A Resource Usage Prediction-Based Energy-Aware Scheduling Algorithm for Instance-Intensive Cloud Workflows

Zhibin Wang[1,2], Yiping Wen[1,2(✉)], Yu Zhang[1,2], Jinjun Chen[1,3], and Buqing Cao[1,2]

[1] School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China
ypwen81@gmail.com, Jinjun.Chen@gmail.com, cao6990050@163.com
[2] Key Laboratory of Knowledge Processing and Networked Manufacturing, Hunan University of Science and Technology, Xiangtan, China
[3] Swinburne Data Science Research Institute, Swinburne University of Technology, Melbourne, Australia

Abstract. The applications of instance-intensive workflow are widely used in e-commerce, advanced manufacturing, etc. However, existing studies normally do not consider the problem of reducing energy consumption by utilizing the characters of instance-intensive workflow applications. This paper presents a resource usage Prediction-based Energy-Aware scheduling algorithm, named PEA. Technically, this method improves the energy efficiency of instance-intensive cloud workflow by predicting resources utilization and the strategies of batch processing and load balancing. The efficiency and effectiveness of the proposed algorithm are validated by extensive experiments.

Keywords: Energy · Instance-intensive · Scheduling · Cloud workflow · Batch processing · Prediction

1 Introduction

With the development of cloud computing and software technology, workflow applications in the field of science and engineering have grown steadily in variety and scale. As a typical application type in cloud computing environment, instance intensive cloud workflow usually exists in electronic commerce, advanced manufacturing and other fields. Unlike complex scientific workflow, instance intensive workflow has a large number of potential and relatively simple concurrent instances, in which instance is a single execution event of workflow at a particular time. For example, while processing bank cheques, millions of cheques are processed simultaneously every day, and each check transaction is a fairly simple workflow that takes only a few steps to complete.

Workflow scheduling is the key to managing workflow execution efficiency. In order to achieve high execution efficiency, the performance of instance intensive cloud workflow needs to achieve high throughput, high load balance and high resource utilization. High throughput allows a large number of workflow events to start and

H. Gao et al. (Eds.): CollaborateCom 2018, LNICST 268, pp. 626–642, 2019.
https://doi.org/10.1007/978-3-030-12981-1_44

complete in a set of time periods. Load balancing allows for balancing requests from users and providing users with better quality of service. Resource utilization can effectively manage resources. Up to now, several instance-intensive cloud workflow scheduling algorithms have been proposed. Unfortunately, most studies do not take energy into account, as energy has become one of the main problems of clouds and has received increasing attention due to environmental and financial considerations [17].

On the other hand, several researches have shown that predicting the future of users' demands and cloud resource usage can be applied to handle the energy problem [3]. Besides, it takes considerable time for configuring the virtual machines (VMs) according to the requirements of the application in cloud computing platforms [24]. If the future requirements of resources can be predicted beforehand, a large number of instance-intensive clouds workflow instances can be processed by the already configured VMs, and a high utilization of cloud resources can be maintained to reduce the energy consumption. That is, extra time to launch and configure new VMs can be saved, by which higher throughput may be achieved. Extra energy consumption caused by low utilization of cloud resources can also be reduced.

In addition, the combination of scheduling and batch decision can improve the efficiency of batch jobs instead of dealing with them alone [18]. Methods and prototype systems that support automatic batch execution in workflows have been proposed in [5, 6, 19]. However, there are few researches on workflow scheduling considering batch execution of several workflow instances to reduce energy consumption. For instance-intensive workflows, it does not have a dedicated energy-aware scheduling algorithm. Therefore, the resource prediction-based energy-aware scheduling method for developing instance-intensive cloud workflows has practical significance.

With these observations, this paper presents a resource usage Prediction-based Energy-Aware scheduling algorithm for instance-intensive cloud workflows, named PEA. It uses the technology of predicting resources utilization and the strategies of batch processing and load balancing to reduce energy consumption. Our contributions are three folds. Firstly, a formal concept of combining scheduling with batch processing and resource utilization prediction is proposed. Secondly, a scheduling method based on resource utilization prediction is designed for instance-intensive cloud workflow. Finally, comprehensive experiments and simulations are conducted to demonstrate the validity of the proposed method.

2 System Models and Architecture

Our problem consists in scheduling the instance-intensive workflow meeting the specified makespan in such a way that the energy consumption are minimized. In this section, we describe the energy model and system architecture underneath our approach. For ease of understanding, we summarize the major notations and their meanings used throughout of this paper in Table 1.

Table 1. Key notations and descriptions

Notation	Description
w_i	The i-th instance-intensive workflow
t_{ij}	The j-th workflow activity of wi_i
t_{ijk}	The k-th instance of the workflow activity t_{ij}
p_m	The m-th physical machine
c_m	The calculate ability of p_m
vm_n	The n-th type of virtual machine (VM)
vc_n	The calculate ability of vm_n
r_m	The number of VMs on p_m at time t
$U_m(t)$	The CPU utilization of p_m at time t
φ_m	The basic energy consumption rate of p_m

2.1 Concepts and Definitions

For better introducing PEA and related concepts, we first give some definitions which will be used later.

Definition 1 (Instance-Intensive Workflow). Instance-Intensive Workflow can be defined as $W = \{w_1, w_2, \ldots, w_I\}$, where I is the number of Instance-Intensive workflow, and w_i could be expressed as $w_i = <T_i, E_i, OD_i>$, where

(1) T_i is a set of workflow activities, and a workflow activity could be expressed as $t_{ij} = <ID_{ij}, Type_{ij}, W_{ij}, SD_{ij}, GC_{ij}>$, where ID_{ij} is the activity number, which is unique, $Type_{ij}$ is the execution type of the activity, which can be divided into two types: normal activity and batch-processing activity, W_{ij} is calculation workload, SD_{ij} is the deadline for this activity and GC_{ij} stands for the grouping characteristics values of t_{ij}. $t_{ij} \in T_i, T_i = \{t_{i1}, t_{i2}, \ldots, t_{iJ}\}$, where J is the number of workflow activities.
(2) E_i is the set of directed edges between the workflow activities to represent dependency, which can be expressed as $E_i = \{<t_{ia}, t_{ib}> \mid <t_{ia}, t_{jb}> \in T \times T\}$.
(3) OD_i is the time constrained (i.e. overall deadline).

Definition 2 (Activity Instance). t_{ijk} is the k-th instance of the workflow activity t_{ij}, $1 \leq k \leq K$, where K is the number of instances of j-th activity of i-th workflow.

Definition 3 (Cloud Resource). $P = \{p_1, p_2, \ldots, p_M\}$ is a set of physical machines (PMs), where M is the number of PMs. The PMs provide the hardware infrastructure for creating virtualized resources to meet service demands. $VM = \{vm_1, vm_2, \ldots, vm_N\}$ is the set of N types of virtual machine (VM). A VM type vm_n is specified by the characteristic of computing performance vc_n in million instructions per second.

2.2 Energy Model

The energy consumption for PMs is composed of CPU, memory, disk, power supply, etc. Various research results in the literature show that CPU utilization significantly affects energy consumption [1, 2, 4]. Therefore, this study focuses on the influence of CPU utilization on the overall energy consumption of the system and ignores the influence of other components.

Hsu et al. [4] proposed a simplified energy consumption model of PM. The energy consumption rate at time instant t for the p_m, denoted as PE_m, which can be calculated by Eq. 1.

Where φ_{m} is the basic energy consumption rate of p_m, $U_m(t)$ is CPU utilization rate of p_m at time instant t and α_m is a constant, which could be calculated as $\alpha_m = \varphi_m/7$.

$$PE_m = \begin{cases} \varphi_m & \text{if} \quad U_m(t) = 0 \\ \varphi_m + \alpha_m & \text{if} \quad 0 < U_m(t) \leq 0.2 \\ \varphi_m + 3\alpha_m & \text{if} \quad 0.2 < U_m(t) \leq 0.5 \\ \varphi_m + 5\alpha_m & \text{if} \quad 0.5 < U_m(t) \leq 0.7 \\ \varphi_m + 8\alpha_m & \text{if} \quad 0.7 < U_m(t) \leq 0.8 \\ \varphi_m + 11\alpha_m & \text{if} \quad 0.8 < U_m(t) \leq 0.9 \\ \varphi_m + 12\alpha_m & \text{if} \quad 0.9 < U_m(t) \leq 1 \end{cases} \tag{1}$$

We assume that CPU utilization is the ratio of resources required by the VMs to the PM (Mainly for computing resources). Therefor, $U_m(t)$ could be calculated by Eq. 2.

$$U_m(t) = \sum_{n=1}^{N} \sum_{l=1}^{r_{n,l}} vc_{n,l}/c_m \tag{2}$$

Where c_m is the computing performance of p_m, $r_{n,l}$ represents the number of VMs of type vm_n running on the p_m at time t and $vc_{n,l}$ is the computing performance of vm_n.

Integral to PE_m can be obtained the total energy consumption of p_m, denoted as E_m, could be calculated by Eq. 3.

$$E_m = \int_{t \in \omega_m} PE_m(U_m(t))dt \tag{3}$$

Where ω_{m} is the total operating time of p_m.

2.3 System Architecture

The scheduling architecture of PEA is shown in Fig. 1. It consists of three layers: user layer, scheduling layer and resource layer. The scheduler consists of batch processing, resource monitor, resource predictor and energy-aware resource allocator. Batch processing is used to merge some activity instances to generate an activity execution instance. Resource monitor detects the resource usage of PMs and updates the CPU utilization information of each PM. Resource predictor predicts subsequent resource usage by using resource usage information obtained from the resource monitor and controls the opening and closing of the resource to avoid the invalid waste of the

energy. Energy-aware resource allocator allocates appropriate resources to execute the instance through the strategy of load balancing.

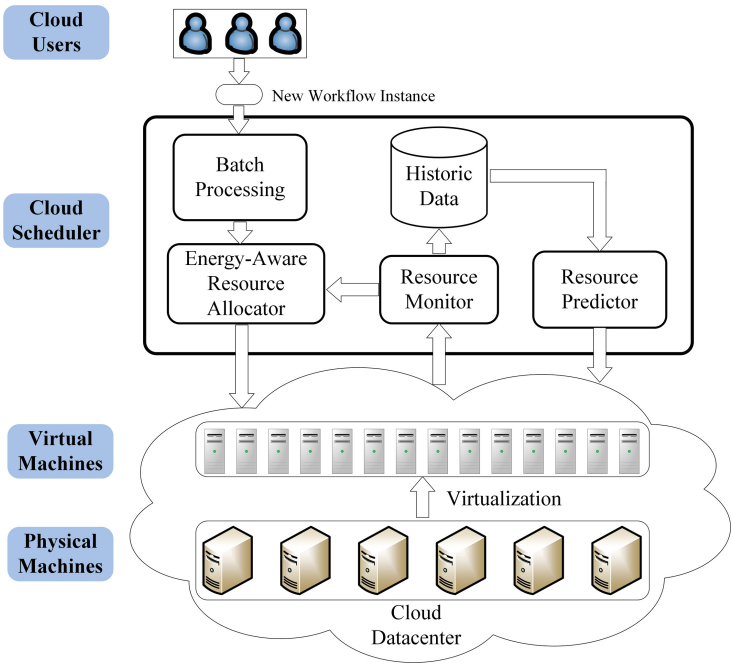

Fig. 1. The scheduling architecture of PEA

The overview of the scheduling process for this architecture is as follows. When new active instances arrive, they first enter the batch processing module, and some of instances are merged into new instances. On the other hand, the resource predictor prepares the resources needed by the instance in advance. Finally, energy-aware resource allocator put these instances into the resources prepared in advance.

The main characteristics of PEA include three points: batch strategy, resource prediction and load balancing. The benefits of this algorithm are summarized below.

- The strategy of batch processing could reduce the energy waste caused by computation processing and repeated transmission of data.
- The technology of resource prediction can adapt to the rapid increase of instance-intensive workflow. Reduce energy consumption caused by the untimely resource allocation and closure.
- The strategy of load balancing allows open PMs to operate at lower energy rates, which reduces the overall energy consumption.

3 Algorithm Implementation

The proposed PEA algorithm is capable of reducing energy consumption while meeting the makespan. It consists of four parts, such as batch processing, resource monitor, resource predictor and energy-aware resource allocator. The implementation of batch processing and resource monitor have been mentioned in our previous work. Detailed introductions of the mechanism of these are referred to [27]. In this section, we will focus on the implementation of resource predictor and energy-aware resource allocator.

3.1 Resource Predictor

Instance-intensive workflow has the characteristics of short-term rapid growth. A large number of task instances require a large allocation of resources in a short period of time. These tasks are often time - constrained, and exceeding them can have a huge impact. In the stock market, for example, failure to deal with it in time can lead to huge economic consequences. Therefore, more idle PMs are usually opened in the process of traditional resource allocation. However, if we keep the resources in the idle state for a long time, this will be extremely energy consuming. We use the method of resource usage prediction to allocation and recovery the PM in advance, so as to achieve the purpose of energy saving.

The model of resource predictor involves a number of steps as shown in Fig. 2. In preprocessing step, first, preprocess the historical data, which consists of three steps: feature selection, construction sequence and data partitioning. Then we use the processed historical data to train the DBN to get the model. Finally, we input the current data into the model to get the forecast information of the future resource utilization to guide the allocation or recovery of the resources.

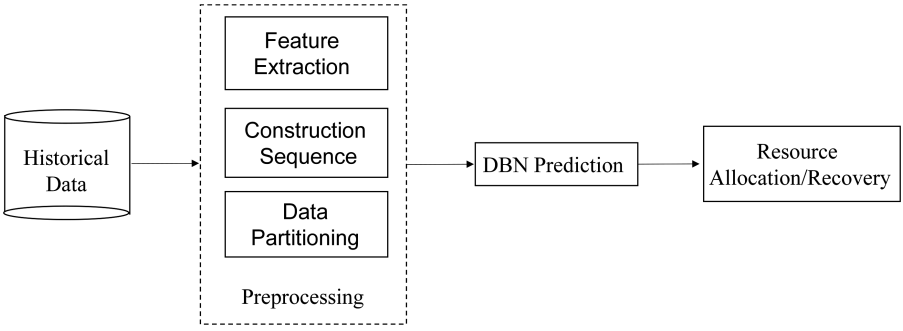

Fig. 2. The structure of resource predictor

Algorithm 1 specifies the process of resource utilization prediction and pre-opened on the physical machine according to the conditions.

Algorithm 1. Resource Prediction Algorithm RPA

Input: The set of cloud Resource history D.

Output: the information of Pre-opened PM.

01: Initialize $Weight_{ij}$ of the predictive model

02: set $W_{ij} \leftarrow Weight_{ij}$

03: select feature F in D

04: for i ← length(F)

05: Δt time serialize (Fi)

06: foreach training model **do**

07: Update ΔW_{ij}

08: $W_{ij} = W_{ij} + \Delta W_{ij}$

09: P_{t+1}=prediction model (W_{ij})

10: if P_{t+1} exceeds the set threshold

11: Pre-open PM

12: return the information of Pre-opened PM

As shown in Algorithm 1, it describes process of forecasting in resource predictor. The input of model is a historical dataset of the data center in the cloud computing environment and the output is resource utilization value in the future. Firstly, model initializes and sets weights (lines 1–2). Then, feature selection has been done (line 3). Time serialization of feature data at Δt time granularity (lines 4–6). Training the prediction model and updating the weight of the model (lines 7–9). Finally, Pre-open the physical machine according to conditions (lines 10–11).

Preprocessing of Resource Predictor. Because the original data in the historical data set is huge and has many features, the direct use of the model will cause the problems such as large error and poor interpretability. Therefore, a series of data processing processes are carried out to improve the accuracy of prediction results. The main steps of data preprocessing include feature extraction, reconstruction sequence and data partition.

Feature Extraction. The resources in cloud computing environment mainly include CPU utilization, content utilization, disk and bandwidth. According to different cloud resource forecast requirements, the selected characteristics are different. In the stage of data analysis, the early analysis of feature selection is done. This paper mainly focuses on the prediction of CPU utilization ratio, so the historical CPU utilization factor is used as input feature in feature selection process.

Construction Sequence. Owing to the high correlation of host load data in the adjacent time interval, [24] proposes to use host load monitoring tool to record workload data

into one-dimensional time series for predictive network training and good prediction results can be achieved. So this paper synthetically considers the practical significance of model input and prediction effect, and formats it according to the present time interval from the selection feature (in this paper, the granularity is per hour, the time interval is set length according to the specific problem). The time series of cloud resources are constructed and the time series data are taken as the input parameters of the depth confidence network in order to improve the prediction accuracy.

Data Partitioning. The prediction model is a model that describes the mapping relationship between the historical data feature F_t and the first period prediction value Y_{t+1}, which can be expressed as $F_t \rightarrow Y_{t+1}$. The purpose of data partition is to select the appropriate t value so as to predict cloud resources better.

DBN Prediction. In [20], they used an unsupervised learning model, the deep belief network (DBN). The structure of DBN can be described as two layers. The lower layer is composed of multiple restricted Boltzmann machines (RBM). The upper layer is a BP neural network. The training process of the deep belief network is divided into two parts, the unsupervised learning part corresponding to the RBM network in the lower layer and the supervised learning part corresponding to the fine-tuning of the BP neural network in the upper layer. We use this method to predict the utilization rate of cloud resources. The specific process is shown in the Fig. 3.

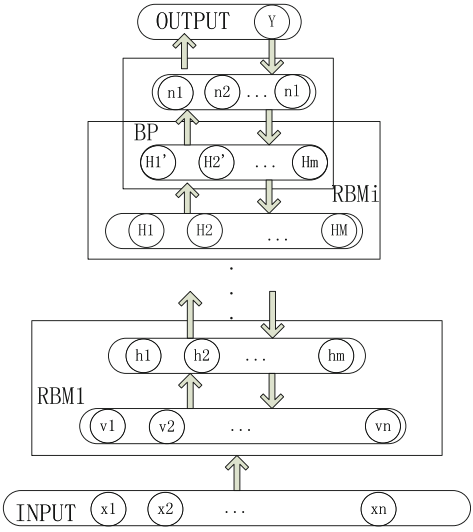

Fig. 3. The structure of DBN

3.2 Energy-Aware Resource Allocator

The main idea of energy-aware resource allocation is to balance the resource usage of open physical machines and limit the resource utilization of each host to a specified threshold. The specific steps can be described as follows:

(1) Specify a suitable VM type for all instances. This VM type should be the minimum VM type that guarantees that the instance can be completed within the deadline.
(2) The current CPU utilization of the PMs obtained in the resource monitor is sorted in ascending order.
(3) The VM type obtained from (1) is mapped to the sorted PM one by one until the CPU utilization of PM after VM loading is lower than or equal to the threshold value.
(4) If no PM meets the requirements, then assign the instance to a new PM that has been pre-opened.
(5) Return (2) until all instances are assigned resources.

Algorithm 2 Resource Allocation Based on CPU Utilization RA

Input: The current resource usage CRU
the pending instances PI
Output: The final resource allocation policies
01: for each PI [i] is not empty
02: Assign a suitable vm_k to a PI [i]
03: DPM←sort(PMs) by the CPU utilization in CRU
04: for each PI [i] is not empty
05: **for** i=1 to M
06: **if** DPM[i] allocate to PI [i],it does not exceed the threshold
07: PI [i] ←DPM[i]
08: if PIQ[i] did not get resources
09: PI [i] ←Pre-opened PM

Algorithm 2 specifies the process of physical machine resource allocation. It is to show the steps mentioned above in the form of pseudo-code. The appropriate virtual machine types is arranged for each instance (lines 1–2). Sort current resources based on CRU (line 3). These sorted resources are in turn attempted to map virtual machine type. If the CPU threshold is met, schedule the current resource to calculate the instance (lines 4–7). If no resource satisfies the condition, this instance is calculated from the pre-opened resource (lines 8–9).

4 Experimental Evaluation

In this section, we conducted a series of comprehensive experiments to evaluate the performance of our proposed PEA. For comparative analysis, we performed comparison experiments with other methods to verify the effectiveness of our proposed PEA method.

4.1 Experimental Context

In this experiment, VM scheduling in the cloud environment will be simulated and tested by cloud simulator CloudSim [26]. The hardware environment used in the experiment is Intel(R) Core(TM) i5-6500 CPU @3.2 GHz, 8 GB memory. The software environment is Eclipse3.5 for Windows7, and cloudsim-3.0.3 is configured to complete the experiment.

In the process of verifying the proposed method, three types of physical machines of different specifications were mainly used to construct the cloud simulation environment. The specific configuration of the physical machine and related energy consumption settings are shown in Table 2. Set the basic power consumptions of the HP ProLiant ML110 G4 and HP ProLiant ML110 G5 to 86 W and 93.7 W based on the energy consumption values of them. Then based on the operating energy specifications for the single-processor HP ProLiant BL460c G6 in the HP white paper, the basic power consumption is set to 192 W.

Table 2. Parameter settings

Physical machine hardware configuration	Basic energy consumption (W)
HP ProLiant ML110 G4 (Intel Xeon 3040, dual-Processor clocked at 1860 MHz, 4 GB of RAM)	86
HP ProLiant ML110 G5 (Intel Xeon 3075, dual-Processor clocked at 2660 MHz, 4 GB of RAM)	93.7
HP ProLiant SL390s G7 (Intel Xeon 5649, dual-Processor clocked at 3060 MHz, 16 GB of RAM)	192

4.2 Performance Evaluation

In this section, we have carried out resource utilization prediction comparison experiments and scheduling method comparison experiments to verify our proposed method.

Resource Usage Prediction Comparison Experiment. In order to get closer to the complexity of the cloud computing environment, this article uses the real data of Google Cloud Data Center for simulation and verification. In May 2011, Google publicly released a 29-day historical dataset and documentation from the data center that detailed the semantics, formats, and patterns. This workload consists of more than 12,000 heterogeneous physical hosts running 4,000 different types of applications and a large amount of data for approximately 1.2 billion rows of resource usage data. We

have selected the information of 40 physics machines for 29 days to carry on the experiment.

In this resource usage prediction experiment, we compared other algorithms such as back propagation (BP), support vector regression (SVR), radial basic function (RBF) and multivariable linear regression (MLR). The evaluation indicators selected are commonly used indicators in the prediction model, including mean absolute error (MAE), mean squared error (MSE), and mean absolute percentage error (MAPE). The specific formula for the evaluation index selected is as follows

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|yt_i - yp_i| \tag{4}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(yt_i - yp_i)^2 \tag{5}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{yt_i - yp_i}{yt_i}\right| \tag{6}$$

Where N is the total number of test sets, yt_i is the actual value, and yp_i is the output value of the network. MAE is the average absolute of the actual value and the predicted value. Compared with the average error, the average absolute error is absolutized because of the deviation, and there is no positive and negative offset. Therefore, the average absolute error can better reflect the actual situation of the predicted value error. MSE is very sensitive to a set of very large or very small errors in measurement. It can well reflect the precision of measurement. MAPE uses the same unit dimension to reflect the extent of the deviation of the measurement result from the true value. They evaluate the results from different angles, and the smaller the value, the higher the prediction accuracy (Figs. 4, 5 and 6).

The intraday average of the three evaluation indicators is shown in Table 3. The experimental results show that the DBN prediction method proposed in this paper is the best compared with the comparison method, and the DBN prediction ability is more stable with the increase of the prediction window. The main reason why the DBN method is significantly better than other prediction methods is that the features extracted by the RBM network can better reflect the complex characteristics of the entire data center workload data, and thus can stably improve the prediction effect.

Scheduling Algorithm Comparison Experiment. We conduct Comparison experiments on energy consumption with others. To reveal the advantages of our PEA algorithm in reducing energy consumption, we compare an energy-aware virtual machine scheduling [7] method is proposed, which is referred to as EVMS. The main idea of EVMS is to migrate activities to low-energy physical machines to reduce energy consumption, and then migrate some activities on low-energy physical machines to higher-energy virtual machines to speed up activity execution time. In this experiment, EVMS is used as the benchmark algorithm for comparison.

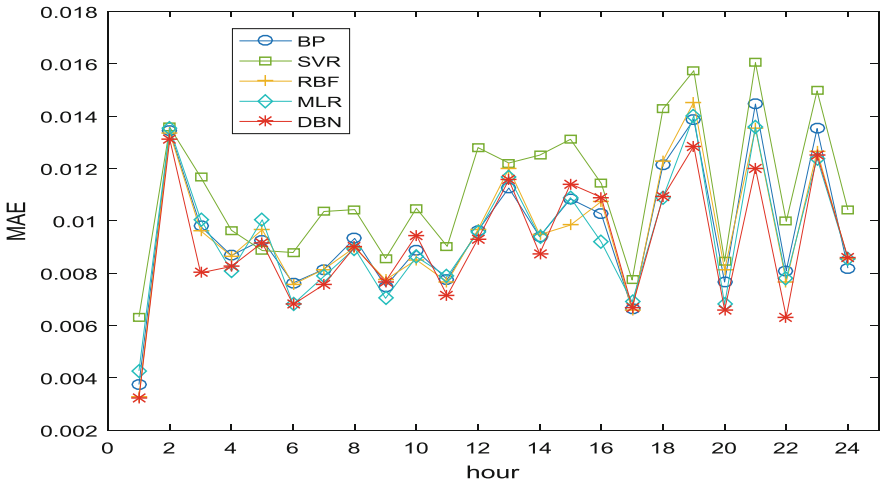

Fig. 4. Performance of resource usage prediction experiment in MAE

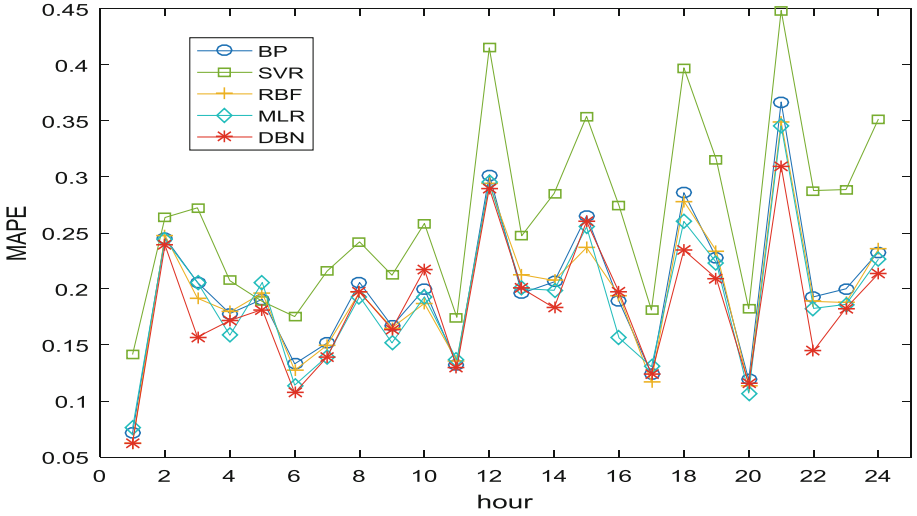

Fig. 5. Performance of resource usage prediction experiment in MAPE

Figure 7 plots the energy consumption results of three algorithms and Table 4 shows the ratio of PEA to the energy consumption of EVMS. From the results we can see that our PEA algorithm has the lower energy consumption. This is because the resource utilization prediction technology in PEA algorithm and two strategies can greatly reduce energy consumption. The technology of resource utilization prediction can effectively reduce energy waste caused by untimely opening and closing of resources. Batch processing strategy can combine multiple examples to reduce the energy loss caused by repeated calculation. The energy consumption is related to the

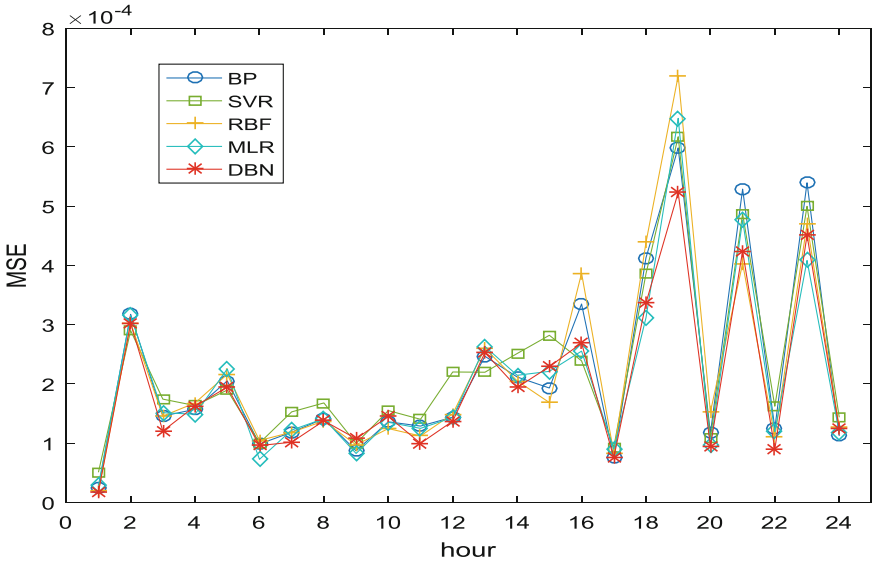

Fig. 6. Performance of resource usage prediction experiment in MSE

Table 3. Comparison of forecasting methods

	MAE	MSE	MAPE
SVR	0.0111	2.2466e−04	0.2660
BP	0.0094	2.1655e−04	0.1996
RBF	0.0095	2.1744e−04	0.1955
MLR	0.0094	2.0469e−04	0.1912
DBN	0.0089	1.9517e−04	0.1847

characteristics of activities. The load balancing strategy with CPU utilization threshold limits ensures that the system works at a low energy rate without affecting the running time.

5 Related Work

In cloud computing, any reduction in energy consumption can bring huge economic savings because the data center contains a large number of computer clusters. So there has been a lot of research on how to reduce energy consumption.

The energy consumption model of computer systems is the first problem that needs to be solved. Aliza et al. [1] lists the impact of various hardware and software on the energy consumption of computer systems. They found that the CPU is the component that has the greatest impact on system energy consumption among all components. Lien et al. [2] found that physical machine CPU utilization and energy consumption are

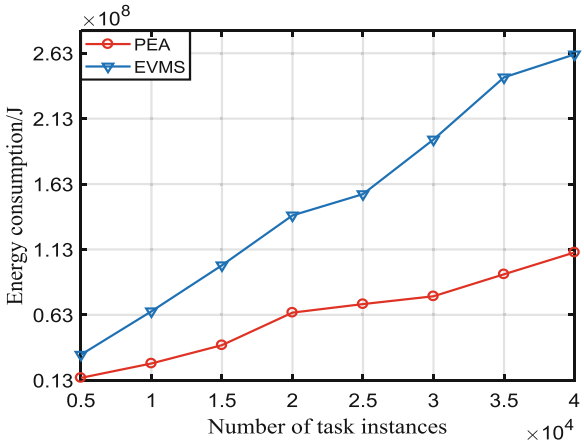

Fig. 7. Comparison experiments in energy consumption.

Table 4. Improvements of energy consumption with PEA compared to each comparison algorithm

	Number of task instances							
	5000	10000	15000	20000	25000	30000	35000	40000
EVMS	45.62%	39.34%	39.49%	46.56%	45.88%	39.19%	38.45%	42.25%

not linear. They collect data on power and CPU utilization, propose a power consumption model based on CPU utilization, and design a virtual instrument software module. Real-time measurement of the power consumption of the streaming server. These observations are used in this paper.

Some researchers use Resource forecasting method, like Kimura et al. [21] proposed a resource allocation model based on regression method, estimating the amount of resources in the virtual computing infrastructure. It predicts the number of vCPUs and the capacity of the required RAM with a nonlinear exponential regression model which allows the better selection of the configuration and the number of virtual machine and reduces the cost of small cloud providers. Ardagna et al. [22] presented workload predition model based on moving average. And a distributed solution was proposed that incorporated workload prediction and distributed non-linear optimization techniques. Roy et al. [23] proposed a forecasting model using Auto Regressive Integrated Moving Average model. A discussed the challenges involved in auto scaling in a cloud environment. Rahmanian et al. [25] proposed a learning automata-based ensemble resource usage prediction algorithm which combines state of the art prediction models in cloud computing environment. This algortithm determines the weights of each component model to achieve accurate prediction according to different situations. Faruk et al. [26] presented iOverbook, which is an autonomous, online and intelligent framework to calculate overbooking rate. It analyzes historical data of datacenter host CPU utilization and uses neural network to predict future CPU

utilization. Specifically, it predicts the average CPU utilization over the specified time interval of the physical host and then calculates the overbooking rate of the CPU on the next hour. Finally, perfoming on the data of real Google cloud computing environment, the experiment shows that iOverbook can help Cloud service providers improve their resource utilization by an average of 12.5% and save 32% power in the datacenter.

Many efficient workflow scheduling techniques for the purpose of reducing the energy consumption have been researched [9–12, 15]. Kim et al. [13] discussed an energy credit scheduler for estimating power consumption in VM based on the number of workloads performed on VM. Based on the estimation model, the scheduling algorithm of virtual environment is designed, and the resource computing task based on minimum energy consumption and minimum budget is realized, and it is implemented in Xen virtualization system. Yassa et al. [14] described the scheduling strategy of DVFS based particle swarm optimization (PSO) algorithm for practical and scientific workloads. To reduce power consumption by using different levels of voltage to supply the workload by sacrificing clock frequency. This multiple voltage involves a tradeoff between the mass and energy of the schedule. The main disadvantages of evolutionary algorithms are slow convergence and long computation time, which are not suitable for instance intensive workflow.

6 Conclusion

In this paper, we present a resource usage Prediction-based Energy-Aware scheduling algorithm, named PEA. The method is promoted with strategies to merge several activity instances, predict the resource usage and balance resource utilization of physical machines to improve energy efficiency in instance-intensive cloud workflows. For processing instance-intensive workflows, our goal is to reduce the overall energy consumption of the system as much as possible under the premise of the activity deadline. Experimental evaluations have been performed to verify the efficiency and effectiveness of our proposed method.

Acknowledgment. This paper was supported by National Natural Science Fund of China (No. 61772193, 61402167, 61702181, 61572187, 61873316 and 61872139), Innovation Platform Open Foundation of Hunan Provincial Education Department of China (No. 17K033), Hunan Provincial Natural Science Foundation of China (No. 2017JJ2139, 2017JJ4036, 2016JJ2056 and 2017JJ2098), and the Key projects of Research Fund in Hunan Provincial Education Department of China (No. 15A064).

References

1. Alizai, M.H., Kunz, G., Landsiedel, O., Wehrle, K.: Promoting power to a first class metric in network simulations. In: International Conference on Architecture of Computing Systems, pp. 1–6 (2010)
2. Lien, C.-H., Liu, M.F., Bai, Y.-W., Lin, C.H., Lin, M.-B.: Measurement by the software design for the power consumption of streaming media servers. In: IEEE Instrumentation and Measurement Technology Conference Proceedings, pp. 1597–1602 (2006)

3. Rahmanian, A.A., Ghobaei-Arani, M., Tofighy, S.: A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment. Future Gener. Comput. Syst. **79**, 57–71 (2017)
4. Hsu, C.-H., Slagte, K.D., Chen, S.-C., Chung, Y.-C.: Optimizing energy consumption with activity consolidation in clouds. Inf. Sci. **258**, 452–462 (2014)
5. Liu, J., Jinmin, H.: Dynamic batch processing in workflows: model and implementation. Future Gener. Comput. Syst. **23**, 338–347 (2007)
6. Liu, J., Wen, Y., Li, T., Zhang, X.: A data-operation model based on partial vector space for batch processing in workflow. Concurrency Comput. Pract. Experience **23**, 1936–1950 (2011)
7. Dou, W., Xiaolong, X., Meng, S., Yang, J.: An energy-aware virtual machine scheduling method for service QoS enhancement in clouds over big data. Concurrency Comput. Pract. Experience **29**, e3909 (2016)
8. Xu, R., Wang, Y., Huang, W., Yang, Y.: Near-optimal dynamic priority scheduling strategy for instance-intensive business workflows in cloud computing. Concurrency Comput. Pract. Experience **29**, e4167 (2017)
9. Rahman, M., Hassan, R., Ranjan, R., Buyya, R.: Adaptive workflow scheduling for dynamic grid and cloud computing environment. Concurrency Comput. Pract. Experience **25**, 1816–1842 (2013)
10. Moreno, M., Mirandola, R.: Dynamic power management for QoS-aware applications. Sustain. Comput. Inf. Syst. **3**, 231–248 (2013)
11. Ma, Y., Gong, B., Sugihara, R., Gupta, R.: Energy-efficient deadline scheduling for heterogeneous systems. J. Parallel Distrib. Comput. **72**, 1725–1740 (2012)
12. Changtian, Y., Jiong, Y.: Energy-aware genetic algorithms for activity scheduling in cloud computing. In: Chinagrid Conference IEEE, pp. 43–48 (2012)
13. Kim, N., Cho, J., Seo, E.: Energy-credit scheduler: an energy-aware virtual machine scheduler for cloud systems. Future Gener. Comput. Syst. **32**, 128–137 (2014)
14. Yassa, S., Chelouah, R., Hubert, K., Granado, B.: Multi-objective approach for energy-aware workflow scheduling in cloud computing environments. Sci. World J. **2013**, 13 (2013)
15. Tang, X., Chen, C., He, B.: Green-aware workload scheduling in geographically distributed data centers. In: IEEE International Conference on Cloud Computing Technology and Science Proceedings, pp. 82–89 (2013)
16. Cui, L., Zhang, T., Xu, G., Yuan, D.: A scheduling algorithm for multi-tenants instance-intensive workflows. Appl. Math. Inf. Sci. **7**, 99–105 (2013)
17. Li, Z., Ge, J., Haiyang, H., Song, W., Hao, H., Luo, B.: Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds. IEEE Trans. Serv. Comput. **11**, 713–726 (2018)
18. Potts, C.N., Kovalyov, M.Y.: Scheduling with batching: a review. Eur. J. Oper. Res. **120**, 228–249 (2000)
19. Pufahl, L.: Modeling and executing batch activities in business processes. University of Potsdam (2018)
20. Zhang, W., Duan, P., Yang, L.T., Yang, S.: Resource requests prediction in the cloud computing environment with a deep belief network. Softw.: Pract. Experience **47**, 473–488 (2017)
21. Kimura, B., Yokoyama, R.S., Miranda, T.O.: Workload regression-based resource provisioning for small cloud providers. In: 2016 IEEE Symposium on Computers and Communication (ISCC), pp. 295–301. IEEE (2016)
22. Ardagna, D., Casolari, S., Colajanni, M.: Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems. J. Parallel Distrib. Comput. **72**, 796–808 (2012)

23. Roy, N., Dubey, A., Gokhale, A.: Efficient autoscaling in the cloud using predictive models for workload forecasting. In: 2011 IEEE 4th International Conference on Cloud Computing, pp. 500–507. IEEE (2011)
24. Sunirma, K., Manna, M.M., Mukherjee, N.: Prediction-based instant resource provisioning for cloud applications. In: IEEE/ACM International Conference on Utility and Cloud Computing, pp. 597–602. IEEE (2015)
25. Rahmanian, A.A., Ghobaei-Arani, M., Tofighy, S.: A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment. Future Gener. Comput. Syst. **79**, 54–71 (2017)
26. Caglar, F., Gokhale, A.: iOverbook: intelligent resource-overbooking to support soft real-time applications in the cloud. In: IEEE International Conference on Cloud Computing, pp. 538–545. IEEE (2014)
27. Wang, Z., Wen, Y., Chen, J., Cao, B., Wang, F.: Towards energy-efficient scheduling with batch processing for instance-intensive cloud workflows. In: International Symposium on Parallel and Distributed Processing with Applications (2018)
28. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw. Pract. Experience **41**, 23–50 (2011)