Energy Cost Minimization in Cloud Datacenter

Authors

Md. Arshad Wasif(154426)
Tanvir Ahmed Akash(154445)

Supervisors

Dr. Muhammad Mahbub Alam
Professor
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

A thesis submitted to the Department of CSE in partial fulfillment of the requirements for the degree of Bachelor of Science in CSE

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT) Gazipur, Bangladesh
November, 2019

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Md. Arshad Wasif and Tanvir Ahmed Akash under the supervision of Dr. Muhammad Mahbub Alam, Professor, Department of Computer Science and Engineering, Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Md. Arshad Wasif	Date
Tanvir Ahmed Akash	Date

Supervisor:

Prof. Dr. Muhammad Mahbub Alam	Date

Abstract

Nowadays, more and more companies migrate business from their own servers to the cloud. With the influx of computational requests, datacenters consume tremendous energy every day, attracting great attention in the energy efficiency dilemma. In this paper, we investigate the energy-aware resource management problem in cloud datacenters, where green energy with unpredictable capacity is considered. Via proposing a robust reinforcement learning-based decentralized resource management framework. Because the reinforcement learning method is informed from the historical knowledge, it relies on no request arrival and energy supply. Experimental results show that our approach is able to reduce the datacenters' cost significantly compared with other benchmark algorithms.

Acknowledgements

It is an auspicious moment for us to submit our thesis work by which are eventually going to end our Bachelor of Science study. At the very beginning, we want to express our heartfelt gratitude to Almighty Allah for his blessings bestowed upon us which made it possible to complete this thesis research successfully. Without the mercy of Allah, we would not be where we are right now.

We would like to express our grateful appreciation to Dr. Muhammad Mahbub Alam, Professor, Department of Computer Science and Engineering, Islamic University of Technology for being our adviser and mentor. His motivation, suggestions and insights for this thesis have been invaluable. Without his support and proper guidance, this thesis would not see the path of proper itinerary of the research world. His valuable opinion, time and input provided throughout the thesis work, from the first phase of thesis topics introduction, research area selection, proposition of algorithm, modification and implementation helped us to do our thesis work in proper way. We are grateful to him for his constant and energetic guidance and valuable advice.

We would like to extend our vote of thanks to all the respected jury members of our thesis committee for their insightful comments and constructive criticism of our research work. Surely they have helped us to improve this research work.

Last but not the least, we would like to express our sincere gratitude to all the faculty members of the Computer Science and Engineering department of Islamic University of Technology. They helped make our working environment a pleasant one by providing a helpful set of eyes and ears when problems arose.

Contents

Chapter 1

Introduction

In this chapter, we first present an overview of our thesis that includes the signification of the problem and the problem statement in detail. Research challenges to be faced in the whole scenario is also discussed based on the problem statement. Thesis objectives, motivations and our contribution are noted in sections. The end of this chapter has the description of the organization of the thesis

1.1 Overview

1.1.1 Data Center Networking

Data center is a pool of resources (computational, storage, network) interconnected using a communication network. Data Center Network (DCN) holds a pivotal role in a data center, as it interconnects all of the data center resources together. DCNs need to be scalable and efficient to connect tens or even hundreds of thousands of servers to handle the growing demands of Cloud computing. Today's data centers are constrained by the interconnection network.The legacy three-tier DCN architecture follows a multi-rooted tree based network topology composed of three layers of network switches, namely access, aggregate, and core layers. The servers in the lowest layers are connected directly to one of the edge layer switches. The aggregate layer switches interconnect together multiple access layer switches. All of the aggregate layer switches are connected to each other by core layer switches. Core layer switches are also responsible for connecting the data center to the Internet. The three-tier is the common network architecture used in data centers. However, three-tier architecture is unable to handle the growing demand of cloud computing. The higher layers of the three-tier DCN are highly oversubscribed. Moreover, scalability is another major issue in three-tier DCN. Major problems faced by the three-tier architecture include, scalability, fault tolerance, energy efficiency, and cross-sectional bandwidth. The three-tier architecture uses enterprise-level network devices at the higher layers of topology that are very expensive and power hungry.

1.2 Problem Statement

Energy Cost Minimization in Cloud Data Center

1.3 Significance

In most of the cities, our life relies on the functioning and availability of one or multiple data centers. It is not an overstatement. Most of the things in every segment of human activity such as energy, lighting, telecommunications, internet, transport, urban traffic, banks, security systems, public health, entertainment and even our physical integrity are controlled by data centers.

In brief, the welfare and security of billions of human beings is passed on to these centers of control and supervision of data and information. Most common people may not worry too much about it. However, big corporations and public institutions, on the other hand, have a responsibility to act seriously against such matters.

A break of these data centers would prevent us completely from sending a simple query to Google, since this giant portal holds more than 700 trillion pages of information on thousands of data centers around the globe. Usually, people prefer to avoid thinking about the issue, as according to them, nothing can be done to change the situation.If compared with old data centers, these new modern data centers have four times higher processing capacity and occupy only 40% of the space. Strictly saying, it is not because of the reduction of components, but due to the ongoing research for the overall system efficiency, which consists of the concepts of virtualization and cloud computing.

1.4 Research Challenges

Everywhere you turn these days “the cloud” is being talked about. This ambiguous term seems to encompass almost everything about us. While “the cloud” is just a metaphor for the internet, cloud computing is what people are really talking about these days. It provides better data storage, data security, flexibility, increased collaboration between employees, and changes the workflow of small businesses and large enterprises to help them make better decisions while decreasing costs.

It is clear that utilizing the cloud is a trend that continues to grow. We have already predicted in our business intelligence trends article the importance and implementation of the cloud in companies like Alibaba, Amazon, Google and Microsoft.

The significance of the cloud is increasing exponentially. Gartner forecasts that the cloud services market will grow 17.3% in 2019 ($206.2 billion) and by 2022, 90% of organizations will be using cloud services. Considering all the potential and development cloud computing has undergone in recent years, there are also many challenges businesses are facing. In this article, we have gathered 10 most prominent challenges of cloud computing that will deliver new insights and aspects in the cloud market. But first, let’s start with a simple explanation of the general characteristics and basic definitions. In January 2018, RightScale conducted its annual State of the Cloud Survey on the latest cloud trends. They questioned 997 technical professionals across a broad cross-section of organizations about their adoption of cloud infrastructure. Their findings were insightful, especially in regards to current cloud computing challenges. To answer the main question of what are the challenges for cloud computing, below we have expanded upon some of their findings and provided additional cloud computing problems that businesses may need to address. Security issues, cost management and containment, lack of resources/expertise, gov-

ernance/control, compliance, managing multiple clouds, Performance, building a private cloud, segmented usage and adoption, migration are the major challenges.

1.5 Contributions

The use of cloud computing services and applications continues to increase at a rapid rate, leading to the rise of vast 'hyperscale' cloud data centers.Both consumer and business applications are contributing to the growing dominance of cloud services. For consumers, streaming video, social networking, and search are among the most popular cloud applications; for business users, enterprise resource planning (ERP), collaboration, and analytics are the top growth areas, according to research from Cisco.And, driven by the rapid increase in use of cloud apps, data center traffic is growing fast, expected to reach 19.5 zettabytes (ZB) per year by 2021, up from a mere 6.0ZB per year in 2016, according to Cisco. Cloud data center traffic will represent 95 percent of total data center traffic by 2021, compared to 88 percent in 2016.The growth of Internet of Things (IoT) applications, such as smart cars, smart cities, and connected health devices, will also expand data center demands. By 2021, Cisco expects IoT connections to reach 13.7 billion, up from 5.8 billion in 2016, according to its Global Cloud Index.

Cloud computing security is an important problem when deploying its services. Users consider security to be the most important aspect when deciding to utilize a service cloud computing. Architectural views of cloud computing are views of stakeholders that have different roles and thereby responsibilities for implementing security mechanisms. Security belongs to cross-cutting aspects and pass over all layers of architecture. The Reference Architecture of cloud computing according to ITU-T Recommendation X.1601 specifies only basic security mechanisms and does not determine security responsibilities to roles in relevant services. Considerable division of responsibilities is in IaaS - Infrastructure as a Service. Responsibility for the security of some layers of architecture is on the service provider and the service customer as well. The division of responsibilities based on their requirements is solved in the article as the security architecture framework.

1.6 Organization of Thesis

The rest of this thesis is organized as follows:
Chapter 2 gives an overview of different approaches for sentiment analysis. This chapter also describes the reasons for choosing angles and mean, standard deviation for gait recognition.
Chapter 3 proposes a solution to increase accuracy and robustness against view and scale variant data. It contains the framework, implementation of the proposed methodologies and also contains other methodologies that we tested.
Chapter 4 presents result analysis and comparison with other implementations and studies.
Chapter 5 presents conclusions and discusses future work.

Chapter 2

Literature Review

The first section of this chapter gives a brief overview of existing methods of cost minimization.

2.1 Analysis of Cost Minimization

Data centers as computing infrastructures for cloud services have been growing in both number and scale. However, they usually consume enormous amounts of electricity that incur high operational costs of cloud service providers. Minimizing these operational costs thus becomes one main challenge in cloud computing. In this paper, we study the operational cost minimization problem in a distributed cloud computing environment that not only considers fair request rate allocations among web portals but also meets various Service Level Agreements (SLAs) between users and the cloud service provider, with an objective to maximize the number of user requests admitted while keeping the operational cost minimized, by exploiting the electricity diversity. To this end, we first propose an adaptive operational cost optimization framework that incorporates time-varying electricity prices and dynamic user request rates. We then devise a fast approximation algorithm with a provable approximation ratio for the problem, by utilizing network flow techniques. Finally, we evaluate the performance of the proposed algorithm through experimental simulations, using real-life electricity price data sets. Experimental results demonstrate that the proposed algorithm is very promising, and the solution obtained is nearly optimal.

2.2 Background

Many cloud service providers, such as Amazon EC2, Google, and Microsoft Azure, provide global services through their distributed data centers. As agreed in their Service Level Agreements (SLAs), cloud service providers must guarantee that their services are within the specified tolerant response times by users. To this end, the cloud service providers usually over-provision their services by switching on more servers than needed. This however results in only 10–30% server utilization, leading to the waste of huge amounts of electricity. It is estimated that the total electricity bill for data centers globally in 2010 was over $11 billion, and this figure is almost doubled every five years. However, it is noticed that electricity prices in

different time periods and regions are different, this creates great opportunities for cloud service providers to minimize the electricity bills of their data centers through allocating user requests to geographically cheaper-electricity data centers while still meeting various user SLA requirements.This solution however may not be applicable to large-scale data centers with multiple SLA requirements. The other is that user requests from different web portals at different regions must be fairly allocated to different data centers. Otherwise, a biased allocation may severely degrade the reputation of the cloud service provider in those under-allocated regions, thereby reducing the potential revenue of the cloud provider in these regions in future.

2.3 Overview of Well Known Cloud Data Centers Cost Minimization

It is well-known that cloud providers deploy many geographically distributed datacenters and usually rent bandwidth from multiple ISPs for their inter-datacenter traffic. Second, the distribution of traffic loads among interdatacenter links is nonuniform and partial links experience extremely low bandwidth utilization. This severely restricts the scalability of deployed applications. Moreover, the lack of any performance guarantee makes service providers unwilling to deploy applications across multiple datacenters. Accordingly, it will in turn decrease the revenue of cloud providers. Fortunately, bandwidth guarantee can enable the desirable network performance for applications across datacenters. Prior bandwidth allocation methods, however, mainly focus on intra-datacenter traffic and cannot be simply used to address inter-datacenter traffic for the following reasons. Common algorithms that are used for cost minimization purpose are exchanged cube connected cycle(ExCCC), dynamic voltage frequency scaling and minbrown.

2.4 Exchanged Cube Connected Cycle

An interconnection pattern of processing elements, the cube-connected cycles (CCC), is introduced which can be used as a general purpose parallel processor. Because its design complies with present technological constraints, the CCC can also be used in the layout of many specialized large scale integrated circuits (VLSI). By combining the principles of parallelism and pipelining, the CCC can emulate the cube-connected machine and the shuffle-exchange network with no significant degradation of performance but with a more compact structure. We describe in detail how to program the CCC for efficiently solving a large class of problems that include Fast Fourier transform, sorting, permutations, and derived algorithms.

2.4.1 Methodology of ExCCC

Popular Hypercubic networks used as parallel machines are the Butterfly network, the Cube-Connected Cycles network, the Shuffle-Exchange network and the De-Bruijn network. For a collection of their properties and many algorithms for them. In particular, these constant-degree networks are able to execute so-called normal hypercube algorithms with only constant slowdown if compared to the execution time on the hypercube which has non-constant degree. Among the characteristic

parameters of networks, the eigenvalues of their adjacency matrices are very important. They reflect many structural properties of the network. For instance, from the eigenvalues it can immediately be decided whether the network is bipartite. Expansion properties, bisection problems, the mixing time of Markov chains and the computation of the isoperimetric number, are fields of application of eigenvalues in algorithmic graph theory. In the area of parallel computing, there is a direct connection between the eigenvalues and the routing number. Further applications can be found in the analysis of parallel loadbalancing algorithms and in the design of interconnection networks. The set of eigenvalues is called the spectral set. In the spectrum of a graph, additionally the multiplicities of the eigenvalues are considered. For formal definitions, see Subsec. II-B. Previously, only the full spectral sets of the DeBruijn network and the two variants of the Butterfly network, have been known.

2.4.2 Problems in ExCCC

The hypercube structure is a very widely used interconnection topology because of its appealing topological properties. For massively parallel systems with thousands of processors, the hypercube suffers from a high node fanout which makes such systems impractical and infeasible. In this paper, we introduce an interconnection network called The Extended Cube Connected Cycles (ECCC) which is suitable for massively parallel systems. In this topology the processor fanout is fixed to four. Other attractive properties of the ECCC include a diameter of logarithmic order and a small average interprocessor communication distance which imply fast data transfer. The paper presents two algorithms for data communication in the ECCC. The first algorithm is for node-to-node communication and the second is for node-to-all broadcasting. Both algorithms take O(log N) time units, where N is the total number of processors in the system. In addition, the paper shows that a wide class of problems, the divide and conquer class, is easily and efficiently solvable on the ECCC topology. The solution of a divide and conquer problem of size N requires O(log N) time units. nterconnection networks often constrain the performance of multicores chips or parallel computers. Cube Connected Cycles (CCC) is an attractive interconnection network because of its symmetry, small constant node degree and a small diameter.

2.5 Dynamic Voltage Frequency Scaling

Dynamic voltage and frequency scaling (DVFS) is widely used to match system power consumption with required performance. DVFS is most effective when dynamic power is the dominant power consumption mode. Since power consumption varies quadratically with voltage but gate delay varies only linearly, we can improve the processor's efficiency by running at lower voltages. However, the power supply voltage must be kept high enough to provide adequate performance for the current workload. Operating system software monitors the workload, determines the proper settings for voltage and clock speed, and configures the hardware appropriately.Race-to-dark (RTD) is designed for logic with high leakage currents. This algorithm executes tasks as fast as possible so that the processor can be put into a sleep mode that minimizes leakage current

A simple model allows us to compare DVFS and RTD. Assume that the computing task requires n cycles of execution. The clock period is T and power supply voltage is V. The goal of power management is to minimize energy consumption while ensuring that execution time is less than a deadline Xd.

2.5.1 DVFS Architecture

An architecture for dynamic voltage and frequency scaling operates the CPU within this space under a control algorithm. Figure 2.12 shows a DVFS architecture. The clock and power supply are generated by circuits that can supply a range of values; these circuits generally operate at discrete points rather than continuously varying values. Both the clock generator and voltage generator are operated by a controller that determines when the clock frequency and voltage will change and by how much. A DVFS controller must operate under constraints in order to optimize a design metric. The constraints are related to clock speed and power supply voltage: not only their minimum and maximum values, but how quickly clock speed or power supply voltage can be changed. The design metric may be either to maximize performance given an energy budget or to minimize energy given a performance bound While it is possible to encode the control algorithm in hardware, the control method is generally set at least in part by software. Registers may set the value of certain parameters. More generally, the complete control algorithm may be implemented in software.

Chapter 3

Proposed Methodologies

This chapter presents our proposed method that utilizes to evaluate and analyse the sentiment of videos and comments and finally we determine the disparity in their sentiments . The first section provides an overview of the proposed architecture by outlining the components of the system. In the subsequent sections, these components are described in details.

3.1 Framework

we consider the model that includes cloud DCs, users and their requests, green energy, and grid. DCs distributed in different areas are denoted by a set C. Different from typical architectures,4 users could submit their requests to different DCs. In our proposed blockchain-based framework, these requests can be scheduled by DCs themselves, which removes the dependency on a scheduler in cloud DCs. Particularly, we consider a set of users' requests R each of which asks for more or less computational resources to run VM. Because our proposed framework is decentralized, requests are privileged to be submitted to each DC. In Figure 3.1, we illustrate the possible requests submissions with dashed paths. Additionally, our model considers a discrete time series, which is expressed as T. The data processing rate of requests r is denoted by d j kj([r 1,n]) that would fluctuate over time t. When a large amount of data arrives, the increased computational resource use will consume more energy.

center.PNG center.PNG

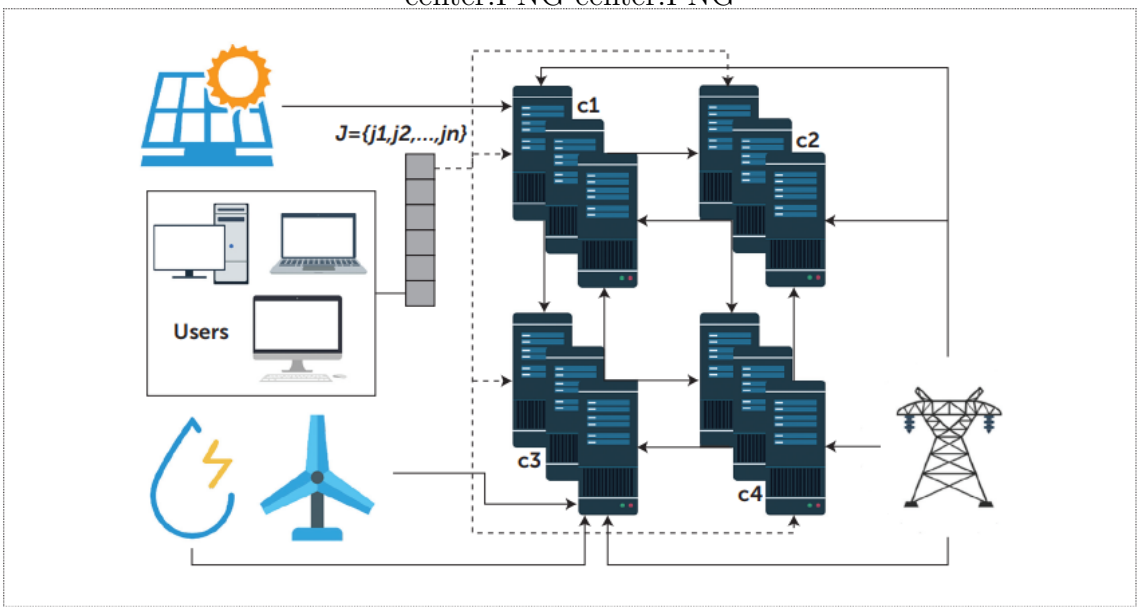

Figure 3.1: Architectural of traditional data center

3.2 Blockchain-Based Resource Management

Blockchain is a linked data structure that is kept by every participant of a blockchain network. It was proposed by S. Nakamoto to solve the consensus problem of the Bitcoin network.A user uses his private key to sign a transaction while interacting with Node0. Hence, the transaction could be traced via the user's public key, and the digital signature also strengthens security and data integrity. Then, the transaction is broadcast to one-hop neighbor of Node0 (i.e., Node1 and Node2). The neighboring nodes (i.e., Node1 and Node2) verify the broadcast transaction obeying the transaction protocol and broadcast it to neighbors (Node3 and Node4), or the transaction will be dropped. By repeating the procedures mentioned previously, this transaction finally spreads across the whole blockchain network. All transactions generated by the network during an agreed time interval by all participants are packaged into one block by a mining node.

3.2.1 Blockchain-Based Resource Management Framework

The core mechanisms of our proposed framework are transaction, mining, and a smart contract. Transaction is used to execute migration, mining brings superior robustness, and a smart contract supports various user-designed algorithms.Within the cloud DCs context, the transaction serves request migration by recording the resource allocation of request VMs. To illustrate the mechanism of applying transaction in request migration, we introduce our proposed DC-adapted transaction protocol. The content of this protocol includes the request ID, the request migration source and destination, and resource allocation, which are presented in Table 1. As shown in the table, a transaction whose ID is 1 represents that Alice submits a request that asks for CPU core, GB RAM, GB disk storage, Mbps I/O, and ò GPU cards. Note that the number of required resources is negative, which represents that the request requires resources. The transaction whose ID is 2 shows that Node0 migrates a VM created by Alice to Node1. Specially, we use a transaction, whose source is nodename and destination is update, to update the available resource of the node. The typical situation to use this transaction is introducing a new node, e.g., transaction whose ID is 0 that introduces Node0 in Table 1 to the DCs network. When a nodename is in the destination row, it should add the resource listed in the transaction and vice versa. Assuming that Node0 has no resource at the beginning and transactions in Table 1 happen according to the order, the CPU resource left can be calculated and found as 2.Records are privileged to be deleted and created but not directly modified, which is done to prevent conflicts among nodes. A smart contract is a script stored in our blockchain-based framework, which is triggered by transactions sent to it. The DCs migrate requests and VMs by executing the smart contract. Consider a simple DCs network that has participants DC1, DC2,...,. The network always migrates requests and VMs to the DC that has the lowest load. The smart contracts stored in each DC can be designed.

Chapter 4

Experimental Analysis

In this chapter, we discuss about the dataset creation, comparison, dataset, result analysis based on different criteria.

4.1 Database Creation

The IT organization deploys the DBaaS solution enabling end users (developers and DevOps) to provision a database of their choice, on-demand, from a catalog of supported databases, which could include both relational and non-relational databases. The IT organization can configure the DBaaS to support specific releases of these software titles, and can further restrict the configurations that specific users can provision. For example, developers may only be allowed to provision databases with a small memory footprint using traditional disks while DevOps could provision higher capacity servers with SSD's. Finally, the IT organization can setup policies for standard database operations like backups, DR and security policies to ensure that the data is properly saved from time to time to allow for recovery when required.

4.2 Evaluation Methodologies

The proposed Cloud system environment in this paper consists of virtual machines, physical machines, and an SLA (Service Level Agreement) manager. Physical machines are the basic available resources and it is assumed that there are a limited number of physical machines. The physical machines provide a set of virtual machines which are configured dynamically according to user requests. When the limited physical machines are provided to users from a pool of resources, the provided resources have two types; one is the dedicated resources and the other is the undedicated resources to give some extra margin in case of sudden request rise as shown in the slash regions. In this Cloud system environment, if a new user requests resources when all of the resources are already assigned, then the undedicated resources allocated to others are provided to the new users via dynamic reconfiguration. We proposed mechanisms that sort high performance resources by analyzing the history information of the undedicated resources for providing highly trusted resources dynamically when the user requests arise. This is aimed to provide highly trusted resources to users based on the analysis of an idle server's history information together with the proposed algorithms. To maintain the current status of the system

and provide the requested resources additionally, we analyze each node's log information at a regular interval so that we can sort and rank the resources and provide the best resources to a user as soon as possible. We designed and implemented the proposed algorithms with the Java programming language and for the experiments; we assumed that there exist 100 undedicated nodes in the slash regions shown in Figure 1. We also assumed that each node's log information is recorded at the same time interval which is 15 seconds. However, gathering real status log data from 100 real machines every 15 seconds is an unreasonable task so we randomly generated data for node specification (high, medium, low) and the resource usage.

Chapter 5

Conclusion

5.1 Summary

With significant improvements in cloud computing technologies and applications in IoT, we have witnessed an explosion of data. Massive amounts of data are generated in DCs, which not only evoke various promising data-driven services but also consume tremendous energy every day. In this paper, we are concerned with the cost minimization issues in cloud DCs and study how to reduce the total cost of energy consumption from the traditional power grid, request scheduling cost, and request migration in DCs. To this end, we develop a blockchain-based decentralized resource management framework, where requests can be scheduled by DCs themselves without depending on the scheduler in cloud DCs. Furthermore, we propose the RL-based request migration method with an embedded smart contract in our framework for cost saving. Finally, simulations are operated based on Google cluster traces and the real-world electricity price, demonstrating the superior performance in energy cost saving compared with other benchmark algorithms in DCs.