Accepted Manuscript

Energy cost minimization with job security guarantee in Internet data center

Zhongjin Li, Jidong Ge, Chuanyi Li, Hongji Yang, Haiyang Hu, Bin Luo, Victor Chang

PII: S0167-739X(16)30763-4
DOI: http://dx.doi.org/10.1016/j.future.2016.12.017
Reference: FUTURE 3258

To appear in: *Future Generation Computer Systems*

Received date: 31 August 2016
Revised date: 10 December 2016
Accepted date: 12 December 2016

Please cite this article as: Z. Li, J. Ge, C. Li, H. Yang, H. Hu, B. Luo, V. Chang, Energy cost minimization with job security guarantee in Internet data center, *Future Generation Computer Systems* (2016), http://dx.doi.org/10.1016/j.future.2016.12.017

***Highlights (for review)**

1. The energy cost optimization architecture is proposed for IDC operator.
2. A heuristic algorithm is devised to select security services to guarantee the job security.
3. The temporal diversity of electricity price is considered in minimizing the energy cost.
4. The energy cost minimization algorithm is based on Lyapunov optimization technique.
5. Extensive evaluation experiments demonstrate the effectiveness of our algorithms.

Energy Cost Minimization with Job Security Guarantee in Internet Data Center

Zhongjin Li[a,b], Jidong Ge[a,b*],Chuanyi Li[a], Hongji Yang[c], Haiyang Hu[a,b,d] Bin Luo[a] and Victor Chang[e]

[a] *State Key Laboratory for Novel Software Technology, Software Institute, Nanjing University, Nanjing 210093, China*

[b] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876,China*

[c] *Centre for Creative Computing (CCC), Bath Spa University, England, UK*

[d] *School of Computer, Hangzhou Dianzi University, Hangzhou 310018, China*

[e] *International Business School Suzhou, Xi'an Jiaotong Liverpool University, Suzhou, China*

Abstract

With the proliferation of various big data applications and resource demand from Internet data centers (IDCs), the energy cost has been skyrocketing, and it attracts a great deal of attention and brings many energy optimization management issues. However, the security problem for a wide range of applications, which has been overlooked, is another critical concern and even ranked as the greatest challenge in IDC. In this paper, we propose an energy cost minimization (ECM) algorithm with job security guarantee for IDC in deregulated electricity markets. Randomly arriving jobs are routed to a FIFO queue, and a heuristic algorithm is devised to select security levels for guaranteeing job risk probability constraint. Then, the energy optimization problem is formulated by taking the temporal diversity of electricity price into account. Finally, an online energy cost minimization algorithm is designed to solve the problem by Lyapunov optimization framework which offers provable energy cost optimization and delay guarantee. This algorithm can aggressively and adaptively seize the timing of low electricity price to process workloads and defer delay-tolerant workloads execution when the price is high. Based on the real-life electricity price, simulation results prove the feasibility and effectiveness of proposed algorithm.

Keywords: Internet data center; security service; risk probability constraint; energy cost minimization; deregulated electricity markets

1. Introduction

Cloud computing, comprises of infrastructure platforms called Internet data center (IDC), is a large-scale distributed computing to meet the skyrocketing demand of big data applications and services. As an IDC typically consists of tens of thousands of servers, the energy consumption or energy cost is one of the critical problems. For example, many IDCs (e.g., Microsoft, Google, Akamai and INTEL) spend millions of dollars on electricity costs every year, which result in a large portion of operation expense [1, 2]. Hence, a considerable cost can be saved even reducing a few percent energy cost.

In cloud computing environment, the jobs or application requests from the cloud users can be submitted to IDC, which are also considered as virtual network (VN) requests [3, 4]. These jobs or applications may be delay tolerant big data, such as scientific computing and data intensive MapReduce applications [2]. Generally, jobs or applications are arrived at IDC randomly. Therefore, the scheduling problem, which is also can be considered as the energy cost problem, is key issue for IDC to ensure the QoS of jobs and reduce the energy overhead [5, 6].

Recently, a great attention has been paid to IDC energy management by both academia and industry. Extensive research has been developed to minimize the energy cost by utilizing the electricity price dynamics across geographically distributed regions [7, 8], and apply VM migration to achieve the

goals of saving energy [9, 10]. Especially, the electricity price manifests spatial and temporal diversity in the real life. For instance, in North America, owing to the different power generation profiles and electricity markets have been deregulated, the electricity prices are not constant but vary on the basis of an hour or 15-min [6]. To consider the temporal diversity of electricity price, the energy storage for energy cost saving is studied [11, 12], and both service delay and energy cost are taken into account in geographically distributed data center [2].

Security is another critical concern and even ranked as the greatest challenge in cloud computing environment. A survey from international data corporation shows that security is one of the greatest concerns in cloud computing [13]. Many works tackle the security problem on clusters [14], grid computing [15], heterogeneous distributed system [16, 17], cloud computing [18-21] and real-time embedded systems [22]. Unfortunately, because cloud computing environment is used to execute various applications of users, applications and users all may be the sources of malicious attack [23]. Furthermore, security mechanism is overlooked and has not been employed to counter any security threats [24, 25]. Therefore, it is necessary to deploy security services to protect various applications running in the IDC. However, security workload is incurred by adding security services to applications. Hence, it is a big challenge to tradeoff energy cost and service quality.

In this paper, we propose an energy cost minimization (ECM) algorithm with job security guarantee for IDCs where the electricity price exhibits temporal diversity. These jobs may be delay tolerant big data applications that take from several minutes to more than many hours. Our targets can be described as follows: 1) guaranteeing the risk probability constraint of each arriving job; 2) exploiting the temporal diversity of electricity price to minimize energy cost. First, a heuristic algorithm is devised to select security levels for workload shaping to guarantee the job security. Then, the energy optimization problem is formulated by taking the temporal diversity of electricity price into account. An online ECM algorithm, based on the Lyapunov optimization framework, is applied to solve that optimization problem. Our purpose is to minimize energy cost by deciding: 1) how to select security services to guarantee the risk probability constraint; 2) how many workloads should be processed in each time slot; and 3) how many resources should be provided by the IDC.

The main contributions of this paper can be summarized as follows:

- We propose, design and evaluate an energy cost optimization architecture, which mainly consists of Cloud Users, Job FIFO Queue, IDC Operator (includes Job Analyzer, Workload Shaping with Security Guarantee, Energy Cost Minimization and Server Management) and Servers (see Section 3.1).
- We devise a heuristic algorithm for arriving jobs to select appropriate security services to guarantee the job security. Based on it, the security workload shaping can be finished. In our architecture, the total workload consists of task execution workload and security workload (see Sections 3.5 and 4.1).
- We exploit the temporal diversity of electricity price to minimize the energy cost in deregulated electricity markets. The ECM algorithm is based on Lyapunov optimization technique which can facilitate energy cost versus delay trade-off for IDC operator (see Sections 3.6 and 4.2).
- Based on real-life electricity price data sets, the simulation results show that our approach can achieve energy cost saving and security guaranteeing simultaneously (see Section 5).

The rest of this paper is organized as follows. Section 2 summarizes the related work. In Section 3, we describe the system architecture, models and problem formulation. Section 4 introduces the algorithm design. The performance evaluation approaches and results, comparisons with similar work,

research contributions and limitations are conducted in Section 5. Conclusions and envisages our future work are given in Section 6.

2. Related Work

Security is one of the critical problems in distributed computing environment. However, most existing well-known scheduling studies neglect the security problems, and only few groups of researchers consider the security-driven scheduling policy for applications. Azzedin and Maheswaran [26] presented a trust brokering system which implicated the security meaning and was applied to the public resource grids. Song et al. [15] proposed six risk-resilient scheduling strategies for job security-assured under different risky conditions in grid environment. Xie and Qin [14] built three security overhead models for measuring execution time incurred by the security-critical tasks in clusters. Also the performance evaluations of security heterogeneity scheduling algorithm were studied in distributed computing systems [16]. Tang et al. [17] used the differential equation to build system node trust model and proposed a security-driven scheduling architecture for directed acyclic graph (DAG) applications. As for the workflow applications in cloud, Zeng et al. [18] introduced a security-aware and budget-aware (SABA) scheduling strategy to minimize the makespan with budget constraint. Then, Li et al. [27] proposed a security and cost aware scheduling (SCAS) algorithm for workflow application to optimize the execution cost with deadline and risk probability guarantee in clouds. Due to financial sector confronts the problems of inaccurate and inadequate assessment, Chang [28] deployed complex models in cloud to improve accuracy on risk analysis and prediction. The balance between benefits and risks should be considered for the projects of organization. Hence, based on organizational sustainability modeling (OSM) [29], Chang et al. [30] proposed a new technique, capital asset price modeling (CAPM), to evaluate the risks and benefits of commercial projects.

Energy consumption or energy cost problem of cloud data center has been attracted many attentions [21, 31-34]. Qureshi et al. [1] proved that electricity prices exhibit both temporal and spatial variations in deregulated electricity markets. According to the feature of electricity price, Rao et al. [7] proposed an energy cost minimization algorithm with guaranteeing quality of service under multiple electricity markets environment. Liu et al. [35] derived three distributed algorithms to achieve optimal geographical load balancing and also proved that geographical load balancing can significantly reduce brown energy use under special conditions. Shao et al. [6] used the mixed-integer nonlinear programming (MINLP) technique to achieve the optimal load balancing and energy cost management for IDCs. Luo et al. [36] proposed an energy cost optimization-IDC (eco-IDC) algorithm to minimize energy cost with service delay guarantee for data center.

In the light of risk preferences of IDC operators, Yu et al. [37] studied the problem of achieving the optimal tradeoff between operation risk and energy cost for IDC operators and proposed a risk-constrained decision framework to solve this problem. Sun et al. [38] proposed a power-efficient resource provisioning technique while meeting the service level agreements in cloud data center.

New aspects of power usage in data center have been emerged for energy cost reduction. Urgaonkar et al. [11] utilized energy storage devices to reduce the time average electric utility bill based on the Lyapunov optimization technique. Yu et al. [39] minimized energy cost by taking both workload and battery into consideration. Guo et al. [12] developed an online algorithm to minimize energy cost by integrating the center-level load balancing, the server-level configuration, and the battery management while satisfying the time guaranteeing of services. Yu et al. [40] investigated the problem of minimizing the energy cost with the uncertainties in electricity price, workload, renewable

energy generation, and power outage state. Liu et al. [41] integrated renewable supply, dynamic pricing, and cooling supply to reduce electricity cost, environmental impact and improve the overall sustainability of data center operations.

The Lyapunov optimization technique is first proposed in [42] for network stability problems. It was used to solve the energy optimal cross-layer control problems in time varying wireless networks [43]. Recently, the Lyapunov optimization technique has been widely utilized for wireless network, virtualized data center, social network, Internet data center, etc [44, 45]. Urgaonkar et al. [46] investigated optimal resource allocation and power management and employed Lyapunov optimization technique for job admission control, routing, and resource allocation in the virtualized data center. Do et al. [47] employed Lyapunov optimization technique to determine which social content should be send to mobile devices without requiring mobile users to be online all the time. Yao et al. [2] studied a stochastic optimization problem that takes job scheduling and server management into account, and a two-time-scale control algorithm based on Lyapunov optimization framework was proposed to reduce power cost.

The aforementioned studies focus on the security problem but ignore the energy consumption or energy cost, and others take the energy cost optimization problem into consideration and overlook the security of applications. Both energy problem and security problem are critical for IDC. Different from the above works, we investigate the energy cost minimization with job security guarantee for Internet data center in deregulated electricity markets.

3. System Architecture, Models and Problem Formulation

In this section, we model an IDC system and formulate an energy cost optimization problem. First, we describe system architecture, IDC resource and energy cost model, job arrival mode and security model. Then, we present the workload shaping with security guarantee and propose a security levels selection problem. Finally, a stochastic optimization problem is formulated to minimize the energy cost for the IDC. For ease of understanding, the major notations and their meanings used throughout of this paper are summarized in Table 1.

Table 1. Notations.

Symbol	Definition
M	The amount of servers in IDC
$R(t)$	The computing resource provided by the IDC in time slot t;
$f(t)$	The working frequency of server;
$P(\cdot)$	Power function of server;
sl_i	The set of security levels of task t_i;
$sl_i^{k(x)}$	The level of xth type of kth security service;
SL^k	The set of levels of kth security service;
λ^k	The risk coefficient of kth security service;
$n(t)$	The number of jobs arrival at IDC in time slot t;
$SW_i^{k(x)}$	Security workload of kth security service of task t_i;
SW_i	Total security workload of task t_i;
EW_i	The execution workload of task t_i;
W_i	The workload of task t_i;
$W(j)$	The workload of job j;

$W(t)$	The total workload of all arriving job in time slot t;
$P_{risk}(t_i, sl_i^{k(x)})$	The risk probability of the kth security service on task t_i;
$P_{risk}(t_i)$	The risk probability of task t_i;
$P_{risk}(j)$	The job risk probability;
λ	Average job arrival rate;
$C(t)$	Energy cost of IDC in time slot t;
$p(t)$	Electricity price in time slot t;
$Q(t)$	Workload queue backlog in time slot t;
$z(t)$	the amount of executed workloads in time slot t;
$L(Q(t))$	Lyapunov function;
$\Delta(Q(t))$	One time slot conditional Lyapunov drift;
V	Control parameter.

3.1 System Architecture

A similar architecture is proposed in [34]. However, it is not effectively incorporate the cloud security problems. Tang et al. [17] propose a security-driven scheduling architecture which does not take the energy management issues into consideration. The aim of our architecture is to minimize the energy cost under the job risk probability constraint for IDC. The proposed energy cost optimization architecture is depicted in Fig. 1. Four basically entities involved are introduced as follows:

1. **Cloud Users**: Submit applications or jobs from anywhere in the world to the IDC.

2. **Job FIFO Queue:** All arriving big data jobs are queued into this queue. Note that each job may contain many small tasks. In order to ensure the job security, each task should be executed with security services.

3. **IDC Operator**: Minimize the energy cost with job security guarantee in deregulated electricity markets:

a) *Job Analyzer*: Analyze the arriving big data applications including risk probability constraint, the number of tasks, the output and input data size of each task and so on.

b) *Workload Shaping with Security Guarantee*: Select appropriate security services to guarantee the job security. Then, the security workload shaping can be finished.

c) *Energy Cost Minimization:* Devise an online algorithm to minimize the energy cost by Lyapunov optimization framework. This algorithm comprehensively considers the workloads and stochastic electricity price.

d) *Server Manager*: Adjust working frequency of servers to minimize the power consumption according to the resource requirement.

4. **Servers**: The physical servers provide the hardware infrastructure to meet service demands.

The IDC Operator entry is the main component in our system architecture. It is responsible for analysis jobs, calculating workloads, devising algorithms, managing servers and so on. The operation process of IDC operator is shown in Fig. 2. In the beginning of each time slot, IDC operator receives jobs from cloud users and put them into the Job FIFO Queue. Immediately, the jobs analysis is conducted for new arriving jobs, which analyzes the security requirement, the number of tasks, workload of each task, etc. Then, the method of workload shaping with security guarantee is used to calculate the workload of each job, and the total workload of current time slot can be updated. Next, according to the workload, the energy cost minimization algorithm is implemented to minimize the energy cost based on the current electricity price. Finally, IDC Operator processes the jobs and

manages the servers (mainly adjusts the working frequency of servers) on the light of result of ECM algorithm. In the following sections, we will introduce models and problems related to the energy cost optimization architecture.

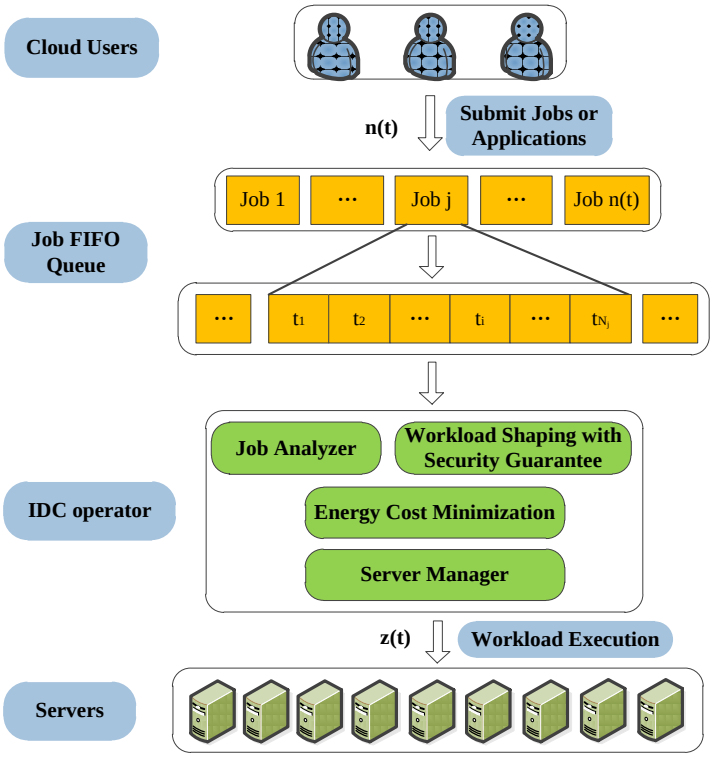


Fig. 1. The energy cost optimization architecture.

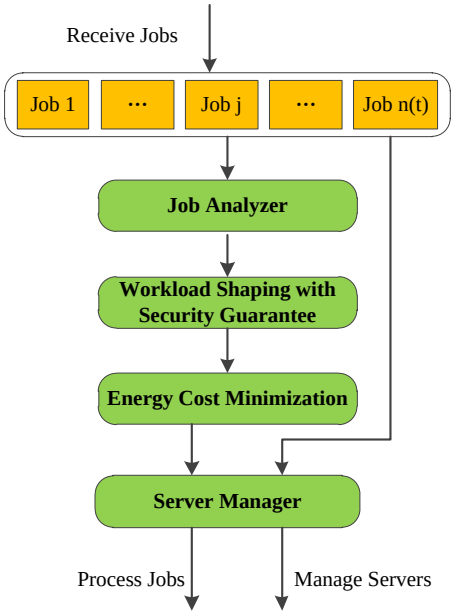


Fig. 2. The operation process of IDC operator

3.2 IDC Resource and Energy Cost model

Suppose the discrete-time system evolves over a sequence of equal-length time slots, i.e., $t = 0,1,2,...$. Let M is the number of homogeneous servers in IDC which provide resources measured in unit of *basic resource unit* [36]. A basic resource unit includes a series of CUP cores, a great sum of memory and so on. Thus, an IDC resource capacity is in unit of *basic resource unit· time slot*. The IDC needs to provide a certain amount of computing resource $R(t)$ according to the workload requirement. Suppose there exists $R_{\min}$ and $R_{\max}$ such that $R_{\min} \le R(t) \le R_{\max}$.

Modern CPUs can work at different voltage or frequency at runtime by using dynamic voltage and frequency scaling (DVFS) technique. The IDC operator can use this technique to modulate the current CPU speed which affects the computing resource and CPU power consumption [2, 46]. In time slot t, the server m is running in the working frequency $f_m(t)$, and then the computing resource of IDC can be written as

$$R(t) = \sum_{m=1}^{M} f_m(t) \tag{1}$$

Note that each server can run under different frequencies in the range of $[f_{\min}, f_{\max}]$ and $f_{\min} \ge 0$. Since existing CPUs only support discrete frequency levels, those supported frequencies in the range can be used.

The power requirement of resource capacity is denoted as $P(R(t))$. Symbol $P(\cdot)$ is the power function associated with resource capacity. The power function is known to IDC, and there exists a maximum value P_{max} such that $P(R(t)) \le P_{max}$ for all time slot t. Such power consumption will introduce some cost of the form "power × price". Moreover, the IDC may face the electricity price varying in deregulated electricity markets, and we denote price by $p(t)$ at time slot t. Price $p(t)$ is independent in every time slot t and takes a value in the finite state space. Then, the energy cost $C(t)$ of IDC in time slot t is computed by Eq. (2).

$$C(t) = P(R(t)) \cdot p(t) \tag{2}$$

Assume that p_{max} as the maximum electricity price that the IDC can experience. It is easy to see that if $C_{max} = P_{max} \cdot p_{max}$, then $C(t) \le C_{max}$ for all t.

Now, we discuss the assumption in our model. Above, in time slot t, suppose all severs have the same working frequency $f(t)$. Let us focus on single server and consider the following formulation: Let the power consumption of a server n with service frequency f_m be $P(f_m)$ [34, 48, 49].

$$P(f_m) = \alpha \cdot F(f_m) + \delta \tag{3}$$

where parameter δ denotes the constant power consumption, e.g., idle power consumption, $\alpha \cdot F(f_m)$ denotes the power consumption under operating frequency f_m, and α stands for the proportionality constant. Generally, function $F(f_m) = f_m^{\theta}$ is a convex function of frequency f_m, and $\theta = 3$ generally [34, 48, 49]. If the M servers run at different frequency $f_1, f_2, ..., f_M$, then the total power consumed can be written as Eq. (4).

$$P_{total} = \sum_{m=1}^{M} P(f_m) = \alpha \cdot \sum_{m=1}^{M} F(f_m) + \delta \cdot M \tag{4}$$

According to the Jensen's Inequality [44], we have

$$P_{total} = \alpha \cdot \sum_{n=1}^{M} F(f_m) + \delta \cdot M \ge \alpha \cdot M \cdot F(\frac{\sum_{m=1}^{M} f_m}{M}) + \delta \cdot M \tag{5}$$

This indicates that, to reduce the power consumption, all servers should have the same operating frequency [2]. Suppose all servers are running in the same frequency $f(t)$ in time slot t. Then, the Eq. (1) can be rewritten as

$$R(t) = M \cdot f(t) \tag{6}$$

This conclusion will be used to operate server frequency in our Server Management module.

3.3 Job Arrival Model

The arriving jobs that are big data applications from IDC users are queued into a job arrival queue. Generally, a job may include many small tasks as shown in Fig. 1. In fact, the tasks are processed by servers, not the jobs. In every time slot t, we denote the amount of newly arriving jobs as $n(t)$, and all jobs arrive at the end of each time slot. The variable $n(t)$ is the stochastic arrival with $\mathrm{E}\{n(t)\}=\lambda$, and it is assumed to be non-negative. Moreover, suppose that there exists a maximum N_{max} such that $n(t) \le N_{max}$ for all time slot t.

For an arriving job j, $j=1,2,...,n(t)$, let $P_{risk}^{c}(j)$, $T(j)$ and N_j represents the job risk probability constraint, tasks set, and the number of independent tasks respectively. The contents of security and the risk probability will be introduced in the Section 3.4 and Section 3.5. Moreover, a task $t_i \in T(j)$ can be represented by a tuple $t_i=(EW_i, D_i^{in}, D_i^{out})$. This tuple can be interpreted as follows: EW_i is the execution workload; D_i^{in} and D_i^{out} are corresponding the size of input data set and the size of output data set. Suppose that tasks belonging to a job or from different jobs may have the different execution workloads and the size of data set.

Generally, user requests or jobs can be generally classified as delay-sensitive, or delay-tolerant. In this paper, we focus on the jobs in big data delay-tolerant requests, which include compute-intensive or data-intensive jobs, such as scientific computing and data intensive MapReduce applications. For example, Google often has a large number of "long duration" jobs running on back-end servers [50]. These jobs take from several minutes to many hours and thus are relatively delay tolerant. As big data parallelizing applications keep growing in cloud computing environment, we assume that each job or request consists of a set of independent tasks, and a job is completed when all its tasks are finished [36, 39].

3.4 Security Model

Security service mechanisms have not been employed by many IDCs to counter the security threats of malicious users. There are three serious malicious attacks in cloud computing environments. Fortunately, authentication service, integrity service and confidentiality service can guard against these common threats respectively [14, 51]. Encryption mechanisms protect applications or data by enciphering methods. Meanwhile, integrity services ensure that no one can modify or tamper with data without being detected while they are executing. Then, authentication services pretend who intend to access content by malicious behaviors [14]. With these security services in place, the IDC operator can flexibly form an integrated security protection against a diversity of threats and attacks. Based on these above, a task execution process with security protection is show in Fig. 3 [27].

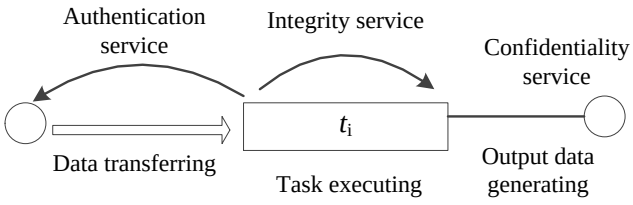


Fig. 3. A task execution process with security protection.

Three examples of authentication methods, hash functions for integrity and cryptographic

algorithm for confidentiality are shown in Table 2, Table3 and Table4 [14]. As a matter of convenience, we use letters *a*, *g* and *c* to represent the authentication, integrity and confidentiality respectively. It is noted that each task may require these three security services with various security levels. For example, sl_i is the set of security levels of task t_i provided by the IDC operator, which can be specified as a vector $sl_i = \{sl_i^{a(x)}, sl_i^{g(x)}, sl_i^{c(x)}\}$, where $sl_i^{a(x)}$ represents the security level of the xth type of authentication service, and the same explanation for $sl_i^{g(x)}$ and $sl_i^{c(x)}$.

As the performance of security services shown in Table 2, Table 3, Table 4, a security level is assigned to each security method in the range from 0 to 1, i.e., if no security service is used, the corresponding security level is 0, and when the security level 1 means that the security service is strongest yet slowest. According to different security services, in Table 2, we can see that the computation time is longer when the security level is larger, while in Table 3 and Table 4 it is shown that the processing rate is lower when the security level is higher. Nevertheless, these security services have the same principle that means the higher security level, the better security service, and the more execution time overhead. Note that the security level is inverse proportion to the time overhead of each algorithm.

Table 2. Authentication methods.

Authentication methods	sl^a : security level	The type of security service	Computation time: ms
No method used	0	0	-
HMAC-MD5	0.55	1	90
HMAC-SHA-1	0.91	2	148
CBC-MAC-AES	1	3	163

Table 3. Hash functions for integrity.

Hash functions	sl^g : security level	The type of security service	Processing rate: KB/ms
No function used	0	0	-
MD4	0.18	1	168.75
MD5	0.26	2	96.43
RIPEMD	0.36	3	37.50
RIPEMD-128	0.45	4	33.75
SHA-1	0.63	5	29.35
RIPEMD-160	0.77	6	21.09
Tiger	1.00	7	15.00

Table 4. Cryptographic algorithms for confidentiality.

Cryptographic algorithms	sl^c : security level	The type of security service	Processing rate: KB/ms
No algorithm used	0	0	-
SEAL	0.08	1	168.75
RC4	0.14	2	96.43
Blowfish	0.36	3	37.50
Knufu/Khafre	0.40	4	33.75
RC5	0.46	5	29.35
Rijndael	0.64	6	21.09
DES	0.90	7	15.00
IDEA	1.00	8	13.50

Recently, services and applications are moving their data to the cloud and centralize management and designed to reduce cost and increase operational efficiency. Moreover, security, trust, and privacy always remain challenges for organizations deployed in cloud computing. Then, Chang et al. [20] proposed multilayered security framework, cloud computing adoption framework (CCAF), for business clouds. This framework integrates three major security technologies, such as firewall, identity management, and encryption, and it can be adopted and successfully implemented in cloud services. Our security model is a representation similar to multi-layered system since three different security services need to be executed for a task. The difference in this paper is that it has elements of multi-layered security but it is a simplified version.

3.5 Workload Shaping with Security Guarantee

For each task of a job, it needs security services to ensure its successful execution. The security service also introduces some time overhead to the computing systems. The definitions of time overhead of kth security service can be found in detail in [14, 52]. The security overhead of integrity and confidentiality services mainly depends on the security service level and the size of dataset. Different from the time overhead of security services, we invert the security service into the security workload which is denoted as follows.

$$SW_i^{k(x)} = H^k(sl_i^{k(x)}, D_i^k),\ k \in \{g, c\} \tag{7}$$

where $k(x)$ is the xth type of kth security service, and $sl_i^{k(x)}$ is its security level; $SW_i^{k(x)}$ represents the security workload (in *basic resource unit*) of kth security service. $D_i^g = D_i^{in}$ and $D_i^c = D_i^{out}$ are the size of dataset to be protected by integrity service and confidentiality service respectively [27]. The function $H^k(\cdot,\cdot)$ can be induced from [14], and we can easily get the following property:

Property 1. *The function* $H^k(\cdot,\cdot)(k \in \{g, c\})$ *should satisfy the following conditions:*

- If $sl_i^{k(x)} = 0$ or $D_i^k = 0$, then $H^k(0, D_i^k) = H^k(sl_i^{k(x)}, 0) = 0$;
- If $sl_1^{k(x)} = sl_2^{k(x)}$ and $D_1^k < D_2^k$, then $H^k(sl_1^{k(x)}, D_1^k) < H^k(sl_2^{k(x)}, D_2^k)$;
- If $D_1^k = D_2^k$ and $sl_1^{k(x)} < sl_2^{k(x)}$, then $H^k(sl_1^{k(x)}, D_1^k) < H^k(sl_2^{k(x)}, D_2^k)$;

The three conditions reflect the security service workload associated with security levels and the protected data. However, as for authentication service, the security overhead is a constant and only depends on the security service type. Hence, the security workload of authentication service is computed by Eq. (8).

$$SW_i^{k(x)} = H^k(sl_i^{k(x)}),\ k \in \{a\} \tag{8}$$

The same property holds that $H^a(0) = 0$ and $H^a(sl_1^{a(x)}) < H^a(sl_2^{a(x)})$ when $sl_1^{a(x)} < sl_2^{a(x)}$. Then, the total security workload of task t_i is represented by formula (9).

$$SW_i = \sum_{k \in \{a,g,c\}} SW_i^{k(x)} \tag{9}$$

The workload of task t_i is denoted as follows.

$$W_i = EW_i + SW_i \tag{10}$$

where EW_i is the execution workload of task t_i. So, different from the existing works, the workload of a task includes two components. Then, the job workload is computed by Eq. (11).

$$W(j) = \sum_{i=1}^{N_j} W_i \tag{11}$$

Finally, the total arriving workloads of all job in time slot t is described as Eq. (12).

$$W(t) = \sum_{j=1}^{n(t)} W(j) \tag{12}$$

Note that different security levels may have different impact on the task security. Based on the security service introduced above, we quantitatively analyze the risk probability of task t_i with different security levels. The distribution of risk probability for fixed time interval follows a Poisson distribution. This is due to the fact that the coefficient of Poisson probability distribution can be seen as the arrival rate of malicious attacks. Thus, the risk probability of a task with single security service can be represented by an exponential distribution [16, 17]:

$$P_{risk}(t_i, sl_i^{k(x)}) = 1 - \exp(-\lambda^k (1 - sl_i^{k(x)})), k \in \{a, g, c\} \tag{13}$$

The risk coefficient λ^k may be different among three security services. For example, 3 snooping attacks, 2.5 alteration attacks and 1.8 spoofing attacks may be suffered by IDC in any time interval. The negative exponent indicates that risk probability grows with the difference $1 - sl_i^{k(x)}$. Hence, the risk probability of task t_i can be computed by integrating three security services, which is the Eq. (14).

$$P_{risk}(t_i) = 1 - \prod_{k \in \{a,g,c\}} (1 - P_{risk}(t_i, sl_i^{k(x)})) \tag{14}$$

Given a task set $T(j)$ of job j, the risk probability $P_{risk}(j)$ of job j is calculated based on Eq. (15).

$$P_{risk}(j) = 1 - \prod_{i=1}^{N_j} (1 - P_{risk}(t_i)) \tag{15}$$

In Section 3.3, each job has a risk probability constraint which is the server requirement of IDC user. As for IDC operator, the security workload should be minimized while guaranteeing the risk probability constraint. It is obvious that the less security workload, the lower energy consumption and energy cost. How to select security services for each task to ensure the job risk probability constraint is our first problem.

$$\text{Minimize:} \quad SW(j) = \sum_{i=1}^{N_j} SW_i \tag{16a}$$

$$\text{Subject to:} \quad P_{risk}(j) \le P_{risk}^c(j) \tag{16b}$$

$$sl_i^{k(x)} \in SL^k, k \in \{a, g, c\} \tag{16c}$$

Note that the levels of each security service are discrete. As for a task, there are $K = 4 \times 8 \times 9$ types of security service composition in the real-world applications (see Table 2, Table 3 and Table 4). Hence, the time complexity of this problem is $O(K^{N_j})$ which is exponential. Then, a heuristic algorithm, service levels selection algorithm, is devised to solve this problem. It has the polynomial time complexity and will be described in the Section 4.1.

3.6 Energy Cost Minimization Problem

Above, the workload shaping problem has been presented, and we can calculate all the new arriving jobs workload based on Eq. (12). In this section, we are interested in minimizing the long-term energy cost according to the current workloads in IDC, i.e., the expected energy cost averaged over the infinite time horizon, which is represented as follows.

$$\overline{C} = \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathrm{E}\{C(\tau)\} \tag{17}$$

The electricity price is changing in each time slot.

Let $Q(t), t = 0,1,...,$ be the workload queue in IDC which represents the queue backlog of workloads to be processed at the beginning of every time slot t. Generally, if the IDC processes all the workloads in the queue $Q(t)$ in spite of the price, it will incur high energy cost but low service delay. On the contrary, if the IDC executes the workloads only when the electricity price is low, the queue length $Q(t)$ will increase rapidly. Hence, there is a *cost-delay tradeoff* in conducting the workload execution. However, the workload queue should be stable in the time average sense, i.e.,

$$\overline{Q} = \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathrm{E}\{Q(\tau)\} < \infty \tag{18}$$

where $\overline{Q}$ represents the time-average workload backlog. Eq. (18) implies that all arriving jobs in IDC will be processed in bounded time, and a larger value $\overline{Q}$ means a longer delay for applications [44]. Then, the dynamics of workload queueing in each time slot can be represented by Eq. (19).

$$Q(t+1) = max[Q(t) - z(t), 0] + W(t) \tag{19}$$

where $z(t)$ represents the amount of workloads executed by IDC in time slot t and $Z_{max} \geq z(t), \forall t$ denotes the maximum workloads can be served in any time slot. Hence, in each time slot t, the IDC operator should make an online decision to minimize the energy cost under queue stability constraints for all jobs:

$$\text{Minimize:} \quad \overline{C} = \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\{C(\tau)\} \tag{20a}$$

$$\text{Subject to:} \quad \overline{Q} < \infty \tag{20b}$$

It is impractical to solve this stochastic optimization problem directly as it would require a-priori knowledge of the job arrival and electricity prices in advance. In the Section 4.2, we will present an online operation algorithm based on Lyapunov optimization framework to solve the energy cost minimization problem.

4. Algorithm Design

Minimizing the security workload under the security guarantee means that IDC only needs to provide less resource and can save energy cost. The time complexity of finding all the security service compositions is exponential. A heuristic algorithm with lower time complexity is needed. Moreover, the electricity price of IDC is highly stochastic and unpredictable, we can only use the current information (i.e. workload queue backlog, current electricity price) to make an online operation decision. First, a service levels selection algorithm based on heuristic approach is devised to minimize the security workload with job risk probability constraint. Then, we design an energy cost minimization algorithm along with workload queue stability based on the Lyapunov optimization framework [43, 44]. Moreover, a performance analysis is given for our ECM algorithm that can offer provable energy cost and delay guarantees.

4.1 Service Levels Selection Algorithm

Let us discuss the maximum risk probability of a job. If all tasks of a job are not serviced by any security services, i.e., $sl_i^{k(x)} = 0, k \in \{a, g, c\}, i = 1,2,...,N_j$, the job j may suffer the maximum risk. Note that $\lambda^k > 0$ and $0 \leq sl_i^{k(x)} \leq 1$ ($k \in \{a, g, c\}, i = 1,2,...,N_j$). Hence, task's risk probability of the kth security service is in the range [0, 1) based on Eq. (13). The same logic and range applies to task risk probability and job risk probability according to Eqs. (14) and (15) respectively. Rearranging terms of

Eq. (15) and taking the logarithm, we have

$$\begin{aligned} 1-P_{risk}(j) &= \prod_{i=1}^{N_j}(1-P_{risk}(t_i)) \\ \Rightarrow \log(1-P_{risk}(j)) &= \sum_{i=1}^{N_j}\log(1-P_{risk}(t_i)) \end{aligned} \tag{21}$$

Applying $1-P_{risk}(t_i)=\prod_{k\in\{a,g,c\}}(1-P_{risk}(t_i,sl_i^{k(x)}))$ yields:

$$\log(1-P_{risk}(j)) = \sum_{i=1}^{N_j}\sum_{k\in\{a,g,c\}}\log(1-P_{risk}(t_i,sl_i^{k(x)})) \tag{22}$$

Similarly, taking $1-P_{risk}(t_i,sl_i^{k(x)})=\exp(-\lambda^k(1-sl_i^{k(x)}))$ into Eq. (22), we get

$$\log(1-P_{risk}(j)) = \sum_{i=1}^{N_j}\sum_{k\in\{a,g,c\}}(-\lambda^k(1-sl_i^{k(x)})) \tag{23}$$

When security level $sl_i^{k(x)}=0, k\in\{a,g,c\}, i=1,2,...,N_j$ the job j will have the maximum risk probability $P_{risk}^{\max}(j)$. Then,

$$P_{risk}^{\max}(j)=1-\exp(-N_j(\lambda^a+\lambda^g+\lambda^c)) \tag{24}$$

We can see that the maximum risk probability of a job is only related to risk coefficients. If the IDC never suffer any attacks, i.e., $\lambda^a=\lambda^g=\lambda^c=0$, the risk probability is 0, and we need not any security services. However, the maximum risk probability will be higher with more malicious attacks. In this paper, we assume that $\lambda^k>0, k\in\{a,g,c\}$ and hence the maximum risk probability of a job $P_{risk}^{\max}(j)<1$. Moreover, the risk probability constraint self-defined by user should be equal or less than the maximum risk probability, that is $P_{risk}^{c}(j)\le P_{risk}^{\max}(j), j=1,2,...,N_j$. So, it is necessary to apply security services to protect jobs execution.

Next, we introduce a heuristic algorithm, the security levels selection algorithm, for each job to minimize the security workload while satisfying the job risk probability constraint. The pseudo code of the algorithm is outlined in Fig. 4. It is difficult to solve the security levels selection problem directly based on the Eqs. (16a)-(16c). According to Eq. (23), we transform constraint Eq. (16b) as below.

$$\log(1/(1-P_{risk}^{c}(j))) \ge \sum_{i=1}^{N_j}\lambda^a(1-sl_i^{a(x)})+\sum_{i=1}^{N_j}\lambda^g(1-sl_i^{g(x)})+\sum_{i=1}^{N_j}\lambda^c(1-sl_i^{c(x)}) \tag{25}$$

Then, the Eqs. (16a)-(16c) can be rewritten as follows.

$$\text{Minimize:}\quad SW(j)=\sum_{i=1}^{N_j}(SW_i^{a(x)}+SW_i^{g(x)}+SW_i^{c(x)}) \tag{26a}$$

$$\text{Subject to: (25), (16c)} \tag{26b}$$

We can see that one term $\lambda^k(1-sl_i^{k(x)})$ correspond to a security workload $SW_i^{k(x)}$. Then, the ratio of security workload and security level that means the growth with security workload $SW_i^{k(x)}$ and the term $\lambda^k(1-sl_i^{k(x)})$ is defined as follows.

$$u(t_i,k(x))=\frac{SW_i^{k(x)}}{\lambda^k(1-sl_i^{k(x)})+\eta}, k\in\{a,g,c\} \tag{27}$$

where η is small positive constant and the reason of adding η in denominator is to prevent the case of zero divided. It can be deduced from Eq. (27) that the lower ratio, the less security workload under this security level.

Assume that all tasks in job j are firstly mapped to the lowest security levels, i.e., $map: t_i \to sl_i^{k(x)}=0, k\in\{a,g,c\}, i=1,2,...,N_j$. Then, we have the minimum value of Eq. (11), i.e., $SW(j)=0$. Let sum as the value of right-hand-side (R.H.S) of Eq. (25) which is denoted by Eq. (28):

$$sum=\sum_{i=1}^{N_j}\lambda^a(1-sl_i^{a(x)})+\sum_{i=1}^{N_j}\lambda^g(1-sl_i^{g(x)})+\sum_{i=1}^{N_j}\lambda^c(1-sl_i^{c(x)}) \tag{28}$$

When all tasks are mapped to the lowest security levels, we have maximum value $sum^{\max}$ and maximum job risk probability $P_{risk}^{\max}(j)$. Hence, the constraint of Eq. (25) is not satisfied because of the fact $P_{risk}^{c}(j) \le P_{risk}^{\max}(j)$. There are 3 types of authentication services, 7 types of integrity services and 8 types of confidentiality services except security services of lowest levels. So, all the ratios of each security service on task t_i can be listed as below.

$$\begin{cases} U(t_i,a) = \{u(t_i,a(1)), u(t_i,a(2)), u(t_i,a(3))\} \\ U(t_i,g) = \{u(t_i,g(1)), u(t_i,g(2)), u(t_i,g(3)), u(t_i,g(4)), u(t_i,g(5)), u(t_i,g(6)), u(t_i,g(7))\} \\ U(t_i,c) = \{u(t_i,c(1)), u(t_i,c(2)), u(t_i,c(3)), u(t_i,c(4)), u(t_i,c(5)), u(t_i,c(6)), u(t_i,c(7)), u(t_i,c(8))\} \end{cases} \quad (29)$$

We denote all the ratios of task t_i as $U(t_i) = \{U(t_i,a), U(t_i,g), U(t_i,c)\}$. Then, the ratios of job j can be represented as set $U(j) = \{U(t_1), U(t_2), ..., U(t_{N_j})\}$, and the number of terms in this set is $K \cdot N_j$, where $K = 3 \times 7 \times 8$. Next, the implementation steps of security levels selection algorithm are presented as below.

Step a: We first map all tasks to the lowest security levels, i.e. $map: t_i \to sl_i^{k(x)} = 0, k \in \{a,g,c\}, i = 1,2,...,N_j$. Hence, $sum = sum^{\max}$ and the constraint Eq. (25) is not satisfied in terms of above analysis (line 1).

Step b: Then, compute all ratios of job j according to Eqs. (27) and (29), and Sort the all terms in set $U(j)$ by their value in ascending order. The lower ratio of a term, the less security workload will be incurred (line 2-3).

Step c: Take the first term $u^1(t_i,k(x))$ in the set which has the minimum ratio, and recalculate the value of sum when task t_i mapping to $k(x)$th security service (line 5).

Step c-1: If $sum \le \log(1/(1-P_{risk}^{c}(j)))$, that means inequality (25) is satisfied. Moreover, the security workload of job j is minimum (line 6-7).

Step c-2: Otherwise, the Eq. (25) is not satisfied. We need record the current mapping scheme and update the job security workload. By removing the first term $u^1(t_i,k(x))$ from the set $U(j)$, the sub-minimum ratio becomes the minimum ratio in the set $U(j)$ (line 8-11).

Step d: Repeating the above *Steps c*, *c-1* and *c-2* until the set $U(j)$ is empty (line 4-13) or inequality (25) is satisfied (line 6-7).

Algorithm 1: Security levels selection algorithm

BEGIN
01. Initialize the mapping scheme, i.e., $map: t_i \to sl_i^{k(x)} = 0, k \in \{a,g,c\}, i = 1,2,...,N_j$, and set $sum = sum^{\max}$, $W(j) = 0$;
02. Calculate all the ratios of job j based on Eqs. (27) and (29), and put them into the job security level utility set $U(j)$;
03. Sort the all terms of set $U(j)$ by their value in ascending order;
04. **while** set $U(j)$ is not empty
05. Take the first term $u^1(t_i,k(x))$ in the set which has the minimum ratio, and recalculate the value of sum when task t_i mapping to $k(x)$th security service.
06. **if** $sum \le \log(1/(1-P_{risk}^{c}(j)))$
07. Exit the while loop;
08. **else** $sum > \log(1/(1-P_{risk}^{c}(j)))$
09. Record the current mapping scheme, i.e., $map: t_i \to sl_i^{k(x)}$;
10. Update the job security workload $SW(j)$;
11. Remove the first term $u^1(t_i,k(x))$ from the set $U(j)$;

12. **end if**
13. **end while**
END

Fig. 4. The pseudo code of security levels selection algorithm.

There is all always a solution in our security levels selection algorithm because of $0 \le P_{risk}^{c}(j) \le P_{risk}^{\max}(j)$. A extreme mapping scheme is that all tasks select the maximum levels of security service, i.e. $sum = 0$ in this case. The time complexity of calculating the all terms in set $U(j)$ is $O(K \cdot N_j)$. The worst time complexity of sorting the all terms in set $U(j)$ and while loop is $O(K \cdot N_j \log(K \cdot N_j))$ and $O(K \cdot N_j)$, respectively. As a result, the time complexity of security levels selection algorithm is polynomial $O(K \cdot N_j \log(K \cdot N_j))$. By using security levels selection algorithm for each job, the total workload of new arriving jobs can be computed by Eq. (12).

An example of the security workload shaping process with security levels selection is illustrated in Fig. 5. First, the security levels selection algorithm is applied for each arriving job. It can map a security service composition to every task. Then, the security workload of a task can be calculated according to the security levels mapping scheme.

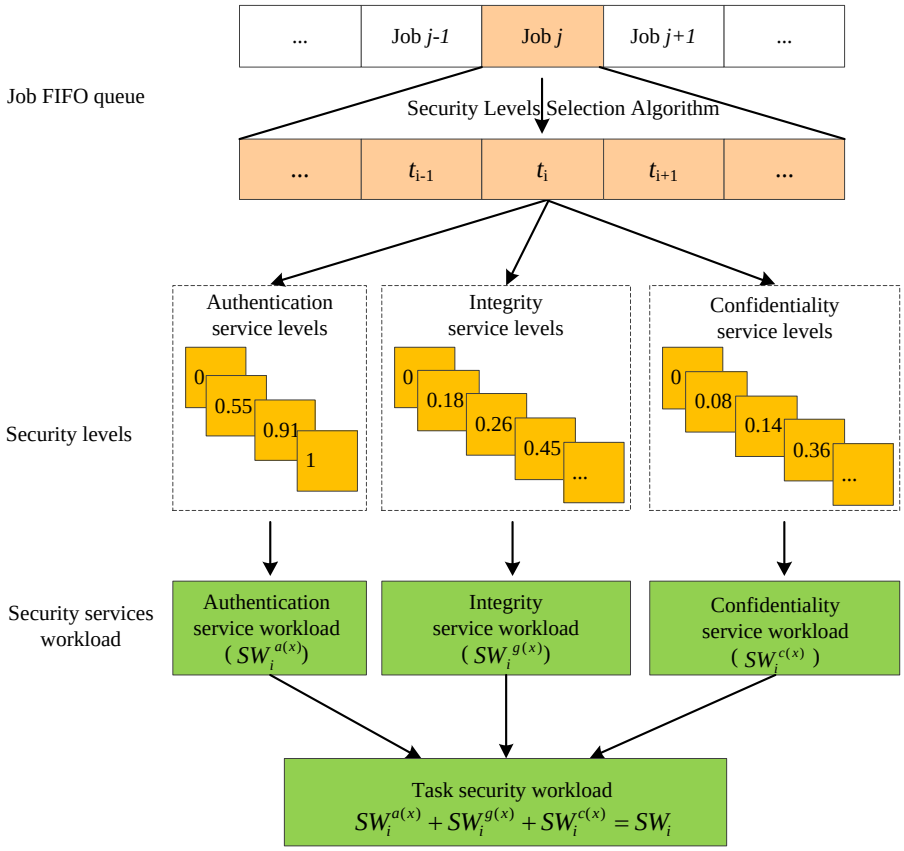


Fig. 5. Security workload shaping process.

4.2 Energy Cost Minimization Algorithm

The total workload of all arriving big data jobs can be obtained by workload shaping with security levels selection algorithm. In this section, we introduce an online energy cost minimization algorithm based on Lyapunov optimization framework [43, 44]. It is an online decision algorithm and can offer

provable energy cost and delay guarantees. Moreover, this algorithm does not require any a-priori knowledge, and it only takes the workload queue in IDC and the temporal diversity of electricity price into account.

To solve the problem (20a)-(20b), we define the Lyapunov function, $L(Q(t))$, which represents a scalar metric of workload queue backlog for reflecting delays of jobs, as follows:

$$L(Q(t)) = \frac{1}{2}Q(t)^2 \tag{30}$$

where $Q(t)$ is the workload queue which can evolve over slot $t \in \{0,1,2,...\}$, and $L(Q(t)) \geq 0, \forall t$. The value of $L(Q(t))$ reflects the backlog of workload queue, such as a larger value of $L(Q(t))$ means that the queue is congested; on the contrary, the queue is idle. The Lyapunov drift $\Delta(Q(t))$ is used to keep the system stable by pushing the Lyapunov function towards a lower congestion state, which is represented as follows:

$$\Delta(Q(t)) = \mathrm{E}\{L(Q(t+1)) - L(Q(t)) \mid Q(t)\} \tag{31}$$

Following the Lyapunov optimization approach [44], we put the energy cost of one time slot to both sides of Eq. (31), which leads to the *drift-plus-penalty* term: $\Delta(Q(t)) + V \cdot \mathrm{E}\{C(t) \mid Q(t)\}$, where control parameter $V > 0$ that represents an important weight on how much the IDC operator emphasizes energy cost. Such a control decision can be motivated as follows: we want to make $\Delta(Q(t))$ small to push queue backlog towards a lower congestion state, but we also want to make $\mathrm{E}\{C(t) \mid Q(t)\}$ small so that we do not incur large energy cost expenditure. Then, the following lemma defines an upper bound.

Lemma 1. For any possible actions under constraint (20b) that can be implemented at slot t, we have

$$\begin{aligned} \Delta(Q(t)) + V \cdot \mathrm{E}\{C(t) \mid Q(t)\} \leq B + V \cdot \mathrm{E}\{C(t) \mid Q(t)\} \\ + Q(t) \cdot \mathrm{E}\{W(t) - z(t) \mid Q(t)\} \end{aligned} \tag{32}$$

where $B = (W_{max}^2 + Z_{max}^2)/2$, and $W_{\max} \geq W(t), \forall t$ represents the maximum amount of workload can arrive per time slot and $Z_{\max} \geq z(t), \forall t$ represents the maximum amount of workload that can be executed in a time slot.

Proof. According to Eq. (30), we have

$$L(Q(t+1)) - L(Q(t)) = \frac{1}{2}[Q(t+1)^2 - Q(t)^2] \tag{33}$$

Then, using the fact that for any real number x, $(max[x,0])^2 \leq x^2$, we have

$$Q(t+1)^2 - Q(t)^2 \leq W(t)^2 + z(t)^2 + 2Q(t) \cdot [W(t) - z(t)] \tag{34}$$

Then,

$$\begin{aligned} \Delta(Q(t)) &= \mathrm{E}\{L(Q(t+1)) - L(Q(t)) \mid Q(t)\} \\ &\leq \frac{1}{2}\mathrm{E}\{[W(t)^2 + z(t)^2] \mid Q(t)\} + Q(t) \cdot \mathrm{E}\{[W(t) - z(t)] \mid Q(t)\} \end{aligned} \tag{35}$$

As $W(t) \leq W_{max}$ and $z(t) \leq Z_{max}$, we have

$$\frac{1}{2}\mathrm{E}\{[W(t)^2 + z(t)^2] \mid Q(t)\} \leq \frac{1}{2}(W_{max}^2 + Z_{max}^2) \tag{36}$$

Then, we get

$$\Delta(Q(t)) \leq B + Q(t) \cdot \mathrm{E}\{W(t) - z(t) \mid Q(t)\} \tag{37}$$

Now adding to both sides the energy cost over the frame, i.e., the term $V \cdot \mathrm{E}\{C(t) \mid Q(t)\}$ prove the lemma 1.

□

Following the design principle of Lyapunov framework, the underlying objective is to minimize the upper bound of the *drift-plus-penalty* term. Rather than directly minimize *drift-plus-penalty* term every slot t, our strategy actually seeks to minimize the bound given in the right-hand-side (R.H.S) of Eq. (32). This is done via the framework of opportunistically minimizing a conditional expectation. Then, our algorithm finally minimizes the R.H.S of Eq. (32) by minimizing the following simplified term:

$$\text{Minimize} \qquad V \cdot C(t) - Q(t) \cdot z(t) = V \cdot P(R(t)) \cdot p(t) - Q(t) \cdot z(t) \tag{38}$$

Note that the amount of resource provided by IDC is equal to the amount of workload processed in time slot t, that is $R(t) = z(t)$. Furthermore, based on the conclusion discussed in Section 3.2, all servers are running in the same frequency $f(t)$ in time slot t. Then, we have $R(t) = z(t) = M \cdot f(t)$ and

$$P(R(t)) = P_{total} = \alpha \cdot M \cdot F(f(t)) + \delta \cdot M \tag{39}$$

where $F(f(t)) = f^{\theta}(t)$ and $\theta = 3$ is a constant. Then, the problem of minimizing Eq. (38) can be rewritten as follows.

$$\text{Minimize: } V \cdot M \cdot (\alpha \cdot f^{\theta}(t) + \delta) \cdot p(t) - M \cdot Q(t) \cdot f(t) \tag{40a}$$

$$\text{Subject to: } f(t) \in [f_{\min}, f_{\max}] \tag{40b}$$

We define $y(f(t)) = V \cdot M \cdot (\alpha \cdot f^{\theta}(t) + \delta) \cdot p(t) - M \cdot Q(t) \cdot f(t)$. As $Q(t)$ and $p(t)$ can be observed at the beginning of time slot t, there are only one variable $f(t)$ in function $y(f(t))$. Servers in IDC only support discrete frequency levels, and only those supported frequencies in the range can be used. However, we consider $y(f(t))$ is the continuous function that will not affect the optimal solution. The first derivative of function $y(f(t))$ is

$$y'(f(t)) = \alpha \cdot \theta \cdot V \cdot M \cdot f^{\theta-1}(t) \cdot p(t) - M \cdot Q(t) \tag{41}$$

and the second derivative is

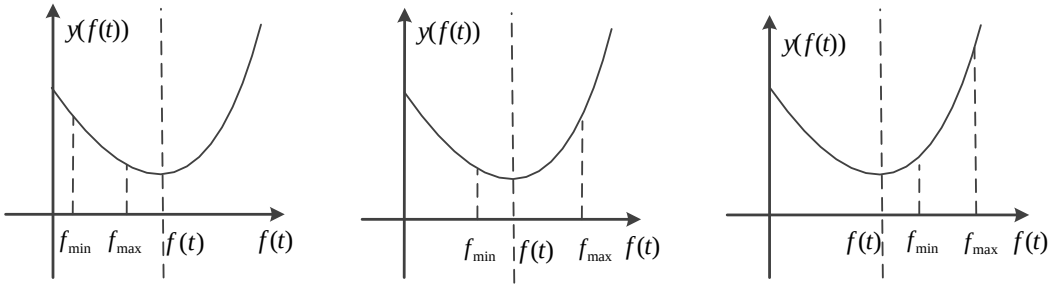
$$y''(f(t)) = \alpha \cdot \theta \cdot (\theta - 1) \cdot V \cdot M \cdot f^{\theta-2}(t) \cdot p(t) \tag{42}$$

Because $\theta = 3$ and $f(t) \in [f_{\min}, f_{\max}]$, the second derivative is nonnegative, i.e. $y''(f(t)) > 0$. So, the minimum point can be computed by taking the first derivative equals zero, which is

$$f(t) = \sqrt[(\theta-1)]{\frac{M \cdot Q(t)}{\alpha \cdot \theta \cdot V \cdot M \cdot p(t)}} \tag{43}$$

It is note that this minimum point is the unique extreme point in the domain $(0, \infty)$. Moreover, $y'(0) < 0$ and $y'(f(t) + \eta) > 0$ where η is a positive. So, the function $y(f(t))$ is monotone increasing in the domain $(0, f(t))$ and is monotone decreasing in the range $(f(t), \infty)$, which are shown in Fig. 6. Then, we discuss the optimal solution as follows:

- If $f(t) > f_{\max}$ (see Fig. 6(a)), which means the workload queue backlog may be too larger or the electricity price may be is too lower in time slot t. Hence, the IDC operator intends to execute as many workloads as possible to reduce the workload queue backlog and minimize the cost. Finally, the optimal frequency of all servers is $f^{opt}(t) = f_{\max}$.
- If $f_{\min} \le f(t) \le f_{\max}$ (see Fig. 6(b)), the IDC operator will process the proper amount of workload according to the current workload queue backlog and electricity price. Note that servers can only operate under discrete frequency levels in the range of $[f_{\min}, f_{\max}]$ by DVFS technique. Suppose there exits two frequency f_1 and f_2 ($f_1 < f_2$), and $f_1 \le f(t) \le f_2$. Hence, $f^{opt}(t) = f_1$ if $y(f_1) < y(f_2)$; otherwise, $f^{opt}(t) = f_2$.
- If $f(t) < f_{\min}$ (see Fig. 6(c)), which means the workload queue backlog may be too less and the electricity price may be too higher. The IDC operator will process as few workloads as possible to save the energy cost. So, the optimal working frequency is $f^{opt}(t) = f_{\min}$.



(a) $f(t) > f_{max}$ (b) $f_{min} \le f(t) \le f_{max}$ (c) $f(t) < f_{min}$

Fig. 6. The monotonicity of function $y(f(t))$.

Based on the above analysis, we summarize that our ECM algorithm intend to process workloads in the following conditions: 1) when the electricity price $p(t)$ is low enough, the IDC operator will catch the chance to execute more workloads; 2) when the queue $Q(t)$ is congested, workloads must be finished to guarantee the queue stability. The pseudo code of ECM algorithm is outlined in Fig. 7. First, the IDC operator observes the workload queue backlog and the real-time electricity price at the beginning of each time slot t (line 1). Then, the optimal working frequency of all servers can be got (line 2-9). Finally, IDC operator calculates the optimal amount of workloads will be processed in time slot t (line 10), and update workload queue (line 11). The time complexity of ECM algorithm is constant that is suitable for IDC operator to make online decision to minimize the energy cost.

Algorithm 2: ECM algorithm

BEGIN

01. At the beginning of each time slot t, monitor the workload queue backlog $Q(t)$ of IDC and the real-time electricity price $p(t)$;

02. Get the first derivative of function $y(f(t))$ according to Eq. (41), and compute the minimum point based on Eq. (43);

03. **if** $f(t) > f_{max}$

04. Set $f^{opt}(t) = f_{max}$;

05. **else if** $f_{min} \le f(t) \le f_{max}$

06. Set $f^{opt}(t) = f_1$ if $y(f_1) < y(f_2)$; otherwise, set $f^{opt}(t) = f_2$;

07. **else if** $f(t) < f_{min}$

08. Set $f^{opt}(t) = f_{min}$;

09. **end if**

10. Calculate $z(t) = M \cdot f^{opt}(t)$ which is the optimal amount of workloads that will be processed in time slot t;

11. Update workload queue $Q(t+1)$ when the current time slot t ends according to the dynamics Eq. (19).

END

Fig. 7. The pseudo code of ECM algorithm.

The performance bounds of ECM algorithm are stated in the following theorem.

Theorem 1. Assume that the job arrival rate λ is strictly within the network capacity region $\mathbf{\Lambda}$, and the ECM algorithm is applied at each time slot t. For any control parameter $V > 0$, it generates the

time-average energy cost $\overline{C}$ and queue backlog $\overline{Q}$ satisfying that:

$$\overline{C} = \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\{C(\tau)\} \le C^* + \frac{B}{V} \tag{44}$$

$$\overline{Q} = \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\{Q(\tau)\} \le \frac{B + VC^*}{\varepsilon} \tag{45}$$

where B and ε are positive constants, and C^* is the theoretical optimal time-average energy cost. Proof (pp. 47, [44])

Theorem 1 can be understood as follows: for any parameter $V > 0$, if we can use the ECM algorithm to ensure the drift condition is satisfied on every time slot, then the time average expected penalty satisfied Eq. (44) and hence is either less than the target value C^*, or differs from C^* by no more than the value D/V, which can be made arbitrarily small as V is increased. However, the time average queue backlog bound increases linearly in the V parameter, as shown by Eq. (45). This presents a cost-backlog tradeoff of $[O(1/V), O(V)]$. Such a cost-delay tradeoff allows ECM algorithm to make flexible design choices according to different application types and user contexts.

5. Performance Evaluation

In this section, we evaluate the performance of the proposed algorithm based on real-world electricity prices. First, we describe the experimental setup for performance evaluation. Next, we assess the total energy cost and delay of four reference algorithms and our ECM algorithm. Then we show the performance impact of parameter V on five algorithms. After that, we appraise the proposed ECM algorithm on the basis of risk probability constraint varying and evaluate the performance impact of three security services. Finally, comparisons with similar work and a brief introduction of research contributions and limitations are given.

5.1 Experimental Setup

The goal of this experimental study is to evaluate the performance of the proposed ECM algorithm. We describe the main components of our simulations: system parameters, job arrival and security parameters, electricity price and four algorithms in comparison.

5.1.1 System parameters

We consider an IDC having $M = 10000$ servers and we model power function as

$$P(f(t)) = M \cdot (\alpha \cdot f^3(t) + \delta) \tag{46}$$

where α and δ are constants determined by the IDC. Specifically, δ is the average idle power consumption of a server, and $\alpha \cdot f^3(t) + \delta$ gives the power consumption of a server running at computing frequency $f(t)$. Assume that the computing frequency is in the range [1.2, 3.2] [49]. Then, the minimum and maximum resource capacity of the IDC are $R_{\min} = 12000$ and $R_{\max} = 32000$ (in *basic resource unit*) respectively. Moreover, set $\alpha = 6.1$ and $P_{idle} = 100\text{W}$ such that the peak power consumed by a server is 250W. The model Eq. (46) is based on the measurements reported in [2, 48, 49, 53].

5.1.2 Job arrival and security parameters

Suppose the number of arriving jobs that are big data applications in each slot $n(t)$ follows a *Poisson* distribution with parameter 5, and the number of tasks per job follows the uniform distribution in the range [10, 100]. Furthermore, the execution workload per task is uniformly distributed in the range [10, 100] (in *basic resource unit*).

In order to ensure the security of each task, the IDC should process the security workload. The risk coefficients in our experiments are set $\lambda^a = 3.0$, $\lambda^g = 2.5$ and $\lambda^c = 1.8$, respectively. Then, based on the three risk coefficients and the number of task distribution, the maximum risk probability is approximately equal to 1. So, suppose the constraint of risk probability for each job follows the uniform distribution in the range [0, 1].

For the integrity service and confidentiality service, the security workload function (in *basic resource unit*) is devised as follows.

$$H^k(sl^{k(x)}, D^k) = \beta^k \cdot sl^{k(x)} \cdot D^k, \quad k \in \{g, c\} \tag{47}$$

We can see that Eq. (47) satisfies the property 1. As for authentication service, the workload function is calculated by Eq. (48).

$$H^k(sl^{k(x)}) = \beta^k \cdot sl^{k(x)}, \quad k \in \{a\} \tag{48}$$

For each task, the protected data D^k is in the range [0.1, 1] GB, and $\beta^a = 1600$, $\beta^g = 2400$ and $\beta^c = 800$ [[14], Fig. 3].

5.1.3 Electricity price

We downloaded the hourly electricity prices of Palo Alto, which is the Google's data centers host, in real-time electricity market [54], and the time horizon used in this paper is from June 1 to June 30, 2015. To fully exploit the temporal electricity price, we would like to be aware of prices at a time granularity that is set to 5 minutes in this paper [1]. Because the electricity price is varying on hourly, the interpolation method is used to generate prices at 5-minute intervals [2]. Thus, the time horizon in this simulation experiments s is 8640 slots.

5.1.4 Algorithms in comparison

The following four algorithms are compared in terms of energy cost and queuing delay in the experiments:

***Algo*-1:** This algorithm does not employ the proposed Lyapunov optimization technique. Thus, arriving jobs are not queued, and it starts to execute arriving jobs once they are received, that is $W(t) = z(t)$. Moreover, these jobs are executed without security services, i.e., $sl_i^{a(x)} = sl_i^{g(x)} = sl_i^{c(x)} = 0$ for each task.

***Algo-2*:** This algorithm starts to execute all arriving jobs once they are received. Thus, the IDC is without job queue that is $W(t) = z(t)$ in every time slot t. However, each job requires security services to ensure its risk probability constraint, and security levels selection algorithm which is based on heuristic method is used in this algorithm.

***Algo-3*:** It uses our proposed ECM algorithm without security services, i.e., $sl_i^{a(x)} = sl_i^{g(x)} = sl_i^{c(x)} = 0$ for each task. Different from *Algo-1* and *Algo-2*, the arriving tasks are queued in the IDC, which will be processed when the electricity price is low or the queue is congested.

***Algo-4*:** This algorithm applies the Lyapunov optimization framework to minimize the energy cost. Moreover, it ensures the risk probability constraint for jobs by using enumeration method to select security levels for tasks. We have discussed in Section 3.5 that the enumeration method can get the optimal security levels mapping scheme. However, it has the exponential time complexity $O(K^{N_j})$ that cannot be used for online decision.

Above four algorithms are simulated to compare our ECM algorithm in energy cost (in dollars) and queuing delay (in number of time slot). The characteristics of them and our ECM algorithm are

summarized in Table 5.

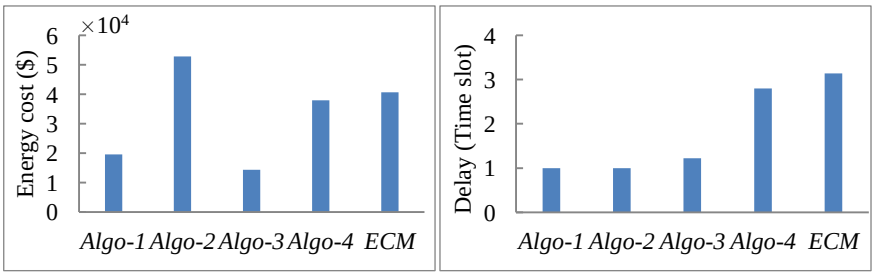
Table 5. Algorithms summary.

	Algo-1	Algo-2	Algo-3	Algo-4	ECM
Job queue	No	No	Yes	Yes	Yes
Security service	No	Yes	No	Yes	Yes
Security levels selection	No	Heuristic	No	Enumeration	Heuristic
Lyapunov technique	No	No	Yes	Yes	Yes

5.2 Performance Comparison of Five Algorithms

We fix the parameter $V = 10$ and conduct the five algorithms in energy cost and average delay. As shown in Fig. 8 (a), we can make the following observations about energy cost: 1) Compare with Algo-1 and Algo-2 respectively, Algo-3 and ECM have the Lower energy cost. This is because Algo-3 and ECM use the Lyapunov optimization technique to minimize the energy cost. The arriving jobs are queued in the IDC, which can be processed when the electricity price is low, i.e., the IDC operator can fully exploit the temporal diversities of electricity price; 2) Algo-2 exhibits more energy cost than Algo-1. This is reflected by the fact that each task in Algo-2 requires security services to ensure its security execution, which will incur a great amount of security workload and power demand for IDC (see Section 3.3). There is the same relationship between Algo-3 and ECM; 3) Algo-4 has the less energy cost than ECM algorithm. This is because Algo-4 uses enumeration method to select security levels that will result in optimal and minimum security workloads. However, the time complexity of Algo-4 is exponential. It is not applicable for online scheduling.

As for average delay shown in Fig. 8 (b), Algo-1 and Algo-2 have the same and lowest delay, this results from the fact that arriving jobs are not queued, and IDC operator executes these jobs once they are received. The ECM tends to have the longer average delay due to two reasons that: 1) arriving workloads in the queue are waiting for low electricity price; 2) security services result in more workload that IDC only processes fewer workloads in one time slot, which increases the length of workload queue. The Algo-3 has no security services but with job queue, the delay of which is medium. Because of less security workloads, the workload queue in Algo-4 will less than ECM's. So, Algo-4 outperforms ECM in average delay.



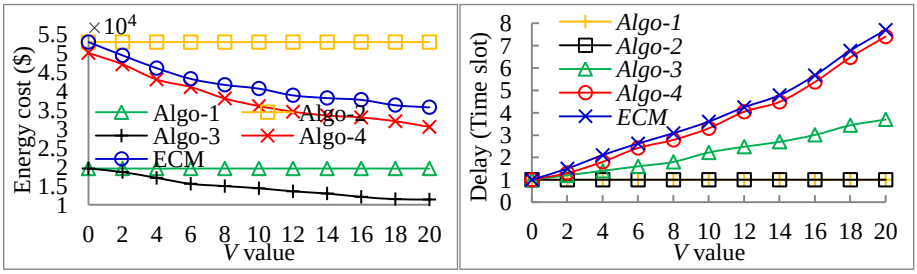
(a) Energy cost (b) Delay

Fig. 8. Energy cost and delay of five algorithms.

5.3 Performance Vary under Different Parameter V

Fig. 9 illustrates the performance of five algorithms under varying control parameter V. As Aglo-1 and Aglo-2 are independent of parameter V, we plot them as baselines in contrast with Algo-3, Algo-4 and

ECM. The parameter V controls the energy-delay tradeoff of Algo-3, Algo-4 and ECM. As shown in Fig. 9, the energy cost drops and the time-average delay grows as V goes from 0 to 20. The energy cost of Algo-1 and Algo-2 are always larger than the energy cost of Algo-3 and ECM, respectively, while they are equal when $V=0$. This is due to the fact that security services incur lots of energy cost, and we only care about the queue delay when parameter V is set to 0. Note that energy cost falls quickly at the beginning and then tends to descend slowly while the time-averaged queue backlog grows linearly with V. This finding confirms the $[O(1/V),O(V)]$ energy-delay tradeoff as captured in Eqs. (44) and (45). The energy cost and average delay of Algo-4 are always lower than the energy cost and average delay of ECM respectively. As the same reason explained above, Algo-4 has the less security workloads than ECM. In general, increasing V leads to larger delay as well as larger power cost reductions. Hence, parameter V in ECM controls the trade-off between delay and power cost. As without security workloads, Algo-1 and Algo-3 have the least energy cost and delay. However, IDC may suffer from the security threats, and the jobs may be failure without security services. In order to ensure the job risk probability constraint, security services should be used in spite of increasing the security workloads.



(a) Energy cost (b) Delay

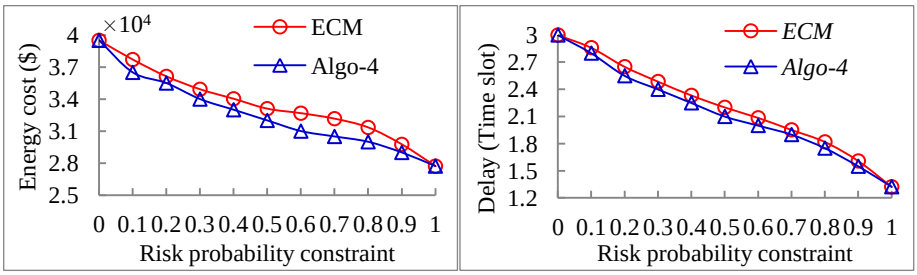
Fig. 9. Energy cost and delay under different parameter V.

5.4 Impact of Risk Probability Constraint

For purpose of revealing the impact of risk probability constraint of our ECM algorithm, we fix $V=10$ and assume that all jobs have the same risk probability constraint. This is not restrictive and only for experiment purpose in this section. The performance effects of varying risk probability constraint are reported in Fig. 10. It can be seen that the energy cost and delay become lower as risk probability increases. This phenomenon can be explained as follows: given a large risk probability constraint, the workload of security service is small according to Eqs. (7) and (8). Then, we need less electrical energy to execute the arriving tasks. What is more, The IDC operator can process more tasks in one time slot under the same computing resource that leads to lower average delay. Overall, though larger risk probability constraint will reduce the energy cost and delay, the jobs may experience more threats and attacks when executing in the IDC.

Generally, Algo-4 can map all tasks to the optimal security levels, and hence it has less security workloads than ECM. So, the energy cost and delay of Algo-4 are lower than ECM's. However, we can see from the Fig. 10(a) that the energy cost of Algo-4 is equal to the energy cost of ECM when the risk probability constraint is 0 and 1. This is due to the fact that the security levels mapping scheme is determined for Algo-4 and ECM algorithm in this case, i.e., $sl_i^{a(x)}=sl_i^{g(x)}=sl_i^{c(x)}=1$ for all tasks when $P_{risk}^c(j)=0$ and $sl_i^{a(x)}=sl_i^{g(x)}=sl_i^{c(x)}=0$ when $P_{risk}^c(j)=1$. Hence, Algo-4 and ECM algorithm have the same security workloads when $P_{risk}^c(j)=0$ or $P_{risk}^c(j)=1$. Similarly, in Fig. 10(b), the delay of Algo-4 is equal to the delay of ECM when $P_{risk}^c(j)=0$ or $P_{risk}^c(j)=1$. We can also conclude that the performance of

ECM algorithm is close to the Algo-4. However, the time complexity of ECM algorithm is polynomial, which is lower than the time complexity of Algo-4. In the cloud computing environment, IDC operator should make the online decisions to execute jobs or applications according to the real-time electricity price. So, only the algorithm with low time complexity can be used in this case.



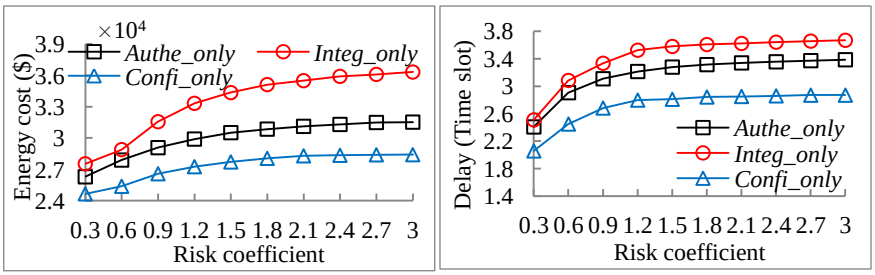
(a) Energy cost (b) Delay

Fig. 10. Energy cost and delay under different risk rate constraint.

5.5 Impact of Three Risk Coefficients

We have discussed the relationship between risk probability and risk coefficient in Section 5. For instance, the risk probability is zero when risk coefficient $\lambda^k = 0$. In this case, the IDC will not suffer this kind of malicious attack, and the corresponding security service does not need any more. In this section, we study how the risk coefficients have influence on our proposed algorithm. We fix V to be 10 and use abbreviations Authe_only, Integ_only and Confi_only to represent authentication service only, integrity service only and confidentiality service only respectively.

The simulation results are given in Fig. 11 for three risk coefficients. Overall, the Confi_only achieves the lowest energy cost and delay, Authe_only has the medium performances and Integ_only performs the worst. This can be explained by the fact that we set $\beta^c < \beta^a < \beta^g$ in Section 5.1. A larger parameter β will lead to more security workload. We can also see from Fig. 11 that the three curves are higher slope when parameter $\lambda^k \le 1.5, k \in \{a, g, c\}$, beyond which curves become flat. This can be explained by the fact that the risk probability changes evidently when the risk coefficient in the small range based on Eq. (13). At the same time, the energy cost and delay change with the same pace. Generally speaking, risk coefficients have significantly impacts on our ECM algorithm.



(a) Energy cost (b) Delay

Fig. 11. Impact of three risk coefficients.

5.6 Comparisons with Similar Work

This section is focused on comparisons with five similar algorithms. The purposes of all these algorithms listed in Table 6 are to minimize the energy cost of IDC in the deregulated electricity markets. eco-IDC [36] studies on electricity price prediction using statistical models or machine

learning techniques. However, both power prices and workload are stochastic in nature and can be hard to predict accurately. Algorithms eco-IDC [36], WBS (workload and battery scheduling) [39] and our proposed ECM consider the single queue for the job arriving and processing. Based on the model of geographically distributed data centers [2], our single queue model can be expanded to multi-queue model. The Lyapunov optimization technique is first proposed for network stability problems [42], and it can offer explicit performance guarantees in these stochastic settings. Because the change of electricity price is stochastic, the Lyapunov optimization technique can be very suitable for optimizing the energy cost for IDC. The service requirements of deadline guarantee or delay tolerant are all useful for some applications in the cloud computing environment. The problem service delay guarantee has been presented in next Section 5.7, and it will be seen as one of our future work. In a word, none of these algorithms takes the security guarantee into consideration except our proposed ECM algorithm. It is well known that security is a critical concern and even ranked as the greatest challenge in cloud computing environment. One of contributions of this paper is that a heuristic algorithm is devised for jobs to select appropriate security services to guarantee the job security.

Table 6. Comparisons with similar work.

Algorithms	Queue type	Method	Prediction or online	Deadline guarantee or delay tolerant	Security Guarantee
eco-IDC [36]	Single queue	Statistical Or machine learning	Prediction	Deadline guarantee	No
SAVE [2]	Multi-queue	Lyapunov optimization technique	Online	Delay tolerant	No
WBS [39]	Single queue	Lyapunov optimization technique	Online	Deadline guarantee	No
HBBF [6]	Multi-queue	mixed-integer nonlinear programming	Online	Delay tolerant	No
ECM	Single queue	Lyapunov optimization technique	Online	Delay tolerant	Yes

5.7 Research Contributions and Limitations

Initially, the energy cost optimization architecture is developed for the IDC operator. Next, the IDC resource and energy cost model, job arrival mode and security model are introduced. Then, the workload shaping with security guarantee is proposed, and a heuristic algorithm is also devised to solve security levels selection problem. Finally, we formulate the energy stochastic optimization problem and use our proposed ECM algorithm to schedule workloads. The prices used in our performance evaluation are the real-life electricity price. Our ECM algorithm is suited for delay tolerant big data applications, and hence it allows IDC operator to reduce energy cost at the expense of increased service delay. We have demonstrated that the cost-delay tradeoff is $[O(1/V), O(V)]$. Such tradeoff allows IDC

operators to make flexible design choices according to different application types and user contexts.

As noted above, our work has several contributions. However, it still has several limitations, including: 1) we assume that each job consists of a set of independent tasks that can be executed in parallel. Note that the components of a job can be correlated, for example, a job may be a scientific workflow that is typical big data application; 2) service delay guarantee is not considered, i.e., providing strict service delay bound has not been incorporated into the energy cost minimization problem, which is one of the major challenges in cloud computing environment. Part of our future efforts is to explore these issues.

6. Conclusions and Future Work

In this paper, we devise the energy cost optimization architecture for IDC operator to minimize the energy cost under the job risk probability constraint. The jobs may be delay tolerant big data applications or data intensive MapReduce applications that demands large-scale infrastructures such as Internet data center to provide computing resources. Due to high time complexity of optimal security levels mapping scheme, a heuristic algorithm with polynomial time complexity is developed to select security levels for tasks. Then, we formulate the energy stochastic optimization problem and propose our ECM algorithm to schedule workloads taking the temporal diversity of electricity price into account. The ECM algorithm, which is based on Lyapunov optimization framework, offers provable energy cost and delay guarantees. It aggressively and adaptively seizes the timing of low electricity price to process tasks, and defers delay-tolerant tasks execution when the price is high.

Four reference algorithms are conducted in our experiments in comparison with our ECM algorithm in terms of energy cost and queuing delay. The experiments confirm the $[O(1/V), O(V)]$ energy-delay tradeoff of ECM algorithm. However, the performance of ECM algorithm is close to the enumeration algorithm, but with lower time complexity. In a word, Extensive evaluation experiments based on the real-life electricity price demonstrate the effectiveness of our ECM algorithm.

As a future work, we plan to incorporate the big data scientific workflow scheduling method and delay guarantee into our energy cost optimization problem. Moreover, we are going to consider some new aspects in better usage of power in IDC, such as renewable energy, energy storage, battery and so on.

ACKNOWLEDGMENTS

This work was supported by the Key Program of Research and Development of China (2016YFC0800803), the National Natural Science Foundation, China (No.61272188, 61572162, 61572251), the Natural Science Foundation of Jiangsu Province (No.BK20131277), the Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2013-1-14), the Fundamental Research Funds for the Central Universities. Jidong Ge is the corresponding author.

REFERENCES

[1] Qureshi, R. Weber, H. Balakrishnan, J. Guttag, B. Maggs, Cutting the electric bill for internet-scale systems, ACM International Conference on the Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM 2009, Barcelona, Spain, pp. 17-21.

[2] Y. Yao, L.B. Huang, A.B. Sharma, L. Golubchik, M.J. Neely, Power cost reduction in distributed data centers: a two-time-scale approach for delay tolerant workloads, IEEE Transactions on Parallel and Distributed Systems 25 (1)

(2014) 200-211.

[3] G. Sun, H. Yu, V. Anand, L. Li, H. Di, Optimal provisioning for virtual network request in cloud-based data centers, Photonic Network Communications 24 (2) (2012) 118-131.

[4] G. Sun, H. Yu, L. Li, V. Anand, Y. Cai, H. Di, Exploring online virtual networks mapping with stochastic bandwidth demand in multi-datacenter, Photonic Network Communications 23 (2) (2012) 109-122.

[5] G. Sun, H. Yu, V. Anand, L. Li, A cost efficient framework and algorithm for embedding dynamic virtual network requests, Future Generation Computer Systems 29 (5) (2013) 1265-1277.

[6] H.J. Shao, L. Rao, Z. Wang, X. Liu, Z.B. Wang, K. Ren, Optimal load balancing and energy cost management for internet data centers in deregulated electricity markets, IEEE Transactions on Parallel and Distributed Systems 25 (10) (2014) 2659-2669.

[7] L. Rao, X. Liu, L. Xie, W. Liu, Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment, IEEE International Conference on Computer Communications, INFOCOM 2010, pp. 1-9.

[8] L. Rao, X. Liu, M.D. Ilic, J. Liu, Distributed coordination of internet data centers under multiregional electricity markets, Proceeding of IEEE 100 (1) (2011) 269-282.

[9] G. Sun, D. Liao, V. Anand, D. Zhao, H. Yu, A new technique for efficient live migration of multiple virtual machines, Future Generation Computer Systems 55 (2016) 74-86.

[10] G. Sun, D. Liao, D. Zhao, Z. Xu, H. Yu, Live migration for multiple correlated virtual machines in cloud-based data centers, IEEE Transactions on Services Computing (2016).

[11] R. Urgaonkar, B. Urgaonkar, M.J. Neely, A. Sivasubramaniam, Optimal power cost management using stored energy in data centers, the 2011 ACM International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2011, pp. 221-232.

[12] Y. Guo, Y. Fang, Electricity cost saving strategy in data centers by using energy storage, IEEE Transactions on. Parallel and Distributed Systems 24 (6) (2013) 1149-1160.

[13] F. Gens, IT cloud services user survey, pt.2: Top benefits & challenges, October 2008. URL http://blogs.idc.com/ie/?p=210.

[14] T. Xie, X. Qin, Scheduling security-critical real-time applications on clusters, IEEE Transactions on Computers 55 (7) (2006) 864-879.

[15] S. Song, K. Hwang, Y.K. Kwok, Risk-resilient heuristics and genetic algorithms for security-assured grid job scheduling, IEEE Transactions on Computers 55 (6) (2006) 703-719.

[16] T. Xie, X. Qin, Performance evaluation of a new scheduling algorithm for distributed systems with security heterogeneity, Journal of Parallel and Distributed Computing 67 (10) (2007) 1067-1081.

[17] X. Tang, K. Li, Z. Zeng, B. Veeravalli, A novel security-driven scheduling algorithm for precedence constrained tasks in heterogeneous distributed systems, IEEE Transactions on computers 60 (7) (2011) 1017-1029.

[18] L.F. Zeng, B. Veeravalli, X.R. Li, SABA: a security-aware and budget-aware workflow scheduling strategy in clouds, Journal of Parallel and Distributed Computing 75 (2015) 141-151.

[19] V. Chang, Towards a big data system disaster recovery in a private cloud, Ad Hoc Networks 35 (2015) 65-82.

[20] V. Chang, Y.H. Kuo, M. Ramachandran, Cloud computing adoption framework: a security framework for business clouds, Future Generation Computer Systems 57 (2016) 24-41.

[21] Z. J Li, J. D Ge, C. Y Li, H. J Yang, H. Y Hu, B. Luo, Energy cost minimization with risk rate constraint for internet data center in deregulated electricity markets, International Conference on Internet of Things and Big Data, IoTBD 2016, pp. 407-418.

[22] W. Jiang, K. Jiang, X. Zhang, Y. Ma, Energy optimization of security-critical real-time applications with guaranteed security protection. Journal of Systems Architecture-Embedded Systems Design 61 (7) (2015) 282-292.

[23] W. Yurcik, X. Meng, G. Koenig, J. Greenseid, Cluster security as a unique problem with emergent properties, Fifth LCI International Conference on Linux Clusters: The HPC Revolution 2004, May 2004.

[24] A. Behl, K. Behl, An analysis of cloud computing security issues, 2012 World Congress on Information and Communication Technologies, WICT 2012, pp. 109-114.

[25] M. Ali, S.U. Khan, A.V. Vasilakos, Security in cloud computing: opportunities and challenges, Information Science 305 (2015) 357-383.

[26] F. Azzedin, M. Muthucumaru, A trust brokering system and its application to resource management in public-resource grids, 18th International Parallel and Distributed Processing Symposium, IPDPS 2004, pp. 22-31.

[27] Z. J Li, J. D Ge, H. J Yang, L. G Huang, H. Y Hu, H. Hu, B. Luo, A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds, Future Generation Computer Systems (65) (2016) 140-152.

[28] V. Chang, The business intelligence as a service in the cloud, Future Generation Computer Systems 37 (2014) 512-534.

[29] V. Chang, G. Wills, A model to compare cloud and non-cloud storage of big data, Future Generation Computer Systems 57 (2016) 56-76.

[30] V. Chang, R.J. Walters, G.B. Wills, Organisational sustainability modelling-an emerging service and analytics model for evaluating cloud computing adoption with two case studies, International Journal of Information Management 36 (1) (2016) 167-179.

[31] J. Li, Z. Li, K. Ren, X. Liu, H. Su, Towards optimal electric demand management for internet data centers, IEEE Transactions on Smart Grid 2 (4) (2011) 1-9.

[32] L. Rao, X. Liu, L. Xie, W. Liu, Coordinated energy cost management of distributed internet data centers in smart grid, IEEE Transactions on Smart Grid 3 (1) (2012) 50-58.

[33] Z. Liu, M. Lin, A. Wierman, S.H. Low, L.L.H. Andrew, Greening geographical load balancing, the 2011 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2011, pp. 233-244.

[34] Z. J Li, J. D Ge, H. Y Hu, W. Song, H. Hu, B. Luo, Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds, IEEE transactions on Services Computing (2015).

[35] Z. Liu, M. Lin, A. Wierman, S.H. Low, L.L.H. Andrew, Greening geographical load balancing, IEEE/ACM Transactions on Networking 23 (2) (2015) 657-671.

[36] J. Luo, L. Rao, X. Liu, Temporal load balancing with service delay guarantees for data center energy cost optimization, IEEE Transactions on Parallel and Distributed Systems 25 (3) (2014) 775-784.

[37] L. Yu, T. Jiang, Y. Cao, Q. Zhang, Risk-constrained operation for internet data centers in deregulated electricity markets, IEEE Transactions on Parallel and Distributed Systems 25 (5) (2014) 1306-1316.

[38] G. Sun, H. Yu, V. Anand, D. Liao, L. Li, Power-efficient provisioning for online virtual network requests in cloud-based datacenters, IEEE Systems Journal 9 (2) (2015) 427-441.

[39] L. Yu, T. Jiang, Y. Cao, Q. Qi, Joint workload and battery scheduling with heterogeneous service delay guarantees for data center energy cost minimization, IEEE Transactions on Parallel and Distributed Systems 26 (7) (2015) 1937-1947.

[40] L. Yu, T. Jiang, Y. Cao, Energy cost minimization for distributed internet data centers in smart microgrids considering power outages, IEEE Transactions on Parallel Distributed System 26 (1) (2015) 120-130.

[41] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, C. Hyser, Renewable and cooling aware workload management for sustainable data centers, ACM International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2012, pp.175-186.

[42] L. Tassiulas, A. Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, IEEE Transactions on Automatic Control 37 (12) (1992) 1936-1949.

[43] L. Georgiadis, M.J. Neely, L. Tassiulas, Resource allocation and cross-layer control in wireless networks, Foundations and Trends in Networking 1 (1) (2006) 1-149.

[44] M. J. Neely, Stochastic network optimization with application to communication and queueing systems, Synthesis Lectures on Communication Networks, Morgan & Claypool publishers 2010, pp. 1-211.

[45] P. Shu, F. M Liu, H. Jin, M. Chen, F. Wen, Y. P Qu, B. Li, eTime: energy-efficient transmission between cloud and mobile devices, IEEE International Conference on Computer Communications, INFOCOM 2013, pp. 195-199.

[46] R. Urgaonkar, U. Kozat, K. Igarashi, M. Neely, Dynamic resource allocation and power management in virtualized data centers, IEEE Network Operations and Management Symposium, NOMS 2010, pp. 479-486.

[47] N. Do, Y. Zhao, S. T Wang, C. H Hsu, N. Venkatasubramanian, Optimizing offline access to social network content on mobile devices, IEEE International Conference on Computer Communications, INFOCOM 2014, pp. 1950-1958.

[48] L. Wang, Y. Lu, Efficient power management of heterogeneous soft real-time clusters, IEEE Real-Time Systems Symposium, RTSS 2008, pp. 323-332.

[49] F. Cao, M.M. Zhu, Energy-aware workflow job scheduling for green clouds, 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom/iThings/CPScom 2013, pp. 232-239.

[50] A.K. Mishra, J.L. Hellerstein, W. Cirne, C.R. Das, Towards characterizing cloud backend workloads: insights from google compute clusters, the Eleventh International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2010, pp. 34-41.

[51] M. Bishop, Computer security, Addison-Wesley, 2003.

[52] T. Xie, X. Qin, Security-aware resource allocation for real-time parallel jobs on homogeneous and heterogeneous clusters, IEEE Transactions on Parallel and Distributed Systems 19 (5) (2008) 682-697.

[53] A. Gandhi, M. Harchol-Balter, R. Das, C. Lefurgy, Optimal power allocation in server farms, the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2009, pp. 157-168.

[54] US NYISO, http://www.nyiso.com/, 2016.

*Biographies (Photograph)

Energy Cost Minimization with Job Security Guarantee in Internet Data Center

Zhongjin Li[a,b], Jidong Ge[a,b*],Chuanyi Li[a], Hongji Yang[c], Haiyang Hu[a,b,d] Bin Luo[a] and Victor Chang[e]

[a] *State Key Laboratory for Novel Software Technology, Software Institute, Nanjing University, Nanjing 210093, China*

[b] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876,China*

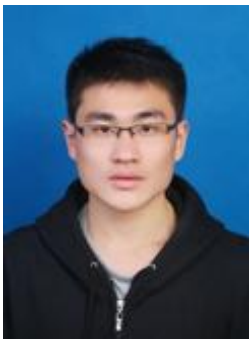
[c] *Centre for Creative Computing (CCC), Bath Spa University, England, UK*

[d] *School of Computer, Hangzhou Dianzi University, Hangzhou 310018, China*

[e] *International Business School Suzhou, Xi'an Jiaotong Liverpool University, Suzhou, China*

Zhongjin Li is a PhD candidate at the Software Institute, Nanjing University, under the supervision of Prof. Jidong Ge and Prof. Bin Luo. His research interests include cloud computing, workflow scheduling, stochastic network optimization.

Jidong Ge is an Associate Professor at Software Institute, Nanjing University. He received his PhD degree in Computer Science from Nanjing University in 2007. His current research interests include cloud computing, workflow scheduling, software engineering, workflow modeling, stochastic network optimization, process mining. His research results have been published in more than 40 papers in international journals and conference proceedings including JASE, ESA, ICSE, APSEC, ICSSP, HPCC, SEKE etc.



Chuanyi Li is a PhD candidate at the Software Institute, Nanjing University, under the supervision of Prof. Jidong Ge and Prof. Bin Luo. His research interests include software engineering, systems and software quality assurance, process modeling, simulation and improvement, process mining.

Hongji Yang is a full Professor and Deputy Director at Centre for Creative Computing (CCC), Bath Spa University, England, UK. He received his PhD degree in Computer Science from Durham University in 1994. His current research interests include cloud computing, computer network, workflow scheduling, and software engineering. His research results have been published in more than 150 papers in international journals and conference proceedings including JASE, ESA, ICSE, APSEC, ICSSP, HPCC, SEKE etc.

Haiyang Hu is a full Professor at School of Computer, Hangzhou Dianzi University, Hangzhou, China. He received the BS, MS, and PhD degrees in computer science from Nanjing University, Nanjing, China, in 2000, 2003, and2006, respectively. His research interests include cloud computing, computer network, workflow scheduling, mobile computing, and distributed computing. His research results have been published in more than 40 papers in international journals and conference proceedings including IEEE TVCG, ACM TALIP, JNCA, WPC, ISF, HPCC, DASFAA, ISF, APWeb etc.

Bin Luo is a full Professor at the Software Institute, Nanjing University. His main research interests include cloud computing, computer network, workflow scheduling, and software engineering. His research results have been published in more than 40 papers in international journals and conference proceedings. He is leading the institute of applied software engineering at Nanjing University.

Victor Chang is an Associate Professor in Information Management and Information Systems at International Business School Suzhou (IBSS), Xi'an Jiaotong Liverpool University, China. He is a Director of PhD Program and the 2016 European and Cloud Identity winner of "Best Project in Research". Victor Chang was a Senior Lecturer in the School of Computing, Creative Technologies at Leeds Beckett University, UK and a visiting Researcher at the University of Southampton, UK. He is an expert on Cloud Computing and Big Data in both academia and industry with extensive experience in related areas since 1998. He completed a PGCert (Higher Education) and PhD (Computer Science) within four years while working full-time. He has over 100 peer-reviewed published papers. He won £20,000 funding in 2001 and £81,000 funding in 2009. He was involved in part of the £6.5 million project in 2004, part of the £5.6 million project in 2006 and part of a £300,000 project in 2013. He won a 2011 European Identity Award in Cloud Migration and 2016 award. He was selected to present his research in the House of Commons in 2011 and won the best papers in 2012 and 2015. He has demonstrated ten different services in Cloud Computing and Big Data services in both of his practitioner and academic experience. His proposed frameworks have been adopted by several organizations. He is the founding chair of international workshops in Emerging Software as a Service and Analytics and Enterprise Security. He is a joint Editor-in-Chief (EIC) in International Journal of Organizational and Collective Intelligence and a founding EIC in Open Journal of Big Data. He is the Editor of a highly prestigious journal, Future Generation Computer Systems (FGCS). His security paper is the most popular paper in IEEE Transactions in Services Computing and his FGCS paper has one of the fastest citation rate. He is a reviewer of numerous well-known journals and had published three books on Cloud Computing which are available on Amazon website. He is a keynote speaker for CLOSER 2015/WEBIST2015/ICTforAgeingWell 2015 and has received positive support. He is the founding chair of IoTBD 2016 www.iotbd.org and COMPLEXIS 2016 www.complexis.org conferences.