Virtual Machines Scheduling in Mobile Edge Computing: A Formal Concept Analysis Approach

Fei Hao, Guangyao Pang, Zheng Pei, Keyun Qin, Yu Zhang, Xiaoming Wang

Abstract—Mobile Edge Computing (MEC) is providing cloud computing capabilities within the radio access networks and offering a new paradigm to liberate the mobile devices from heavy computational workloads. Importantly, MEC can effectively reduce latency, avoid congestion and prolong the battery lifetime of mobile devices by offloading the computation tasks from the mobile devices to a physically proximal MEC servers. Particularly, Virtual Machines (VMs) scheduling is a critical issue for tasks offloading and computation in MEC. Regarding to the VMs scheduling problem in MEC environmnet, this paper pioneers the use of Formal Concept Analysis (FCA) methodology for identifying the mapping from tasks to VMs. Specifically, the VMs profile and tasks descriptions are initially characterized as the formal contexts, respectively. With the constructed formal contexts, the corresponding formal concepts which refer to the rules set, are then generated. To better infuse the rules set of VMs and tasks, this paper defines a similarity measurement between formal concepts of VMs and tasks. Consequently, the matching problem from a given task to a virtual machine is to return the expected virtual machine according to the principle of maximum similarity degree between formal concepts of virtual machine and task. Extensive simulations are conducted with a real dataset for the validation of feasibility and effectiveness of the proposed approach. Specifically, the proposed approach can significantly reduce the energy consumption around 28% comparing to the approach without consideration of energy consumption. Overall, It is demonstrated that FCA-based VMs scheduling is a novel solution for a sustainable VMs scheduling in MEC environment.

Index Terms—Mobile Edge Computing, Virtual Machines, Formal Concept Analysis, Infusion/Matching.

1 INTRODUCTION

Mobile Edge Computing (MEC), as an emerging and extended commercial computing paradigm of Cloud Computing, is attracting much attention from both ICT industries and academia. Specifically, MEC provides cloud computing capabilities within the radio access networks (RAN), offers a new paradigm to liberate the mobile devices from heavy computation workloads [1], [2]. For traditional cloud computing systems, e.g., Amazon Web Services, Google Cloud Platform and Microsoft Azure, are leveraged and thus long latency may be incurred due to data exchange between users and clouds servers. Different from cloud computing system, MEC has the huge potential to significantly reduce latency, avoid congestion and prolong the battery lifetime of mobile devices by offloading the computation tasks from the mobile devices to a physically proximal MEC servers [3], [4].

The essential of computation tasks offloading onto cloud servers is the problem of virtual machines (VMs) scheduling. Technically, virtualization technology is playing a key role in cloud resources management and provisions, such as improving the utilization of cloud resources and service quality of users, reducing energy consumption, achieving load balance [5]. Further, such technology cloud provide users various services by encapsulating hardware computational resources [6]. At present, there have been many literatures which explore the VMs scheduling in cloud computing. VMs scheduling algorithms in cloud environment can be categorized into: task execution time aware scheduling algorithm, resource utilization aware scheduling algorithm, load balancing scheduling algorithm [7] and energy aware scheduling algorithm [8], [9]. However, there is no previous wok that uses Formal Concept Analysis (FCA) methodology for VMs scheduling in both cloud computing and MEC environments.

Thanks to the powerful ability of FCA for characterizing the relationships between objects and attributes. Aiming to find the appropriate mapping from VMs to tasks, this work pioneers the use of FCA for constructing the formal contexts for VMs profile as well as the user request (i.e., tasks descriptions); finding the best mapping for VMs scheduling by calculating the similarity between formal concepts of tasks and VMs. In summary, this paper made the following specific contributions.

- Regarding to the VMs scheduling problem in MEC environment, we are first to use formal concept analysis methodology for identifying the mapping from tasks to virtual machines. Specifically, the formal context of VMs and Tasks are constructed, then the

- F. Hao, G. Pang, Y. Zhang, X. Wang are with Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an, China & School of Computer Science, Shaanxi Normal University, Xi'an, China; Email:feehao@gmail.com; pangguangyao@gmail.com; zhangyu82@snnu.edu.cn; wangxm@snnu.edu.cn.
- Z. Pei is with the Center for Radio Administration & Technology Development, Xihua University, Chengdu, China; Email:pqyz@263.net.
- K. Qin is with the School of Mathematics, Southwest Jiaotong University, Chengdu, China; Email:keyunqin@263.net.

Corresponding author: X. Wang; Email: wangxm@snnu.edu.cn.

Manuscript received April 19, 2005; revised August 26, 2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2019.2894136, IEEE Transactions on Sustainable Computing

corresponding concept lattices are generated. Particularly, the attributes of these two formal context are exactly same. This important feature provides the way for calculating the similarity degree between two concepts and then further achieves the scheduling from tasks to virtual machines.

- To obtain the similarity between VMs and Tasks, the similarity degree between a task and a virtual machine is formally defined. We compare the common parts of intent of the concepts of VMs and Tasks. Intuitively, the more common elements they share, the higher similarity is.
- The extensive simulation demonstrate the significant performance for virtual machines scheduling in MEC environment by jointly considering the feasibility, efficiency and energy consumption of scheduling. The experimental results proved that our approach can reduce the energy consumption around 28% comparing to the VMs scheduling approach without consideration of energy consumption in MEC environment.

The rest of this paper is organized as follows. In Section 2, the related works in the literature are categorized and overviewed. Section 3 provides the preliminary knowledge about Formal Concept Analysis methodology and the relevant properties. The addressed problem is mathematically formulated in Section 4 via analyzing the major challenges of conventional distributed tasks scheduling model. Section 5 presents the FCA-based VMs scheduling strategy in MEC. The evaluation results of the proposed approach are provided in Section 6. Finally, Section 7 concludes this paper.

2 Related Work

This section will overview the existing literatures on VMs scheduling in cloud computing. VMs scheduling algorithms in cloud environment can be divided into: task execution time aware scheduling algorithm, resource utilization aware scheduling algorithm, load balancing scheduling algorithm and energy aware scheduling algorithm. Aiming at achieving the corresponding objectives, these scheduling algorithms is consisted of traditional scheduling algorithm, prediction-based scheduling algorithm, trust-based scheduling algorithm and heuristic intelligence based scheduling algorithm.

- Traditional Scheduling Algorithms [10], [11]: This kind of algorithms include Min-Min, rotation, first come first serve, etc. The principle of Min-Min algorithm is to schedule tasks to the virtual machine with the shortest expected execution time. Rotation algorithm is to assign user tasks to the virtual machine of cloud data center in turn. The first come first served algorithm is to allocate virtual machines in turn according to the order of arrival.
- Prediction based scheduling algorithm [12], [13], [14]: This kind of algorithm usually has a prediction model to predict the demand of the next task according to the running condition of the historical task, so it can create virtual machine in advance or reserve resources to ensure that the task is scheduled to a reasonable virtual machine.
- Trust-based scheduling algorithm [15], [16]. This kind of algorithm introduces the basic concept of trust in cloud computing, establishes the trust relationship between virtual machine and physical machine, and quantifies the impact of scheduling results on security and reliability under different trust relationships by using trust benefit function, thus guaranteeing the quality of scheduling algorithm.
- Heuristic intelligence based scheduling algorithm [11], [17], [18]. The heuristic algorithm includes genetic algorithm, ant colony algorithm, particle swarm optimization algorithm, simulated annealing algorithm and so forth. The heuristic algorithm has a good global optimization search ability, and its application to virtual machines scheduling can greatly improve the quality of scheduling.
- Scheduling algorithms based on hybrid optimization [19], [20]. Because hybrid optimization algorithm has better optimization effect than single optimization algorithm, some researchers use hybrid optimization algorithm in virtual machines scheduling.

The above algorithms mainly consider a single scheduling objective, and lack of comprehensive consideration of each scheduling objective. To address this issue, some researchers have proposed a multi-objective scheduling algorithm for virtual machines, and also achieved the corresponding results. Multi-objective based virtual machine scheduling algorithms can be divided into two kinds, one is to use the above algorithm to realize virtual machine scheduling after processing multiple targets into one target, the other is to establish a multi-objective optimization model for virtual machine scheduling and use the existing multi-objective optimization algorithm to solve it. Different from the general multi-objective optimization problem, this paper attempts to use FCA for tackling this scheduling problem.

3 Preliminaries

Aiming to understand the working process of VM scheduling using FCA, the relevant preliminaries and notations about FCA are provided in this section.

3.1 Formal Concept Analysis

Formal Concept Analysis (FCA) [21] as a powerful computational intelligence methodology for data analysis and rule extraction from the formal context, is widely used in various domains. It is used to characterize the relationships between objects and attributes in a domain. First, a formal context including this binary relationships between objects and attributes is constructed. Further, the objects and attributes are grouped into concepts, and then a formal concept lattice of these concepts can be built up.

Definition 1. ***(Formal Context)*** *A formal context is represented as a 3-tupe K=(O, A, I), where O=$(o_1, o_2, \cdots, o_n)$ indicates a set of objects, A=$(a_1, a_2, \cdots, a_m)$ refers to a set of attributes, and I is the binary relation between O and A, and $(o, a) \in I$ is interpreted as "object o has the attribute a". Usually, we remark this binary relation between objects and attributes with "$\times$".*

Definition 2. ***(↑ and ↓ Operators)*** *Given a formal context K=(O, A, I), the ↑ and ↓ operators on $X \subseteq O$ and $B \subseteq A$ are defined as,*

$$X^{\uparrow} = \{a \in A | \forall x \in X, (x, a) \in I\} \quad (1)$$

$$B^{\downarrow} = \{x \in O | \forall a \in B, (x, a) \in I\} \quad (2)$$

The above both operations are the Galois correspondence of the formal context (O, A, I). $X^{\uparrow}$ is called dual of A and $B^{\downarrow}$ is called dual of O [22].

Definition 3. ***(Extent and Intent of Concept)*** *A concept is defined as a pair C=(X, B) if $X^{\uparrow}$=B and $B^{\downarrow}$=X. Here, X is called the extent of concept; and B is called the intent of concept.*

Definition 4. ***(Super-concept and Sub-concept)*** *Given a formal context K=(O, A, I), and two concepts C_1=(X_1, B_1) and C_2=(X_2, B_2). The partial order $C_1 \ll C_2$ indicates that concept C_2 is the super-concept of concept C_1, and C_1 is the sub-concept of C_2. Formally, the following inequations holds.*

$$(X_1, B_1) \ll (X_2, B_2) \Leftrightarrow X_1 \subseteq X_2 (\Leftrightarrow B_1 \supseteq B_2) \quad (3)$$

Definition 5. *[23]**(Concept Lattice)** A concept lattice can be organized and further built up with the all concepts in terms of their partial order $\ll$, denoted as $L(\mathcal{C}, \ll)$. Its graphical representation is a Hasse diagram [21], [23] where each node indicates a concept and each link between the concepts implies a kind of partial order.*

Example 1. *For a given formal context K which contains 4 objects $\{o_1, o_2, o_3, o_4\}$ and 5 attributes $\{a, b, c, d, e\}$. The binary relations between objects and attributes are shown in Table 1.*

TABLE 1
A Formal Context

O/A	a	b	c	d	e
o_1	×	×		×	×
o_2	×	×	×		
o_3					×
o_4	×	×	×		

By using concept lattice generation algorithm and lattice visualization software Galicia [1]*, the corresponding concept lattice is shown in Figure 1.*

As can be seen from Figure 1, each node indicates a formal concept including extent and intent, such as a concept $(\{o_1, o_2, o_4\}, \{a, b\})$, the extent is $\{o_1, o_2, o_4\}$ and intent is $\{a, b\}$. It is interpreted as the objects $\{o_1, o_2, o_4\}$ own the common attributes $\{a, b\}$. These formal concepts follow the partial order within the concept lattice. Note that the concept $(\{o_1, o_4\}, \{a, b, c\})$ is the sub-concept of the concept $(\{o_1, o_2, o_4\}, \{a, b\})$, and the concepts $(\{o_1, o_2, o_4\}, \{a, b, \})$ and $(\{o_1, o_3\}, \{e\})$ are the super-concepts of the concept $(\{o_1\}, \{a, b, d, e\})$.

4 Problem Statement

In this section, the major challenges of conventional distributed tasks scheduling model are analyzed. Then, the problem formulation on virtual machine scheduling in Mobile Edge Computing environment is mathematically provided.

1. http://www.iro.umontreal.ca/~galicia/

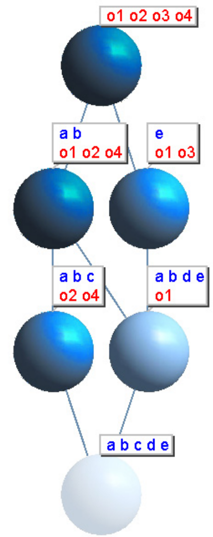

Fig. 1. The Concept Lattice of Formal Context K

4.1 Challenge

Differs with conventional distributed tasks scheduling model, MEC environment, as a large-scale distributed computing paradigm, is facing some unique challenges. As we known, the one-to-one mapping relation between task and physical device could be established in the conventional scheduling model. However, a complex task T is usually partitioned into several sub-tasks, denoted as $T = \{t_1, t_2, \cdots, t_n\}$ which are mapped into multiple virtual machines (as shown in Figure 2). In particular, multiple virtual machines can be placed in the same physical device. In other words, the relation between the tasks and physical devices in MEC are no longer a simple correspondence. With the increasing scale of the problem, the difficulty of scheduling is dramatically increasing [24].

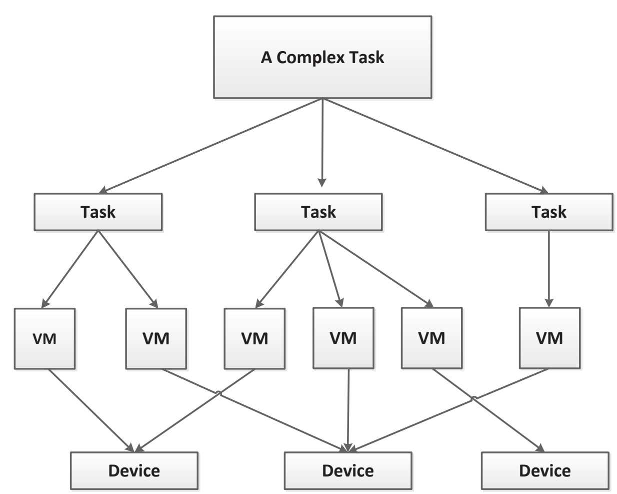

Fig. 2. Virtual Machines Scheduling in Mobile Edge Computing

4.2 Problem Formulation

In order to derive the problem addressed in this work, the following relevant definitions are firstly presented. Further, the problem formally is stated.

Definition 6. ***(VM Scheduling)*** *In Mobile Edge Computing environment, VM scheduling is defined as a mapping between tasks set* $T = \{t_1, t_2, \cdots, t_n\}$ *and virtual machines set* $V = \{v_1, v_2, \cdots, v_m\}$*. It is formulated as follows,*

$$L|\{l_k\} : T \mapsto V \quad (k = 1, 2, \cdots, p) \tag{4}$$

Since VM scheduling in this paper takes both computational capacity and energy into account, the corresponding definition are as follows.

Definition 7. ***(Computational Capacity)*** *We devise the parameter* μ *for evaluating the computational capacity, it is defined as follows,*

$$\mu = \alpha * C * P + \beta * M + \eta * B \tag{5}$$

where C *denotes the numbers of CPU,* P *refers to the processing ability (MPIS),* M*(MB) indicates the memory capacity, and* B *is the average network bandwidth;* α*,* β*, and* η *are the weights for computational capacity, memory capacity and network bandwidth. The configuration of this parameter can enable the VMs fit the various demanding. For example, for a computation-intensive model, we may empirically set* α*=40%,* β*=20%,* η*=40%; and, we may set* α*=20%,* β*=50%,* η*=30% regarding to the resource-intensive model.*

Definition 8. ***(Energy)*** *The energy evaluation for VMs is dependent on the accurate energy model. We regard the CPU, memory, and disk are the major components that consume most of the power in MEC environment. Therefore, the energy model is represented as,*

$$E = E_{CPU} + E_{RAM} + E_{Disk} \tag{6}$$

where E *represents the system total energy consumption;* E_{CPU}*,* E_{RAM} *and* E_{Disk} *denote energy consumption on CPU, memory and disk, respectively.*

Problem 1. ***(VMs Scheduling in Mobile Edge Computing Environment)*** *Based on the above constraints, VMs scheduling in MEC can be formulated as*

$$f(\mu, E) : T \mapsto V \tag{7}$$

The above problem can be interpreted as: the addressed problem aims to find a best mapping from T to V under the constraints of computational capacity μ and energy E. Intuitively, a best mapping from T to V implies that we try to find a optimized scheduling strategy which maximize the computational capacity and minimize the energy consumption.

5 FCA-BASED VMS SCHEDULING IN MEC

This section focuses on the proposed approach about VMs scheduling in MEC based on FCA. First of all, a overall framework of our proposed approach is provided. Then, the detailed implementations for FCA-based VMs scheduling in MEC are further elaborated.

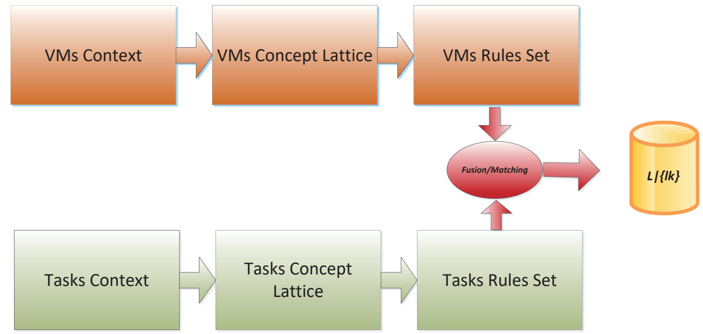

Fig. 3. The Overall Solution Framework

5.1 The Overall Solution Framework

Figure 3 shows the overall solution framework on FCA-based VMs scheduling in MEC environment.

Clearly, we can see that the overall solution framework is composed of three key modules:

- (**FCA-based VMs Profile Characterization**): This module is in charge of characterizing the profiles of VMs by using FCA methodology. The basic idea is that (1) a formal context about VMs and its performance metrics is constructed; (2) with the constructed VMs context, the corresponding VMs concept lattice is built up; (3) further, the VMs rules set derived from VMs profile is obtained.
- (**FCA-based Tasks Profile Characterization**): Similar with VMs profile characterization, another module is to characterize the profiles of tasks by using FCA. According to the users' request for tasks, (1) we may construct a formal context for tasks, called Tasks Context; (2) then, the Tasks Concept Lattice could be generated based on the tasks context; (3) finally, the tasks rules set derived from Tasks profile is extracted.
- (**Fusion/Matching**): This module is used to infuse the VMs rules and task rules for obtaining the best matching (i.e., allocation scheme). Technically, the similarity between formal concepts of tasks and those of VMs are evaluated. According to the principle of maximum similarity degree, the task rules could be effectively infused with VMs rules. In other words, the resulting mappings from VMs to tasks are generated.

5.2 Implementations

This section focuses on the elaboration of implementations for FCA-based VMs scheduling in MEC environment.

5.2.1 FCA-based VMs Profile Characterization

In order to characterize the VMs profile, a formal context of VMs is initially constructed. Concretely, the different VMs are regarded as the objects , and the number of CPU C, processing capacity P, memory capacity M, bandwidth B, and energy consumption grade E are regarded as the attributes in the constructed formal context. Formally, the formal context is represented as follows.

$$K = (V, A, I) \tag{8}$$

where $V = \{v_1, v_2, \cdots, v_m\}$, $A = \{C, P, M, B, E\}$, and I denote the binary relations between VMs and attributes. $I \subseteq V \otimes A$, $(v, a) \in I$ denotes that VM v has the attribute a, and $(v, a) \notin I$ denote that VM v does not have the attribute a, where $v \in V$, $a \in A$.

Remark 1. *In many literature, if $(v, a) \in I$, then we mark it with $\times$; otherwise, the blanks are given for $(v, a) \notin I$.*

According to Definition 1, the generic form of formal context for VMs profile characterization is shown in Figure 4.

	CPU(C)				Processing Ability(P)				Memory Capacity(M)				Energy Consumption(E)				Band width
	$c1$	$c2$	...	$cm1$	$p1$	$p2$	...	$pm2$	$M1$	$M2$	...	$Mm3$	$E1$	$E2$	...	$Em4$	B
$v1$		X				X				X				X			X
$v2$				X			X			X				X			X
...																	X
vn		X						X		X				X			X

Fig. 4. A Generic Form of Formal Context for VMs Profile

Clearly, v_1 is a kind of VM which including the c_2 number of CPUs, processing ability p_2, M_2 size of memory, energy consumption grade E_2 as well as bandwidth B. Note that the bandwidth B is a constant parameter in MEC environment.

Remark 2. *If all VMs are with the same product model (we called* ***Isomorphic VMs****), then they have the same processing ability, i.e., p_i. On the contrary, they have different processing ability if these VMs are with the different product model (we called* ***Heterogeneous VMs****).*

In this paper, the experimental values for the above main parameters can be obtained from the configurations. However, the energy consumption evaluation cannot be directed calculated. The following section will provide the measurement of energy consumption.

5.2.2 Energy Consumption Evaluation

We devise and propose an energy consumption model from the systematic point of view. A fundamental rule of an energy consumption model that should be valuable at the system level is that the model ought to be characterized by easily accessible parameters. As we known, the CPU, memory and disk are the major components for consuming most of the system's energy. Formally, the energy consumption model of VM can be evaluated as follows.

$$E_{system} = E_{CPU} + E_{memory} + E_{disk} \tag{9}$$

where E_{system} denotes the total energy consumption of the VM system; E_{CPU}, E_{memory}, and E_{disk} represent the energy consumption of CPU, memory and disk, respectively.

CPU Energy Consumption Model: One of the biggest power dependants of a VM is CPU [25]. The processor's energy loss consists of a static and dynamic segment, with the static segment being around steady and the dynamic segment fluctuating with the action of the processor.

One main metric that has been regarded as a sensible portrayal of dynamic action is CPU utilization. To avoid complicated training, the energy consumption model can be simplified into the following form:

$$E_{CPU} = E_{CPU_{Idle}} + (E_{CPU_{max}} - E_{CPU_{Idle}}) * U \tag{10}$$

where $E_{CPU_{Idle}}$ and $E_{CPU_{max}}$ indicate the CPU idle and maximum of processor power, which can be evaluated by physical meter; U refers to the CPU utilization, which can be calculated with *cores of VMs/cores of physcial devices* $\times \sharp$ *channles*.

Example 2. *Considering a HP Proliant G_5 VM characterised by an idle and a maximum powers of 2248.8 and 3240 W respectively, therefore the total energy consumed by a CPU utilization of 30% can be easily calculated as:*

$$E_{CPU} = 2248.8 + (3240 - 22248.8) * 30\% = 2546.16W \tag{11}$$

Memory Energy Consumption Model: In addition to CPU energy consumption, memory energy consumption occupies the second largest place among the consumption on the VMs. Extensive evidences show that the main memory consumes about 30% of the total energy [26]. The memory energy is mainly caused from memory processing and page swapping. Usually, the memory energy consumption model is composed of idle and active power. Formally, it is express as follows.

$$E_{memory} = E_{RAM_{idle}} + E_{RAM_{active}} \tag{12}$$

where $E_{RAM_{idle}}$ and $E_{RAM_{active}}$ represent the idle power and active power of memory, respectively. Equation (12) is used in our energy consumption model due to the fact that the memory is produced by different vendors and the architecture and design might differ.

Disk Energy Consumption Model: Nowadays, Solid-State Drive (SSD) as a mainstream data storage media used in data centers VMs. SSDs have no moving mechanical components. Disk is the subsystem that is hardest to model correctly because of the difficulty arising due to the lack of visibility into the power states of the disk drive and the impact of disk hardware caches. SSDs comparatively consume lesser power than Hard-Disk Drives (HDD). The disk energy consumption model can be formulated as,

$$E_{Disk} = E_{Disk_{idle}} + C_r M_{read} + C_w M_{write} \tag{13}$$

where M_{read} and M_{write} are read and write speeds respectively. C_r and C_w are constants. This model is known as the throughput-disk based model.

Example 3. *Given a set of heterogenous VMs $V = \{v_1, v_2, \cdots, v_9\}$, i.e., their processing ability are different; the number of CPUs for those VMs $C = \{1, 2, 4\}$; the processing ability for V is denoted as $P = \{200MHZ, 300MHZ, 400MHZ\}$; the memory for V is $M = \{0.5G, 1G, 2G\}$; and the bandwith B=100M. According to the above approach for VMs profile characterization based on FCA, we thus construct the formal context of VMs as shown in Figure 5.*

Further, the corresponding concept lattice (as shown in Figure 6) of the above formal context can be built up according to Definition 5.

With Definition 2, for a set of VMs $X' = \{v_3, v_9\}$, their common attributes are expressed as $B' = \{c_2, M_3\}$. Similarly,

	$c1$	$c2$	$c3$	$p1$	$p2$	$p3$	$M1$	$M2$	$M3$	B	$E1$	$E2$	$E3$
$v1$		X		X			X			X	X		
$v2$	X				X			X		X	X		
$v3$		X		X					X	X	X		
$v4$		X				X	X			X			X
$v5$	X				X				X	X			X
$v6$			X			X		X		X		X	
$v7$	X			X			X			X		X	
$v8$		X		X				X		X			X
$v9$		X			X				X	X			
$v10$	X					X		X		X		X	

Fig. 5. Constructed Formal Context of VMs

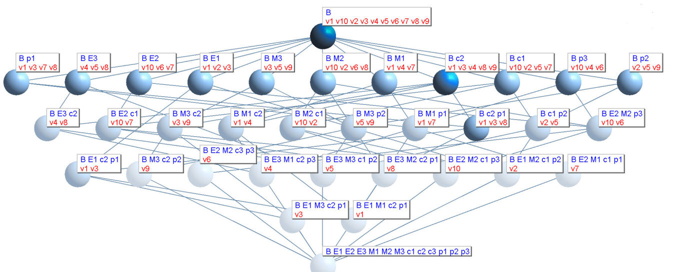

Fig. 6. Concept Lattice of VMs

for a set of attributes $B^{''} = \{c_2, p_1\}$, the related VMs who own these attributes are expressed as $A^{''} = \{v_1, v_3, v_8\}$. In Figure 6, each node indicates a concept, such as $(\{v_5, v_9\}, \{p_2, M_3, B\})$ is a formal concept which can describe the features of VMs profile.

5.2.3 FCA-based Tasks Profile Characterization

To realize the precision scheduling of VMs, we need to investigate the tasks profile which could be characterized via users' request. Figure 7 shows a skeleton frame about user's request on task execution in terms of personalized demanding on the number of CPUs, processing ability, memory capacity, energy consumption grade as well as bandwidth.

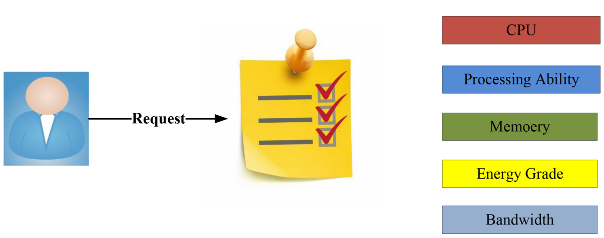

Fig. 7. User's Request on Task Execution

Generally, for different task, a user may has the different request according to the requirements for task. That is to say, if we take the tasks as the objects, the requirements as the attributes, it is easily to construct the formal context of tasks. Formally, the formal context can be represented as

$$K = (T, A, R) \tag{14}$$

where T is a set of tasks, i.e., $T = \{t_1, t_2, \cdots, t_m\}$; $A = \{C, P, M, B, E\}$, and R denote the binary relations between tasks and users' request for the tasks. $R \subseteq T \otimes A$, $(t, a) \in R$ denotes that user's requirement for task t is a, and $(t, a) \notin R$ denote that user has no requirements for task t on the aspect of a, where $t \in T$, $a \in A$.

Example 4. *On the basis of Example 2, we have a set of tasks $T = \{t_1, t_2, \cdots, t_5\}$, and the user's requirements are exactly same with the attributes of VMs. Hence, similar with the process of constructing the formal context of VMs in Example 2, the constructed formal context of tasks is shown in Figure 8. Further, the corresponding concept lattice (as shown in Figure 9) of the above formal context can be built up according to Definition 5.*

	$c1$	$c2$	$c3$	$p1$	$p2$	$p3$	$M1$	$M2$	$M3$	B	$E1$	$E2$	$E3$
$t1$		X		X			X			X	X		
$t2$	X				X			X		X		X	
$t3$		X		X					X	X	X		
$t4$		X				X	X			X			X
$t5$	X				X				X	X			X

Fig. 8. Constructed Formal Context of Tasks

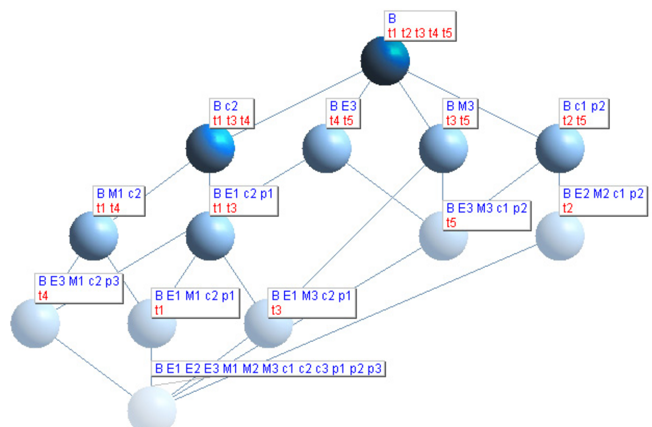

Fig. 9. Concept Lattice of Tasks

5.2.4 Fusion/Matching from Tasks to VMs

The above two sections have provided the construction process for formal context of VMs and Tasks, however we need to figure out the scheduling strategies by fusiong/matching from tasks to VMs according to their generated formal concepts which describe the users' requirements for tasks as well as VMs' their own performance.

For the sake of fusion and matching from tasks and VMs, the similarity degree between formal concepts of VMs and Tasks is firstly defined.

Definition 9. ***(Similarity Degree between Formal Concepts of VMs and Tasks)*** *Let L_V and L_T be the concept lattices for VMs and Tasks, respectively. For any given concepts $(X, A) \in L_V$ and $(Y, B) \in L_T$, the similarity degree between $(X, A) \in L_V$ and $(Y, B) \in L_T$ is defined as,*

$$sim((X, A), (Y, B)) = \frac{|A \cap B|}{|A \cup B|} \tag{15}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2019.2894136, IEEE Transactions on Sustainable Computing

Clearly, we found that the more common elements of intents A and B they have, the higher similarity is.

With the above similarity degree, for any given task t_i, the matching problem from a task to a virtual machine is to return the expected virtual machine $\widehat{v}_j \in V$ according to the principle of maximum similarity degree between formal concepts of virtual machine and task. It is formulated as follows.

$$\widehat{v}_j := \arg\max_{1 \leq j \leq n} sim((t_i^{\uparrow\downarrow}, t_i^{\uparrow}), (v_j^{\uparrow\downarrow}, v_j^{\uparrow})). \quad (16)$$

Eq. (16) indicates that the task t_i should be assigned onto virtual machine $\widehat{v}_j$.

Example 5. *Let us continue the Example 2 and Example 3, the similarity degree between VMs and tasks are easily obtained according to Eq. (15). The results are shown in Table 2.*

TABLE 2
Similarity degree between VMs and Tasks

VM \ Task	t_1	t_2	t_3	t_4	t_5
v_1	1	$\frac{1}{9}$	$\frac{4}{6}$	$\frac{3}{7}$	$\frac{1}{9}$
v_2	$\frac{2}{8}$	$\frac{4}{6}$	$\frac{2}{8}$	$\frac{1}{9}$	$\frac{3}{7}$
v_3	$\frac{4}{6}$	$\frac{1}{9}$	1	$\frac{2}{8}$	$\frac{2}{8}$
v_4	$\frac{3}{7}$	$\frac{1}{9}$	$\frac{2}{8}$	1	$\frac{2}{8}$
v_5	$\frac{1}{9}$	$\frac{3}{7}$	$\frac{2}{8}$	$\frac{2}{8}$	1
v_6	$\frac{1}{9}$	$\frac{3}{7}$	$\frac{1}{9}$	$\frac{2}{8}$	$\frac{1}{9}$
v_7	$\frac{3}{7}$	$\frac{3}{8}$	$\frac{2}{8}$	$\frac{2}{8}$	$\frac{2}{8}$
v_8	$\frac{3}{7}$	$\frac{2}{8}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{2}{8}$
v_9	$\frac{2}{8}$	$\frac{2}{8}$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{4}{6}$
v_{10}	$\frac{1}{9}$	$\frac{4}{6}$	$\frac{1}{9}$	$\frac{2}{8}$	$\frac{2}{8}$

Obviously, we can get the following allocation rules,

- t_1 *should be assigned onto virtual machine* v_1 *since the largest similarity degree* $sim((t_1^{\uparrow\downarrow}, t_1^{\uparrow}), (v_1^{\uparrow\downarrow}, v_1^{\uparrow}))$*=1;*
- t_2 *should be assigned onto virtual machine* v_2 *or* v_{10} *since the largest similarity degrees* $sim((t_2^{\uparrow\downarrow}, t_2^{\uparrow}), (v_2^{\uparrow\downarrow}, v_2^{\uparrow})) = \frac{4}{6}$*, and* $sim((t_2^{\uparrow\downarrow}, t_2^{\uparrow}), (v_{10}^{\uparrow\downarrow}, v_{10}^{\uparrow})) = \frac{4}{6}$*;*
- t_3 *should be assigned onto virtual machine* v_3 *since the largest similarity degree* $sim((t_3^{\uparrow\downarrow}, t_3^{\uparrow}), (v_3^{\uparrow\downarrow}, v_3^{\uparrow}))$*=1;*
- t_4 *should be assigned onto virtual machine* v_4 *since the largest similarity degree* $sim((t_4^{\uparrow\downarrow}, t_4^{\uparrow}), (v_4^{\uparrow\downarrow}, v_4^{\uparrow}))$*=1;*
- t_5 *should be assigned onto virtual machine* v_5 *since the largest similarity degree* $sim((t_5^{\uparrow\downarrow}, t_5^{\uparrow}), (v_5^{\uparrow\downarrow}, v_5^{\uparrow}))$*=1;*

6 EVALUATION

In this section, we firstly present the details about collection of dataset; then, the evaluation results and analysis are provided.

6.1 Dataset

The experimental dataset about VMs are collected from our own cloud platform which consists of software and hardware modules.

- Software Module: Our platform adopts the virtualized system InCloud Sphere developed by Inspur Corporation [2] (a leading provider company on cloud computing and big data services).
- Hardware Module: The hardware is composed of 3 NF8460M4 servers (physical devices). Through a series of self-controllable core technologies, the platform abstracts physical server resources and transforms physical resources such as CPU, memory, network, storage, and I/O into logical resources that can be managed, scheduled, and distributed. In addition, this platform could achieve the high resource utilization, flexible and dynamic resource allocation requirements, and faster business response speed.

As mentioned before, VMs energy consumption are mainly dependent on CPU, memory and disk. The relevant experimental parameters are given as follows.

1) (**Parameters for CPU**) Table 3 shows the relevant parameters for CPUs.

TABLE 3
Configuration of CPUs

CPU Type	$E_{CPU_{idel}}$	$E_{CPU_{max}}$	Processing Ability
Xeon E7-4830v3	45W	115W	2.13G Hz
Xeon E3-1220v5	31W	80W	3.0G Hz

(Remark: Xeon E7-4830v3 has 12 Cores & double channel memory; and Xeon E3-1220v5 has 4 Cores & single channel memory

2) (**Parameters for Memories**) Table 4 shows the vendors for 4GB capacities [28]. Micron, as one of the leading manufacturers of server memories, is considered as our benchmark for the prediction of the memory power.

TABLE 4
Energy Consumption Parameters by Vendor for 4GB capacity

Vendor	$E_{RAM_{active}}$	$E_{RAM_{idle}}$
Micron	9.3	2.6

3) (**Parameters for Disk**) We generated random VMs according the VMs configuration given in Table 5.

Disk(GB)	Read/Write(Mbps)
50	34.5
100	69.5
150	103.2
250	172.5

TABLE 5
Parameters for Disk

Based on the above configurations, we create 20 virtual machines including 14 type-E7 VMs and 6 type-E3 VMs. The energy consumption for type-E7 VMs and type-E3 VMs are presented in Table 6 and Table 7, respectively.

To simplify the representation of energy consumption in the constructed formal context, we adopt the coarse granularity representation for dividing the energy consumption into 5 levels (In this paper, we emprically set the following 5 levels: $\leq$66, [67,76], [77,96],[97,126], $\geq$127). Besides, the bandwidth is taken into account as well. Consequently, the formal context of VMs (as shown in Figure 10) is constructed as follows.

Apparently, this formal context includes 20 VMs and 20 essential attributes. According to the FCA approach, the corresponding concept lattice can be generated as follows.

2. http://en.inspur.com/

TABLE 6
Energy Consumption with Type-E7 VM

VM ID	CPU	Memory	Disk	Total Energy
1	1	1	50	65.6106
2	1	2	100	65.8073
3	1	4	150	65.9967
4	2	2	50	68.5272
5	2	4	150	68.9133
6	4	4	100	74.5573
7	4	8	150	86.6467
8	4	16	250	87.0361
9	6	4	100	80.3906
10	6	8	150	92.48
11	8	8	100	98.1239
12	8	16	250	122.503
13	16	16	150	145.447
14	16	32	250	193.436

TABLE 7
Energy Consumption with Type-E3 VM

VM ID	CPU	Memory	Disk	Total Energy
1	1	1	50	60.9439
2	2	2	50	73.1939
3	4	4	100	97.8906
4	4	16	250	110.369
5	6	8	150	165.48
6	8	16	250	214.169

In MEC environment, a task $T = \{t_1, t_2, t_3\}$ is given. When a user send a request to MEC, i.e., a request could be characterized with both sub-tasks and corresponding attributes. Let us assume that the tasks profile (T, A) as the formal context (shown in Figure 12). The corresponding concept lattice can be generated as follows.

6.2 Experimental Results

In order to achieve the best tasks allocation/VMs scheduling, the following similarity between formal concepts of VMs and tasks are provided in Table 8. Similarly, the tasks allocation rules are obtained as follows:

- t_1 should be assigned onto virtual machine $v_1, v_8, v_{10}, v_{11}, v_{12}, v_{16}, v_{17}$ since the similarity degree between t_1 and those VMs are $\frac{3}{7}$ which is the largest one (as shown in Column 1 of Table 8).
- t_2 should be assigned onto virtual machine $v_2, v_9, v_{13}, v_{14}, v_{15}$, since the similarity degree between t_2 and those VMs are $\frac{3}{7}$ which is the largest one (as shown in Column 2 of Table 8).
- t_3 should be assigned onto virtual machine $v_3, v_5, v_{16}, v_{17}, v_{19}, v_{20}$, since the similarity degree between t_3 and those VMs are $\frac{3}{7}$ which is the largest one (as shown in Column 3 of Table 8).

In addition, we also compare the VMs scheduling approaches with/without consideration of energy consumption in MEC. Toward to this, the energy consumption is acted as the performance evaluation metric for two comparison approaches. To realize the normalization comparison, the following weighted average energy consumption (WAE) is defined.

Definition 10. ***(Weighted Average Energy Consumption)*** *Suppose tasks $t_1, t_2, \cdots, t_m$ are assigned to VMs $v_1, v_2, \cdots, v_m$,*

	CPU (C)						Processing Ability (P)		Memory Capacity (M)						Energy Consumption (E)					Band width
	1	2	4	6	8	16	2.13G	3G	1	2	4	8	16	32	<66	67-76	77-96	97-126	>127	B
1	×						×		×						×					×
2	×							×	×						×					×
3	×						×			×					×					×
4	×						×				×				×					×
5		×					×			×						×				×
6		×						×		×						×				×
7		×					×				×					×				×
8			×				×				×					×				×
9			×					×			×						×			×
10			×				×					×					×			×
11			×				×						×				×			×
12			×					×					×					×		×
13				×			×				×						×			×
14				×			×					×					×			×
15				×				×				×							×	×
16					×		×					×						×		×
17					×		×						×					×		×
18					×			×					×						×	×
19						×	×						×						×	×
20						×	×							×					×	×

Fig. 10. Constructed Formal Context of VMs Dataset

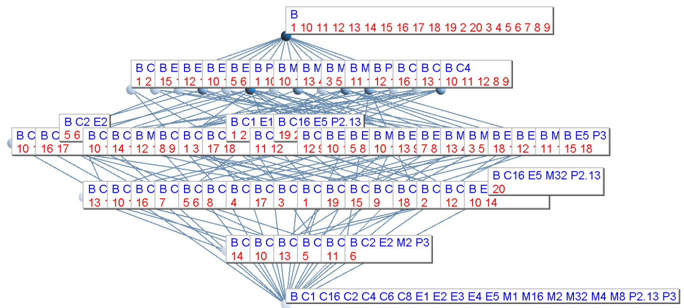

Fig. 11. Concept Lattice of VMs Dataset

respectively, and their corresponding energy consumption are denoted as $e_1, e_2, \cdots, e_m$, the weighted average energy consumption (WAE) can be represented as

$$WAE = \sum_{i=1}^{n} sim((t_i^{\uparrow\downarrow}, t_i^{\uparrow}), (v_i^{\uparrow\downarrow}, v_i^{\uparrow}))e_i \quad (17)$$

Figure 14 shows the performance comparison for VMs scheduling approaches with/without consideration of energy consumption in MEC environment. Clearly, the proposed approach in this paper can significantly reduce the energy consumption around 28% comparing to the approach without consideration of energy consumption. Hence, it is proved that the proposed approach is a green and sustainable scheduling strategy for VMs scheduling in MEC.

7 Conclusions

Aiming to address the problem of VMs scheduling in MEC, this paper firstly established a scheduling model which can be represented with a mapping from VMs to tasks. Due to the excellent features of describing the relations between objects and attributes in FCA, this paper thus constructed the formal contexts of VMs profile as well as tasks descriptions. In order to achieve the process of infusion/matching from VMs to tasks, a similarity measurement between the formal concepts that are generated from the formal context of VMs and tasks, is formally defined. Therefore, the VMs scheduling is regarded as an optimization problem which is to return the appropriate mapping from VMs to tasks in terms of maximum similarity between their concepts. The extensive simulations on real dataset are conducted for demonstrating the feasibility and effectiveness of our

	CPU (C)						Processing Ability(P)		Memory Capacity(M)						Energy Consumption(E)					Band width
	1	2	4	6	8	16	2.13G	3G	1	2	4	8	16	32	<66	67-76	77-96	97-126	>127	B
t1			×				×		×									×		×
t2				×				×	×								×			×
t3						×	×			×								×		×

Fig. 12. Constructed Formal Context of User's Request

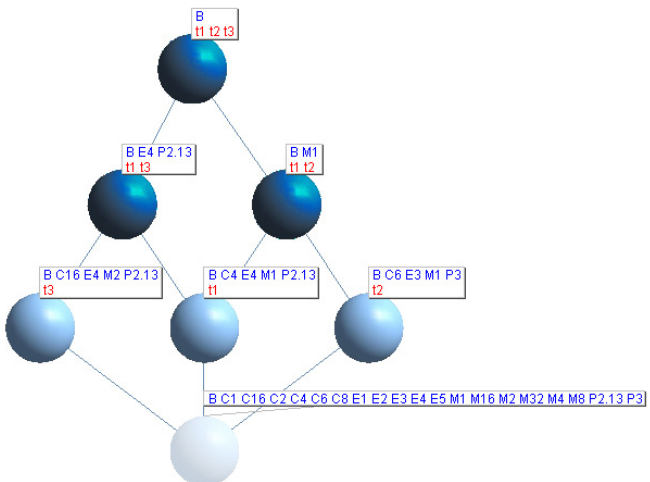

Fig. 13. Concept Lattice of User's Request

approach. It is believed that our approach, a novel kind of VMs scheduling mechanisms for MEC, can be applied into other relevant applications, such as tasks allocation for crowdsourcing, VMs scheduling in data centers and so forth.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant No. 61702317, 61872228) and was also supported by the Fundamental Research Funds for the Central Universities, China (No.GK201801004, GK201802013), and Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shaanxi Province (Grant No. 2017024).

TABLE 8
Similarity degree between VMs and User's Request

VM \ Task	t_1	t_2	t_3
v_1	$\frac{3}{7}$	$\frac{2}{8}$	$\frac{2}{8}$
v_2	$\frac{2}{8}$	$\frac{3}{7}$	$\frac{1}{9}$
v_3	$\frac{2}{8}$	$\frac{1}{9}$	$\frac{3}{7}$
v_4	$\frac{2}{8}$	$\frac{1}{9}$	$\frac{2}{8}$
v_5	$\frac{2}{8}$	$\frac{1}{9}$	$\frac{3}{7}$
v_6	$\frac{1}{9}$	$\frac{2}{8}$	$\frac{2}{8}$
v_7	$\frac{2}{8}$	$\frac{1}{9}$	$\frac{2}{8}$
v_8	$\frac{3}{7}$	$\frac{1}{9}$	$\frac{2}{8}$
v_9	$\frac{2}{8}$	$\frac{3}{7}$	$\frac{1}{9}$
v_{10}	$\frac{3}{7}$	$\frac{2}{8}$	$\frac{2}{8}$
v_{11}	$\frac{3}{7}$	$\frac{2}{8}$	$\frac{2}{8}$
v_{12}	$\frac{3}{7}$	$\frac{2}{8}$	$\frac{2}{8}$
v_{13}	$\frac{2}{8}$	$\frac{3}{7}$	$\frac{2}{8}$
v_{14}	$\frac{2}{8}$	$\frac{3}{7}$	$\frac{2}{8}$
v_{15}	$\frac{1}{9}$	$\frac{3}{7}$	$\frac{1}{9}$
v_{16}	$\frac{3}{7}$	$\frac{1}{9}$	$\frac{3}{7}$
v_{17}	$\frac{3}{7}$	$\frac{1}{9}$	$\frac{3}{7}$
v_{18}	$\frac{1}{9}$	$\frac{2}{8}$	$\frac{1}{9}$
v_{19}	$\frac{3}{7}$	$\frac{1}{9}$	$\frac{3}{7}$
v_{20}	$\frac{3}{7}$	$\frac{1}{9}$	$\frac{3}{7}$

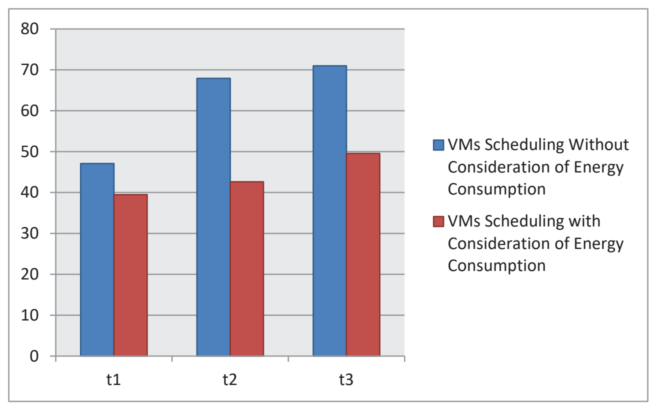

Fig. 14. The performance comparison for VMs scheduling approaches with/without consideration of energy consumption in MEC environment

References

[1] European Telecommunications Standards Institute (ETSI), "Mobile-edge computing-Introductory technical white paper", 2014.
[2] F. Hao, D.S. Park, J. Kang, G. Min: "$2L$-MC^3: A two-layer multi-community-cloud/cloudlet social collaborative paradigm for mobile edge computing, IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2018.2867351, 2018.
[3] Z. Zhao, G. Min, W. Gao, Y. Wu, H. Duan, and Q. Ni: "Deploying edge computing nodes for large-scale IoT: A diversity aware approach", IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2018.2823498, 2018.
[4] K. Kumar and Y. H. Lu: "Cloud computing for mobile users: Can offloading computation save energy?", Computer, Vol.43, No.4, pp.51-56, 2010.
[5] X. Li, P. Garraghan, X. Jiang, Z. Wu, J. Xu: "Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy", IEEE Transactions on Parallel and Distributed Systems, Vol.29, No.6, pp.1317-1331, 2018.
[6] C. Huang, G. Min, Y. Wu, Y. Ying, K. Pei, and Z. Xiang: "Time series anomaly detection for trustworthy services in cloud computing systems," IEEE Transactions on Big Data, DOI: 10.1109/TBDATA.2017.2711039, 2018.
[7] T. Wang, Z. Liu, Y. Chen, Y. Xu, X. Dai: "Load balancing task scheduling based on genetic algorithm in cloud computing", In Proceedings of IEEE International Conference on Dependable, Autonomic and Secure Computing, pp.146-152, 2014.
[8] S. Hosseinimotlagh, F, Khunjush, R. Samadzadeh: "SEATS: smart energy-aware task scheduling in real-time cloud computing", The Journal of Supercomputing, Vol.71, No.1, pp.45C66, 2015.
[9] F. Juarez, J. Ejarque, R.M. Badia: "Dynamic energy-aware scheduling for parallel task-based application in cloud computing", Future Generation Computer Systems, Vol.78, No.1, pp.257-271, 2018.
[10] S. Hafiz, H.X. Tan, A. Ishfaq, R. Sanjay: "Energy and performance-aware scheduling of tasks on parallel and distributed systems", ACM Journal on Emerging Technologies in Computing System, Vol.4, No.8, pp.1-37, 2013.
[11] Z. Zheng, R. Wang: "An approach for cloud resource scheduling based on parallel algorithm", In Proceedings of 3rd International Conference on Computer Research and Development, pp.444-447, 2011.
[12] J. Kang, S. Ranka: "Dynamic slack allocation algorithms for energy minimization on parallel machines", Journal of Parallel and Distributed Computing, Vol.70, No.5, pp.417-430, 2012.
[13] X. Ruan, X. Qin, Z. Zong: "A prediction scheduling algorithm using dynamic voltage scaling for parallel applications on clusters", In Proceedings of 16th Conference on Computer Communications and Networks, pp.535-541, 2010.

[14] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li: "Channel state information prediction for 5G wireless communications: A deep learning approach", IEEE Transactions on Network Science and Engineering, vol. 99, no. PP, pp. 1-1, Jun. 2018

[15] W. Wang, G. Zeng, D. Tang: "Cloud-DLS: Dynamic trusted scheduling for cloud computing", Expert Systems with Applications, Vol.39, No.3, pp.2321-2329, 2013.

[16] Z. Hu, K. Wu, J. Huang: "An utility-based job scheduling algorithm for current computing cloud considering reliability factor", In Proceedings of 3rd International Conference on Software Engineering and Service Science, pp.296-299, 2012.

[17] D. Kusic, J. Kephart, J. Hanson, et.al: "Power and performance management of virtualized computing environments via lookahead control", Cluster Computing, Vol.12, No.1, pp.1-15, 2013.

[18] P. Owezarski, J. Lobo, D. Medhi: "Network and service management for cloud computing and data centers", Journal of network and systems management, Vol.21, No.4, pp.707-712, 2013.

[19] C. Xu, X. Ma, R. Shea, H. Wang, J. Liu: "Enhancing performance and energy efficiency for hybrid workloads in virtualized cloud environment", IEEE Transactions on Cloud Computing, Vol.PP, No.99, pp.1-1, 2018.

[20] J. Wei, X. Zeng: "Optimal computing resource allocation algorithm in cloud computing based on hybrid differential parallel scheduling", Cluster Computing, No.8, pp.1-7, 2018.

[21] B. Ganter: "Formal Concept Analysis: mathematical foundations", Springer-Verlag New York, Inc. 1999.

[22] Z. Brahmi and M. Gammoudi: "Hierarchical organization of agents based on Galois sub-hierarchy for complex tasks allocation in massive MAS", In 3rd International KES Uppsala, Sweden, Lecture Notes In Artificial Intelligence, Vol.5559, pp.460, 2009.

[23] F. Hao, G. Min, Z. Pei, D.S. Park, L.T. Yang: "k-clique community detection in social networks based on formal concept analysis", IEEE Systems Journal, Vol.11, No.1, pp.250-259, 2017.

[24] F. Hao, M. Kodialam, T.V. Lakshman, S. Mukherjee: "Online allocation of virtual machines in a distributed cloud", IEEE/ACM Transactions on Networking, Vol.25, No.1, pp.238-249, 2017.

[25] B.M. Tudor, Y.M. Teo: "On understanding the energy consumption of ARM-based multicore servers", SIGMETRICS Perform. Eval. Rev., Vol.41, No.1, pp. 267-278, 2013.

[26] W. Lin, W. Wu, H. Wang, J. Wang, C. Hsu: "Experimental and auantitative analysis of server power model for cloud data centers", Future Generation Computer Systems, Vol.86, pp.940-950, 2018.

[27] F. Hao, D.S. Sim, D.S. Park, H.S. Seo: "Similarity evaluation between graphs: A formal concept analysis approach", Journal of Information Processing Systems, Vol.13, No.5, pp.1158-1167, 2017.

[28] L. Minas, B. Ellison: Energy efficiency for information technology: How to reduce power consumption in servers and data centers, Intel Press, 2009.

Fei Hao received the B.Sc. degree in Information and Computing Science and the M.Sc. degree in Computer Software and Theory from Xihua University, China, in 2005 and 2008, respectively, and the Ph.D. degree in Computer Science and Engineering from Soonchunhyang University, South Korea, in 2016. He is currently an associate professor with the School of Computer Science, Shaanxi Normal University, China. He has authored over 80 papers in international journals and conferences. He received five best paper awards from KISM 2012, GreenCom 2013, MUE 2015, UCAWSN 2015, and CUTE 2016. He is a recipient of the IEEE Outstanding Leadership Award at CPSCom 2013, the 2015 Chinese Government Award for Outstanding Self-Financed Students Abroad and the IEEE Outstanding Service Awards at SmartData 2017 and DSS 2018. His research interests include social computing, ubiquitous computing, big data analysis and processing and mobile cloud computing. He is a member of ACM and KIPS.

Guangyao Pang received the M.S. degree in software engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2013. He is currently a Ph.D. candidate in School of computer science, Shaanxi Normal University, Xi'an, China. His main research interests include big data, deep learning, and recommendation system.

Zheng Pei received the M.S. and Ph.D. degrees from Southwest Jiaotong University, Chengdu, China, in 1999 and 2002, respectively. He is currently a Professor with the School of Computer and Software Engineering, Xihua University, Chengdu. His research interests are rough set theory, fuzzy set theory, logical reasoning, and linguistic information processing. He has nearly 70 research papers published in academic journals or conference.

Keyun Qin received the M.Sc. and Ph.D. degrees in Applied Mathematics, both from Southwest Jiaotong University, China, in 1994 and 1997, respectively. Currently, he is a Full Professor of College of Mathematics at Southwest Jiaotong University. His current research interests are rough set theory, soft set theory, fuzzy logic based systems and formal concept analysis. He has nearly 80 research papers published in academic journals or conference.

Yu Zhang received the B.Sc. and M.Sc. degrees in Applied Mathematics from Xi'an Jiaotong University(XJTU) in 2006, 2009 respectively. the Ph.D. degree in Information and Mathematics from Ecole Centrale de Lyon, France, in 2014. He currently works as a lecturer in Shaanxi Normal University, China. His research interests include computer vision, pattern recognition and deep learning, with focus on image annotation, image super-resolution, and object detection.

Xiaoming Wang is a professor at the School of Computer Science in Shaanxi Normal University, China. He received his Ph.D. degree in computer software and theory from Northwest University, China, in 2005. He was a visiting scholar at the Department of Computer Science in Georgia State University, USA, from 2007 to 2008. He has served as a reviewer for many journal papers. He has published 3 books and more than 100 papers. His main research interests include delay tolerant networks, network security, mobile sensing network and social computing.