Heterogeneous Job Consolidation for Power Aware Scheduling with Quality of Service *

Fermin Armenta-Cano[1], Andrei Tchernykh[1†], Jorge M. Cortés-Mendoza[1], Ramin Yahyapour[2], Alexander Yu. Drozdov[3], Pascal Bouvry [4], Dzmitry Kliazovich[4], Arutyun Avetisyan[5]

[1]CICESE Research Center, [2]GWDG – University of Göttingen, [3]Moscow Institute of Physics and Technology, [4]University of Luxembourg, [5]ISP RAS

In this paper, we present an energy optimization model of Cloud computing, and formulate novel energy-aware resource allocation problem that provides energy-efficiency by heterogeneous job consolidation taking into account types of applications. Data centers process heterogeneous workloads that include CPU intensive, disk I/O intensive, memory intensive, network I/O intensive and other types of applications. When one type of applications creates a bottleneck and resource contention either in CPU, disk or network, it may result in degradation of the system performance and increasing energy consumption. We discuss energy characteristics of applications, and how an awareness of their types can help in intelligent allocation strategy to improve energy consumption.

1. Introduction

Cloud computing is an innovative distributed computing paradigm that is widely accepted by public and private organizations. The main objective of providers is to obtain maximum profits and guarantee QoS requirements of customers. One of the main concerns is energy expenditures. Intelligent job allocation strategies can be used to improve energy efficiency.

Inefficient resource management has a direct negative effect on performance and cost. In the shared environments, it is often difficult to optimize energy consumption of physical resources and virtual machines (VMs) with different type of tasks (CPU intensive, disk I/O intensive, memory intensive, network I/O intensive, etc.). Detailed energy management at granular levels should be used to optimize resource usage and improve profitability [1].

In this paper, we present an energy optimization model in Cloud computing that takes into account different types of applications. We propose a heterogeneous job consolidation algorithm for power aware scheduling to optimize energy consumption. We evaluate power efficiency of our strategy and compare it with the best in the literature under different scenarios.

The paper is structured as follow. The next section reviews related work on the energy optimization. Section 3 presents the problem definition, while the proposed scheduling algorithms are described in Section 4. Section 5 concludes the paper by presenting main contribution and future work.

2. Related work

Reducing energy consumption in Cloud computing has emerged as one the main research issues both in industry and academia. This is due to the fact that the energy required by the datacenters for its operation, power supply, and cooling, contribute significantly to the total operational costs.

In this section, we discuss power aware resource allocation algorithms presented in the literature.

* This work is partially supported by CONACYT (Consejo Nacional de Ciencia y Tecnología, México), grant no. 178415. Drozdov is supported by the Ministry of Education and Science of Russian Federation under contract No02.G25.31.0061 12/02/2013 (Government Regulation No 218 from 09/04/2010). The work of P. Bouvry and D. Kliazovich is partly funded by Green@Cloud (INTER/CNRS/11/03) and ECO-CLOUD (C12/IS/3977641) projects.

† Corresponding author

EMVM- Energy-aware resource allocation heuristics for efficient management [2]. The authors define an architectural framework and principles for energy-efficient Cloud computing. They present resource provisioning and allocation algorithms utilizing the dynamic consolidation of VMs for energy-efficient management of Cloud computing environments. The approach is validated by conducting a performance evaluation study using the CloudSim toolkit. It is shown that the approach leads to a substantial reduction of energy consumption in Cloud data centers in comparison to static resource allocation techniques.

Presented power consumption model is the following.

$$P(u) = k * Pmax + (1 - k) * Pmax * u,$$

where Pmax is the maximum power consumed when the server is fully utilized; k is the fraction of power consumed by the idle server (i.e. 70%); and u is the CPU utilization.

The total energy consumption E is defined as an integral of the power consumption function over a given period of time

$$E = \int_{t_0}^{t_1} P(u(t))dt.$$

When VMs do not use all provided resources, they can be logically resized and consolidated to the minimum number of physical nodes. While idle nodes can be switched to the sleep mode to eliminate the idle power consumption and reduce the total energy consumption by the data center.

Fig. 1 shows the percentage of energy consumption due to CPU utilization used in this work.

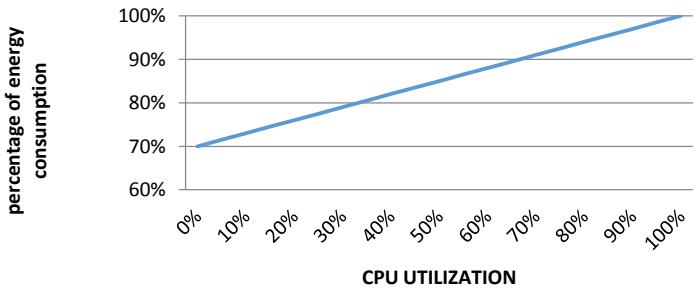

Fig. 1. Percentage of energy consumption due to CPU utilization (%).

HSFL- Hybrid shuffled frog leaping algorithm [3]. The authors propose a data center resource management scheme. It can not only guarantee user quality of service (QoS) specified by SLAs, but also achieve maximum energy saving and green computing goals. Consolidation of resources is achieved by VM migrations technology. Low utilized and idle hosts are switched to power saving mode to achieve energy saving while ensuring that SLAs are adhered to.

Host energy consumption exhibits an almost linear proportion to CPU energy consumption. Moreover, the energy consumption of an idle host accounts for 70% of full-load operation energy consumption. The energy consumed by VM migrations also requires consideration. Energy consumption within a given unit time is defined as follows

$$E(h) = 0.7E_{max}(h) + 0.3Utlz(h)E_{max}(h) + 0.1E_{max}\sum_{i \in v} T(i).$$

$E_{max}(h)$ is the energy consumption when host h is in full load. Utlz(h) is the average utilization rate of the host processor within unit time, v is the collection of VM migrations within the unit time window, and T(i) is the migration time of VM i. The percentage of energy consumption due to CPU utilization is similar to Fig. 1.

AETC- Algorithm of energy-aware task consolidation [4]. The authors propose a technique of energy-aware task consolidation (ETC) to minimize energy consumption. ETC restricts CPU use below a specified peak threshold by consolidating tasks amongst virtual clusters. In addition, the energy cost model considers network latency, when a task migrates to another virtual cluster. They define a default CPU utilization threshold of 70% to demonstrate task consolidation management amongst virtual clusters. Although the idle state of virtual machines and network transmission are assumed to be a constant ratio of basic energy consumption unit in his study. The simulation results

show that ETC can significantly reduce power consumption when managing task consolidation for Cloud systems. ETC is designed to work in a data center for VMs that reside on the same rack or on racks where network bandwidth is relatively constant.

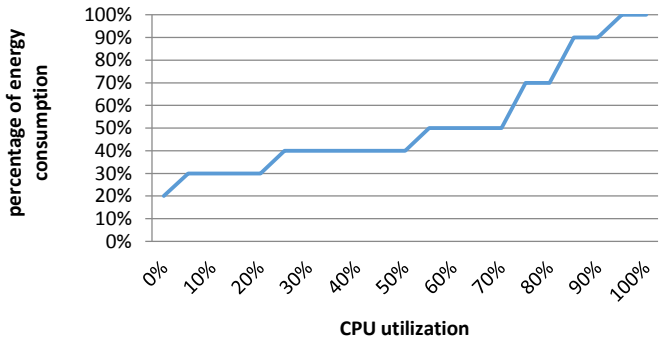

Fig. 2. Stepwise energy consumption due to CPU utilization (%).

The model assumes energy consumption $E(V_i) = \alpha\ W/s$ in the idle state. An additional energy β is required for executing tasks when CPU utilization is increased.

$$E(Vi) = \begin{cases} \alpha\, W/s, & if\ is\ idle \\ \beta + \alpha\, W/s, & if\ 0\% < CPU\ util \le 20\% \\ 3\beta + \alpha\, W/s, & if\ 20\% < CPU\ util \le 50\% \\ 5\beta + \alpha\, W/s, & if\ 50\% < CPU\ util \le 70\% \\ 8\beta + \alpha\, W/s, & if\ 70\% < CPU\ util \le 80\% \\ 11\beta + \alpha\, W/s, & if\ 80\% < CPU\ util \le 90\% \\ 12\beta + \alpha\, W/s, & if\ 90\% < CPU\ util \le 100\% \end{cases}$$

The energy consumption of a virtual machine V_i is defined as follows. The total energy consumption of V_i during the time period $t_0 \sim t_m$ is given the following formula:

$$E_{0,m}(V_i) = \sum_{t=0}^{m} E_t(V_i).$$

Given a virtual cluster, VC_k which consists of n VMs, the energy consumption of VC during the time period $t_0 \sim t_m$ tm is as follows:

$$E_{0,m}(VC_k) = \sum_{i=0}^{n} E_{0,m}(V_i).$$

Fig. 2 shows the percentage of energy consumption due to CPU utilization.

CTES- Cooperative Two-Tier Energy-Aware Scheduling [5]. The authors propose a cooperative two-tier task scheduling approach to benefit both Cloud providers and their customers. It regulates the execution speeds of real-time tasks in a way that a host reaches the optimum level of utilization instead of migrating its tasks to other hosts. They also propose several predictive global task scheduling policies to map arrived tasks to feasible VM, in his technique, a host is locally scheduled to reach its optimum CPU usage instead of migrating its tasks to other hosts. They divide the energy consumption of a host into two parts, static and dynamic energies. His simulation results show that the proposed task scheduling approach reduces the total energy consumption of a Cloud.

The utilization of a host, u is defined as: $u_i(t) = \frac{ah_i(t)}{mh_i}$, where $ah_i(t)$ are the allocated MIPS of $host_i$ in time t and mh_i is the maximum computing power of $host_i$

We suppose that an idle host changes its state to be powered off immediately. Thus, the total power of a host is defined as:

$$P = \begin{cases} P^{static} + P^{dynamic} & u > 0 \\ 0 & o.w \end{cases}$$

P^{static} is the power consumed during the idle time of a computing node. It is defined as $P^{static} = \propto P^{max}$. P^{max} is the power consumed when a host works with its maximum utilization. Utilization. α is the constant ratio of the static power of a host to its maximum power ($0 < \alpha < 1$) which depends on the physical characteristics of a host.

Dynamic power consumption is:

$$P_i^{dynamic} = \left(P_i^{max} - P_i^{static}\right)\ u_i^{\gamma}(t).$$

If the system uses the power P(u), the energy consumption will be $E = \int_0^{t_{min/u}} P(u)dt$ where t_{min} is the time in which a host works at its maximum computing power to finish a certain number of instructions. The percentage of energy consumption due to CPU utilization is similar to Fig. 1.

Therefore, the energy consumption of a host to finish its certain amount of instructions is obtained by:

$$E = [\alpha + (1-\alpha)u^{\gamma}]\frac{P^{max}t_{min}}{u}$$

DVMA- A Decentralized Virtual Machine Migration Approach [6]. The authors propose a decentralized virtual machine migration approach inside the data centers for Cloud computing environments which use virtual machines to host many third-party applications. They define a system models and power models then; they present the key steps of the decentralized mechanism, including the establishment of load vectors, load information collection, VM selection, and destination determination. A two-threshold decentralized migration algorithms is implemented to further save the energy consumption as well as keeping the quality of services. Performance evaluation results of their simulation experiments illustrate that their approach can achieve better load balancing effect and less power consumption than other strategies.

An idle physical node even with 0% of utilization could still consume a plenty of power. Let α be the fraction of power consumed by an idle node compared to a full utilized node and θ the current CPU utilization of the node. Then, we use the power model defined as follows to compute the power consumption of a physical nodes PN_i: $P_i = \alpha * P_i^{max} + (1-\alpha) * P_i^{max} * \theta$, where P_i^{max} is the power consumption of PN_i when it is fully utilized (i.e., it reaches 100% of CPU utilization). The percentage of energy consumption due to CPU utilization is similar to Fig. 1.

EDRP- Energy and Deadline Aware Resource Provisioning [7]. The authors addresses the problem of minimizing the operation cost of a Cloud system by maximizing its energy efficiency while ensuring that user deadlines as defined in Service Level Agreements are met. They take into account two types of workload models, independent batch requests and task graphs with dependencies.

The power consumption of Dx at time t includes the static power consumption $P_{static}^{x}(t)$ and the dynamic power consumption $P_{dynamic}^{x}(t)$. Both are correlated with the utilization rate of Dx at time t: $Util_x(t)$. We evaluate $Util_x(t)$ by considering only the CPU requirements of the hosted VMs indicated in $Q^x(t)$, and do not differentiate between VMs that are running tasks and idle VMs, since background CPU activities are needed even during idle periods. $P_{static}^{x}(t)$ is constant when $Util_x(t) > 0$, 0 otherwise. The relationship between $P_{dynamic}^{x}(t)$ and $Util_x(t)$ is much more complex. Servers have optimal utilization levels in terms of performance-per-watt, which we define as Opt_x for Dx. It is commonly accepted that for modern servers $Opt_x \approx 0.7$, and the increase in power consumption beyond this operating point is more drastic than when $Util_x(t) < Opt_x$. Even for identical utilization levels, the energy efficiency of different servers may vary. This is captured by the coefficients αx and βx, representing the power consumption increase of Dx when $Util_x(t) < Opt_x$ and $Util_x(t) \geq Opt_x$ respectively. $P_{dynamic}^{x}(t)$ is then calculated as:

$$\begin{cases} Util_x(t) * \alpha_x & \text{if } (Util_x(t) < Opt_x) \\ Opt_x * \alpha_x + (Util_x(t) - Opt_x)^2 * \beta_x & \text{if } (Util_x(t) \geq Opt_x) \end{cases}$$

We would like to point out that the exact formulations of $P_{dynamic}^{x}(t)$ do not undermine the analysis, since its increment is faster when $Util_x(t) \geq Opt_x$ than when $Util_x(t) < Opt_x$.

Suppose the upper bound of the maximum schedule length of all applications is L_{max}. The total energy consumption (COSP) is the sum of the power consumption across all servers throughout the operation timeline:

$$\mathrm{COSP} = \sum_{x=1}^{M}\left(\sum_{t=1}^{L_{max}}\left(P^{x}_{static}(t) + P^{x}_{dynamic}(t)\right)\right).$$

In Fig. 3, we show the nonlinear percentage of energy consumption due to CPU utilization used in this work.

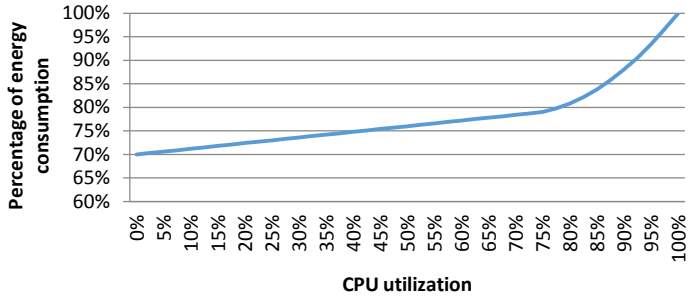

Fig. 3. Nonlinear energy consumption due to CPU utilization (%).

BFDP- Best Fit Decreasing Power [8]. The authors propose a simulation-driven methodology with an energy model based on polynomial regression with Lasso to predict energy consumption to verify its performance, and a resource scheduling algorithm BFDP shifting its optimization goal from resource consolidation to power consumption to improve the energy efficiency without degrading the QoS of the system and they consider four type of jobs, CPU-intensive, Memory-intensive, Network-intensive and I/O-intensive. The authors introduced the mechanism of utilization thresholds in BFDP to alleviate the over-consolidation issue in the Best-Fit strategy. Their results showed that are effective because BFDP creates less SLA violations than the BFDR in light workloads.

The authors uses a nonlinear energy model:

$$y_i = \beta_0 + \sum_{j=1}^{m}\beta_j\phi_j(x_i) + \varepsilon_i,$$

where $\phi_j(x_i)$ is the kernel function of expression y_i, x_i is the CPU and memory utility. β_i is the parameter of the kernel function to be determined through the model training process. ε_i is a constant. Fig. 4 presents a relationship between CPU and memory utilization and full-system power.

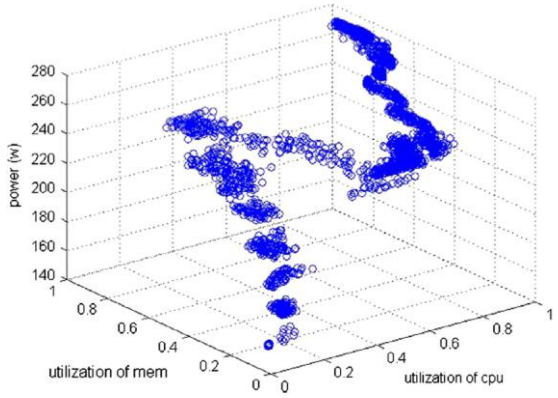

Fig. 4. Relationship between CPU and memory utilization and full-system power [8].

PAHD- Power-aware Applications Hybrid Deployment [9]. The authors present I/O Intensive and CPU-Intensive applications hybrid deployment to optimize resource utilization within virtualization environments. To demonstrate the problem of I/O and CPU resource in virtualization environment,

they use Xen as the Virtual Machine Monitor to make experiments. Under different resource allocation configurations, they evaluate power efficiency up to 2%~12%, compared to the default deployment. Finally they conclude that if the CPU-Intensive application is allocated twice as much CPU compared to I/O-Intensive application, there are an improvement in the power efficiency.

Table 1 shows the summary of the algorithm domains, the main characteristics of described algorithms, and the criteria used to evaluate quality of the algorithms.

Table 1. Related work algorithms.

	Application domain				Characteristics											Evaluation criteria					Ref
	Data centers	Cloud	Hybrid Cloud	Centralized	Decentralized	Hybrid	On line	Off line	Clairvoyant	Nonclairvoyant	QoS	Migration	Static	Dynamic	CPU Intenvive	I/O Intensive	Response time	Utilization	Energy	Deadline	
EMVM		●		●			●			●	●	●		●	●			●	●	●	[2]
HSFL		●		●			●				●	●		●	●			●	●		[3]
AETC		●		●				●	●		●	●			●			●	●		[4]
CTES		●			●		●		●		●			●	●			●	●	●	[5]
DVMA		●			●						●	●			●			●	●		[6]
EDRP		●		●				●	●		●	●	●		●			●	●	●	[7]
BFDP		●		●					●		●	●		●	●	●		●	●		[8]
PAHD	●										●				●	●		●	●		[9]

3. Problem definition

We assume that m servers of the data center are identical, and described by tuples {m, s, mem, band, eff }, where s is a measure of instruction execution speed (MIPS), mem is the amount of memory (MB), band is the available bandwidth (Mbps), and eff is energy efficiency (MIPS per watt). We also assume that data centers have enough resources to execute any job.

The main objective of the proposed strategies is to minimize the total power consumption E of running workloads providing QoS guarantees.

3.1 Job model

We consider n independent jobs $J_1, J_2, \dots, J_n$. The job J_j is described by a tuple $J_j = (r_j, p_j, type_j, d_j, SL_j)$, where $r_j \geq 0$ is the released time, p_j is a processing time. The release time r_j of a job is not available before the job is submitted. SL_j is the SLA from a set $SL = \{SL_1, SL_2, \dots, SL_j, \dots, SL_k,\}$ offered by the provider [19, 10, 20]. Each SLA represents a SL guarantee, and is denoted by the slack factor $f_i \geq 1$. d_j is the deadline of the job J_j and is calculated at the release of the job as $d_j = r_j + f_i * p_j$. Finally $type_j$ characterizes a job as CPU intensive, disk I/O intensive, memory intensive, network I/O intensive, etc.

3.2 Energy model

We present a nonlinear model of the power consumption by considering types of applications. Fig. 5 shows examples of the normalized power consumption of jobs of type A and type B vs CPU utilization (%). Characteristics of CPU intensive, disk I/O intensive, memory intensive, network I/O intensive, etc. applications influence on power consumption differently due to corresponding hardware characteristics.

Moreover, an allocation of two different applications to the same server could cause reduced power consumption, less than the sum of their individual power consumptions. It also has an impact on performance enhancement avoiding creation of a bottleneck and resource contention (either in CPU,

disk or network) that may result in additional degradation of the system performance and increased energy consumption.

We propose a hybrid model that takes into account power consumption of individual jobs and their combinations. Due to diversity of applications and their combinations, we propose to consider aggregated utilization of each type of applications (total utilization that contributes each job type or concentration).

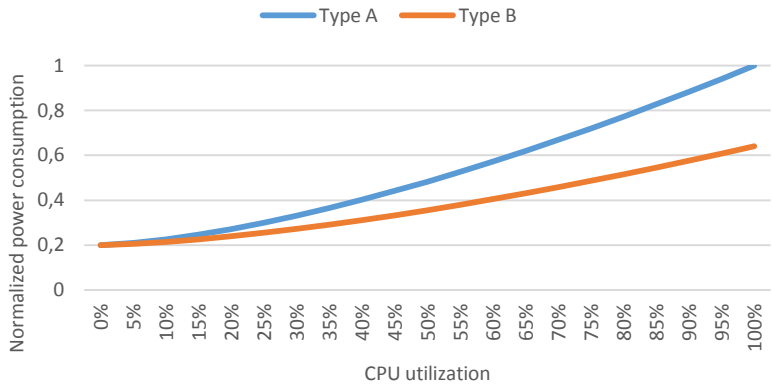

Fig 5. Normalized power consumption of jobs A and B vs CPU utilization (%).

The power consumption of the processor at time t consists of two parts: an idle power consumption when the processor is turned on, but not used $e_{idle_i}^{proc}$, and power consumption when the processor is in use $\mathrm{e}_{\mathrm{used_i}}^{\mathrm{proc}}(t)$:

$$e_i^{proc}(t) = o_i(t) * \left(e_{idle_i}^{proc} + \mathrm{e}_{\mathrm{used_i}}^{\mathrm{proc}}(\mathrm{t}) * \mathrm{U_i(t)}^r\right), \quad (1)$$

where $\mathrm{o_i(t)} = 1$, if the processor is *on* at time t, and $\mathrm{o_i(t)} = 0$ otherwise. $\mathrm{U_i(t)}$ is the utilization at time t. r is a coefficient proposed in [15] to fit non-linear power profiles.

$$\mathrm{e}_{\mathrm{used_i}}^{\mathrm{proc}}(\mathrm{t}) = \left(\left(e_{max_i}^{proc} - e_{idle_i}^{proc}\right) * \beta(\alpha_A(t))\right), \quad (2)$$

where $e_{max_i}^{proc}$ is the maximum power consumption when the processor is fully utilized.

$\beta(\alpha_A(t))$ is the coefficient that represents the increment of power consumption when a processor runs different types of applications. The concentration of type A vs type B at the time t is defined as $\alpha_A(t)$.

$$\beta(\alpha_A(t)) = \begin{cases} 0, & if\ is\ idle \\ 0.55 - \alpha_A(t) * 0.1, & if\ 0 < \alpha_A(\mathrm{t}) \leq 0.5 \\ 0.55 + \alpha_A(t) * 0.2, & if\ 0.5 < \alpha_A(\mathrm{t}) \leq 0.7 \\ 0.55 + \alpha_A(t) * 0.35, & if\ 0.7 < \alpha_A(\mathrm{t}) \leq 0.9 \\ 0.55 + \alpha_A(t) * 0.45, & if\ 0.9 < \alpha_A(\mathrm{t}) \leq 1 \end{cases} \quad (3)$$

Fig. 6 shows the proportion of power consumption, when the processor runs different jobs A and B.

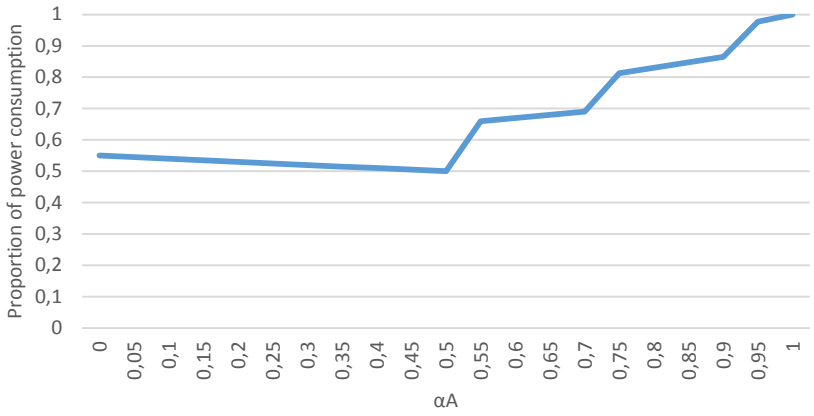

Fig 6. Proportion of the power consumption vs concentration of jobs A.

The total power consumed by the system is the integral of power consumed during operation:

$$E^{op} = \int_{t=1}^{C_{max}} E^{op}(t)dt, \text{ with } E^{op}(t) = \sum_{i=1}^{m} e_i^{proc}(t). \quad (4)$$

We define $e_{idle_i}^{proc} = 0.2 * e_{max_i}^{proc}$. Following [11], with the power consumption of a processor Fujitsu PRIMERGY TX300 S7, we set $r = 1.5$. We set $\beta(\alpha_A) = 1$, for $\alpha_A = 1$ (all jobs are type A), and $\beta(\alpha_A) = 0.55$ for $\alpha_A = 0$ (all jobs are type B).

Fig. 7 shows the normalized power consumption when the processor runs two types of applications.

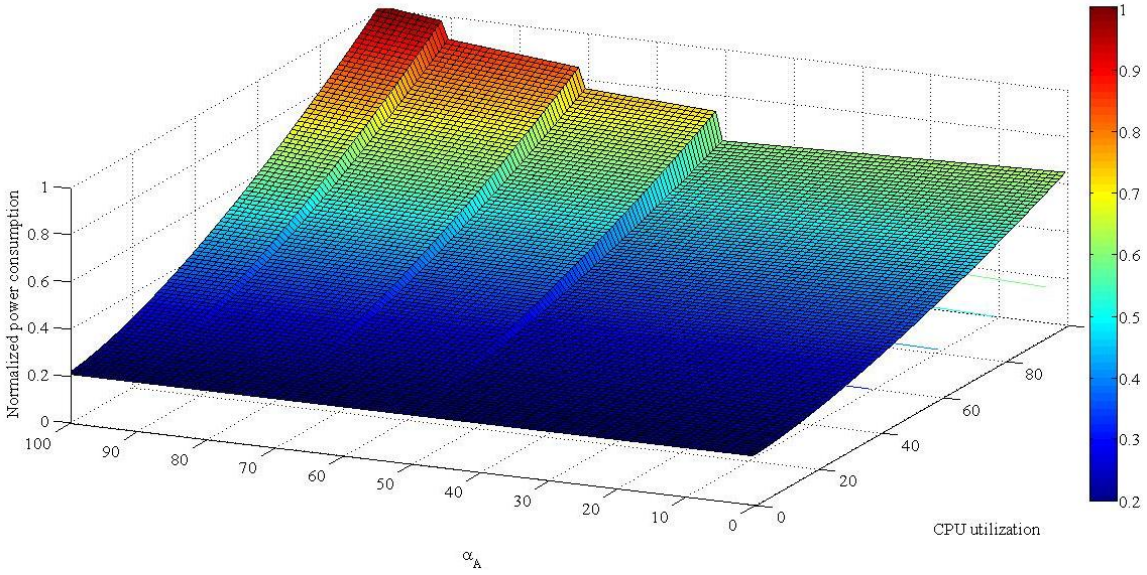

Fig 7. Normalized power consumption of job A and B vs CPU utilization (%) and concentration of jobs A (%).

4. Scheduling algorithms

In this section, we describe our scheduling approach and proposed energy-aware scheduling methods.

4.1 Scheduling approach

We address a two-level scheduling approach [12, 13, 14, 15]. At the upper level, the system verifies whether a job can be accepted or not using a *Greedy acceptance policy*. If the job is accepted then the system selects a machine from the set of admissible machines to execute the job on the lower level.

The greedy higher-level acceptance policy is based on the preemptive Earliest Due Date (EDD) algorithm, which gives priority to jobs according to their deadlines. When a job j arrives to the system, in order to determine whether to accept or reject it, the system searches for the set of machines capable of executing the job j before its deadline assuring that no jobs in the machine will miss their deadlines. If the set of available machines is not empty $(|M^a(r_j)| \geq 1)$ job j is accepted otherwise it is rejected. This completes the first stage of scheduling.

Note that the preemptive EDD algorithm is well suited for our purpose as it is easy to apply and it yields an optimal solution for the $1 \mid \text{prmp}, r_j, \text{online} \mid \text{Lmax}$ problem. By EDD, we verify that all already accepted jobs with a deadline greater than the deadline of the incoming job will be completed before their deadline.

4.2 Allocation strategies

The machine for job allocation can be determined by taking into account different criteria. In this work, we study ten allocation strategies Rand, FFit, RR, ML, MTe, Me, Mu, Mau, Mujt, and Mc. (see Table 2). They are characterized by the type and the amount of information used for allocation decision.

We categorize the proposed methods in three groups: (1) *knowledge-free*, with no information about applications and resources [16, 17, 18]; (2) *energy-aware*, with power consumption information; and (3) *utilization-aware* with utilization of machines information.

Table 2. Job allocation strategies.

Type	Strategy	Description
Knowledge Free	Rand	allocates job j to a suitable machine randomly selected using a uniform distribution in the range $[1..m]$.
	FFit (First Fit)	allocates job j to the first machine available and capable to execute it.
	RR (Round Robin)	allocates job j to the machine available and capable to execute by Round Robin strategy
	ML (Min load)	allocates job j to the machine with the least load at time r_j: $min_{i=1..m}\{n_i\}$,
Energy aware	MTe (Min-Total_energy)	allocates job j to the machine with minimum total power consumption at time r_j : $min_{i=1..m}\left(\sum_{t=1}^{r_j} e_i^{proc}(t)\right)$
	Me (Min-energy)	allocates job j to the machine with minimum power consumption at time r_j : $min_{i=1..m}\left(e_i^{proc}(r_j)\right)$
Utilization aware	Mu (Min-utilization)	allocates job j to the machine with minimum total utilization at time r_j $min_{i=1..m}\left(u_i^{proc}\right)$
	Mau (Max-utilization)	allocates job j to the machine with maximum total utilization at time r_j $max_{i=1..m}\left(u_i^{proc}\right)$
	Mujt (Min-util_job_type)	allocates job j to the machine with minimum utilization of jobs of the same type at time r_j
	Mc (Min-concentration)	allocates job j to the machine with minimum concentration of jobs of the same type at time r_j

5. Conclusions

In this paper, we consider the problem of energy optimization in Cloud computing from the perspective of the Cloud service provider. We formulate and discuss the energy model and energy-aware resource allocation problem that provide energy-efficiency and QoS guarantees simultaneously by heterogeneous job consolidation taking into account types of applications. A generic Cloud computing environment has to process multiple applications for multiple users, which create mixed workloads of different types with different energy consumption.

We consider energy characteristics of applications such as CPU intensive, disk I/O intensive, memory intensive, network I/O intensive, etc. and their influence on power consumption due to the nature of used hardware. We discuss how an awareness of the job type could help to improve energy consumption. Intelligent job allocation has an impact on performance enhancement avoiding creation of bottlenecks and resource contentions either in CPU, disk or network, and on decreasing total energy consumption.

We propose a hybrid model that take into account the power consumption of individual jobs and their combination. We propose using aggregated utilization of applications, and their concentration for job allocation.

However, further study for energy consumption of multiple job types and their concentration is required to assess the actual efficiency and effectiveness of the proposed method. This will be the subject of future work for better understanding of the resource contentions and its impact on the energy consumption, QoS and multi-objective optimization in clouds.

References

1 D. Kliazovich, J. E. Pecero, A. Tchernykh, P. Bouvry, S. U. Khan, A. Y. Zomaya, "CA-DAG: Modeling Communication-Aware Applications for Scheduling in Cloud Computing," Journal of Grid Computing, 2015.

2 A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Gener. Comput. Syst., vol. 28, no. 5, pp. 755–768, May 2012.

3 J. Luo, X. Li, and M. Chen, "Hybrid shuffled frog leaping algorithm for energy-efficient dynamic consolidation of virtual machines in cloud data centers," Expert Syst. Appl., vol. 41, no. 13, pp. 5804–5816, Oct. 2014.

4 C.-H. Hsu, K. D. Slagter, S.-C. Chen, and Y.-C. Chung, "Optimizing Energy Consumption with Task Consolidation in Clouds," Inf. Sci., vol. 258, pp. 452–462, Feb. 2014.

5 S. Hosseinimotlagh, F. Khunjush, and S. Hosseinimotlagh, "A Cooperative Two-Tier Energy-Aware Scheduling for Real-Time Tasks in Computing Clouds," in Proceedings of the 2014 22Nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, Washington, DC, USA, 2014, pp. 178–182.

6 X. Wang, X. Liu, L. Fan, and X. Jia, "A Decentralized Virtual Machine Migration Approach of Data Centers for Cloud Computing," Math. Probl. Eng., vol. 2013, p. e878542, Aug. 2013.

7 Y. Gao, Y. Wang, S. K. Gupta, and M. Pedram, "An Energy and Deadline Aware Resource Provisioning, Scheduling and Optimization Framework for Cloud Systems," in Proceedings of the Ninth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, Piscataway, NJ, USA, 2013, pp. 31:1–31:10.

8 L. Luo, W. Wu, W. T. Tsai, D. Di, and F. Zhang, "Simulation of power consumption of cloud data centers," Simul. Model. Pract. Theory, vol. 39, pp. 152–171, Dec. 2013.

9 Z. Liu, R. Ma, F. Zhou, Y. Yang, Z. Qi, and H. Guan, "Power-aware I/O-Intensive and CPU-Intensive applications hybrid deployment within virtualization environments," in 2010 IEEE International Conference on Progress in Informatics and Computing (PIC), 2010, vol. 1, pp. 509–513.

10 Lezama, A., Tchernykh, A., Yahyapour, R.: Performance Evaluation of Infrastructure as a Service Clouds with SLA Constraints. *Computación y Sistemas* 17(3): 401–411 (2013)

11 S. B. Matthias Splieth, "Analyzing the Effect of Load Distribution Algorithms on Energy Consumption of Servers in Cloud Data Centers," 2015.

12. Tchernykh, A., Lozano, L., Schwiegelshohn, U., Bouvry, P., Pecero, J. E., Nesmachnow, S.: Energy-Aware Online Scheduling: Ensuring Quality of Service for IaaS Clouds. International

Conference on High Performance Computing & Simulation (HPCS 2014), pp 911–918, Bologna, Italy (2014)

13. Tchernykh, A., Schwiegelsohn, U., Yahyapour, R., Kuzjurin, N.: Online Hierarchical Job Scheduling on Grids with Admissible Allocation, Journal of Scheduling 13(5):545–552 (2010)
14. Tchernykh, A., Ramírez, J., Avetisyan, A., Kuzjurin, N., Grushin, D.,. Zhuk, S.: Two Level Job-Scheduling Strategies for a Computational Grid. In R. Wyrzykowski et al. (eds.) *Parallel Processing and Applied Mathematics*, 6th International Conference on Parallel Processing and Applied Mathematics. Poznan, Poland, 2005, LNCS 3911, pp. 774–781, Springer-Verlag (2006)
15. Dorronsoro, B., Nesmachnow, S., Taheri, J., Zomaya, A., Talbi, E-G., Bouvry, P.: A hierarchical approach for energy-efficient scheduling of large workloads in multicore distributed systems. Sustainable Computing: Informatics and Systems 4:252–261 (2014)

16 Tchernykh, A., Pecero, J., Barrondo, A., Schaeffer, E.: Adaptive Energy Efficient Scheduling in Peer-to-Peer Desktop Grids, *Future Generation Computer Systems*, 36:209–220 (2014).

17 Ramírez, J.M., Tchernykh, A., Yahyapour, R., Schwiegelshohn, U., Quezada, A., González, J., Hirales, A.: Job Allocation Strategies with User Run Time Estimates for Online Scheduling in Hierarchical Grids. *Journal of Grid Computing* 9:95–116 (2011)

18 Iturriaga, S., Nesmachnow, S., Dorronsoro, B., Bouvry, P.: Energy efficient scheduling in heterogeneous systems with a parallel multiobjective local search. *Computing and Informatics* 32(2):273–294 (2013)

19 Schwiegelshohn, U., Tchernykh, A.: Online Scheduling for Cloud Computing and Different Service Levels, *26th Int. Parallel and Distributed Processing Symposium* Los Alamitos, CA, pp. 1067–1074 (2012)

20 Tchernykh, A., Lozano, L., Schwiegelshohn, U., Bouvry,P., Pecero, J., Nesmachnow, S., Drozdov, A. Online Bi-Objective Scheduling for IaaS Clouds with Ensuring Quality of Service. *Journal of Grid Computing*, Springer-Verlag, DOI 10.1007/s10723-015-9340-0 (2015)