421-3, 2012