iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 Issue: VIII Month of publication: August 2020

DOI: https://doi.org/10.22214/ijraset.2020.31233

Dispatching the Request based on Energy Price for Cloud Data Centers

Reshma T. S[1], Riya M. Saji[2]
[1, 2]BE (CSE), Anna University

***Abstract:** Cloud services are utilized by the data center resources and this helps hosted applications to use them when needed. Cloud service providers control tens of geographically distributed data centers to supply an enormous number of computational resources. Since each data center is consists of many thousands of physical machines, the foremost concern for cloud service providers is energy consumption. Those companies that are imposed by electric costs and which lead them to significant financial overheads and increases the worth for the cloud users. This paper presents the dispatcher of the request supported the energy-price in an electric-cost-saving way by forwarding client requests to data centers. In our technique, a policy that shifts client requests to electrically cheaper data centers gradually along by taking into consideration application latency requirements and data center loads. The results supported our simulation show that our technique can reduce electric costs by quarter-hour quite randomly dispatching client requests.*

***Keywords:** Cloud computing, Electric Cost savings, Dispatching of request, Mapping node, Peer-to-peer.*

I. INTRODUCTION

Due to the resulting significantly high electric costs, energy consumption is that the major concern for cloud service providers. Tens of megawatts of electricity are consumed by one single data center, which implies is analogous to the energy footprint by the tens of thousands of people households [4]. Such electric cost increases prices for cloud users.

During this paper, we present the dispatcher of the request supported the energy-price in an electric-cost-saving way by forwarding client requests to data centers. We argue that by arranging the data centers carefully over various regions can contribute to saving electric costs. We use *mapping nodes* as authoritative DNS servers which forward client requests to data centers at relatively lower electric-costs. By employing a distributed hash table (DHT), mapping nodes form a structured P2P network for his or her scalability and availability. Mapping nodes decide which data center will handle and meet client requests by monitoring electric prices within the regions of data centers. However, to realize an electric-cost-saving dispatch of client requests, it's not enough to require into consideration only the electrical prices. There are two issues we'd like to think. First, the demand from cloud users. To create the client-perceived response - time of their applications as for brief as possible huge fees are payable by some cloud users. During this case, the cloud service provider wants to host the we use *mapping nodes* as authoritative DNS servers which forward client requests to data centers at relatively lower electric cost. Also, we develop application on data centers with the shortest network distance to clients to shorten latency, irrespective of the electrical prices. Second, the data center loads. Even the electrical price within the data center is cheaper than the others and client requests are forward to the data center. The client requests can't be handled when the data center fully utilizes. The further client requests should be redirected to the following electrically cheapest data center during this case. Besides, whether or not the electrical prices of data centers decreased rapidly, we'd like to avoid drastic workload shifts thereto. By considering these issues, for determining the destinations of the client's request we developed a dispatching request policy. Our technique promises the following benefits. One, even if data centers consume the same amount of energy cloud service providers can reduce the electric bill. Two, mapping nodes considered application latency requirements and data center loads, shifting client requests to electrically-cheaper data centers gradually. The results based on our simulation show that our technique can reduce electric costs by 15 % more than randomly dispatching client requests.

II. VARY ELECTRIC PRICES

According to the analysis of Kayaaslan et al. [5], electricity prices vary from country to country. The foremost expensive countries it's seven times dearer than in cheaper countries. Besides, because of various reasons like the weather, oil crisis, and natural disasters, electricity price changes.

Moreover, the value of electricity varies even within a rustic. Some areas of the U.S. traded for electricity within the wholesale markets. In these markets, supported the balance of supply and demand electricity is priced. The electricity trade rapidly grows and reached 40% of total electricity trading within the U.S.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VIII Aug 2020- Available at www.ijraset.com

Figure .1 shows the value of electricity in three regions of the U.S. We will see from this graph, electricity prices vary from region to region over time and are the most cost effective region varies from one to a different, and therefore the price difference is vital. Quite $60 per MWh is that price varies between New York and California at about 60 hours on the x-axis. Besides, the best and lowest prices within the same area differ by two factors.

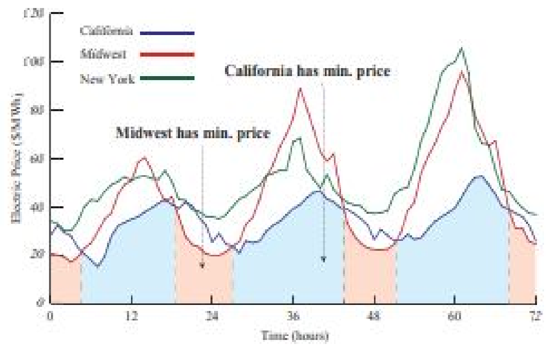

Fig. 1 Electric prices in three regions of U.S.

III. DISPATCHING THE REQUEST BASED ON ELECTRIC PRICE

This paper presents the dispatcher of the request supported the energy-price by forwarding client requests to data centers. In our technique, we use *mapping nodes* as authoritative DNS servers which forward client requests to data centers at relatively lower electric cost. Also, we develop a policy that shifts client requests to electrically cheaper data centers gradually along by taking into consideration application latency requirements and data center loads.

A. Mapping Node

Mapping nodes that are used as authoritative DNS servers thanks to existing DNS servers aren't suitable for changing frequent entries. Supported the electrical prices and/or data center loads, we've got to frequently alter the data center to dispatch client requests appropriately to electrically cheap data centers.

We form a distributed hash table- (DHT-) based mapping nodes contained infrastructure which is found everywhere the planet. This infrastructure helps to realize fast updates and high availability. Mapping nodes manage values and keys. The mapping entries, collection of data of the hosted application, because the values and appearance it up with the hostname of the service provided by the applications because the key. The following are the knowledge contained in a very mapping entry. Hostname is that the key utilized in DHT to define the hostname of applications. Each data center contains an inventory of the IP addresses of the applications load balancer at the destination. The destination is select for every client request by the mapping nodes. The outline of the estimated load of every request is found by the estimated workload. The load balancer's providing workload information utilized by the mapping node to update this field. Estimation of data center load in dispatching requests using this field. Cloud users define their latency limits and set their latency demands for his or her applications.

B. Dispatching Request Policy

With the conservative procedures, the workload request assignment is optimized by each mapping node locally. And through each local procedure, the request dispatching policy assigns the client requests approximately to appropriate data centers. Based on proportions P, the destination of client requests decided by our request dispatching algorithm, where each pi indicates the suitable ratio of the allocated workload to the data center for correct interval i. P is estimated from the workloads assigned to data centers by each node for the previous interval, explained well later. The mapping node returns a load balancer IP address within the data center after accepting requests from clients by this algorithm.

TABLE I
Dispatching Request Policies

	Name	Description
1	random	randomly dispatches
2	latency and cost	based on latency and electric price
3	latency and load	based on latency and data center loads
4	our load	based on latency, electric price, data center loads

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VIII Aug 2020- Available at www.ijraset.com

The workload allocated to every data center is adjusted supported the differences in electric prices and a few of the workloads are shifted from the electrically affordable data center to a more electrically expensive data center if their electric prices are identical. Using the proportion between the costs in both data centers, we decide what number workloads are to be moved. To reduce the violation of the latency demand, the workload is tuned. When the mapping node got a client request, the node simulates a situation during which the client request is forwarded to each data center and retains the amount of the violations of the latency demand in each data center. If the data center can supervise the overall amount of workloads apart from the previously violated requests, we allocate the workload. Else, the remaining workload is moved to the succeeded electrically affordable data center. In the end, we limit the workload changes under a predefined proportion of the previous workload.

IV. EXPERIMENTS

To illustrate the functionality of our dispatcher, we conducted simulation-based experiments using CloudSim toolkit 2.1.1 [2]. CloudSim could be a toolkit for simulating cloud computing environments. Three policies are prepared for comparison. Table I concludes the policies. *Random* sends client requests to every data center randomly. When the electrical price is that the lowest and therefore the latency to the client is below a given latency demand, *Latency and price* dispatches client requests to a data center This policy is love to the routing scheme of Qureshi et al. [7]. *Latency and Load* is that the converse approach to *Latency and Cost*. This policy dispatches requests to the data center with the bottom latency to the clients, as long as its resource utilization is smaller than a threshold.

1) Results: Fig. 2 demonstrates how each policy takedown electric costs. The y-axis denotes electric cost savings supported *Random*. The figure, it shows that our proposal reduces the electrical cost by 15%. Our policy records more electric cost savings than *Random and Latency and Load*. This is because these policies do not consider electric price when sending requests. Although *Latency and Cost* reduce more electric costs than our policy, it has a drawback that the data center becomes overloaded.
2) Fig. 3 shows the rate of latency limit violations on each policy. This figure shows that there is no latency limit disrupt in *Latency and Load*, *Latency and Cost*, or our policy while some violations happen in *Random* policy.

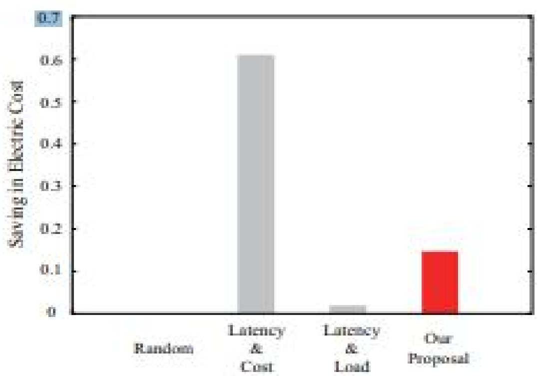

Fig.2 Electric cost saving based on random policy.

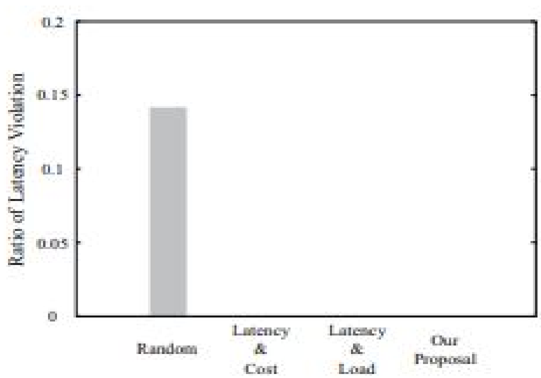

Fig.3 Ratio of latency limit violation.

V. RELATED WORK

A simple cost-aware request routing policy is been proposed by Qureshi et al. that maps client requests to locations where the electrical cost is affordable. Some way to cut back the electrical bill of web search engines operating at data centers that are geographically distributed is been proposed by Kayaaslan et al. They proposed a probabilistic algorithm that dynamically shifts workloads between data centers. However, their perspectives don't take into consideration data center workload. There are approaches to diminish energy consumption to chop electric costs. A server consolidation technique using virtual machine live migration is been presented by Nathuji et al. [6]. Live migration enables data center operators to control various power management policies. Elastic Tree [3] tries to reduce energy consumption of data center networks. It dynamically modifies the set of active network elements (e.g. links and switches) in response to data center traffic loads.

Various types of request-distribution techniques have been proposed. Balancing the load between mirror servers to reduce the response time discern by clients and or achieve high accessible services is their most aim. Round-robin DNS [1] acquire stability in load by returning the address of a duplicate server for each DNS translation request. DONAR, a distributed system that can unload the burden of duplicate selection of geo-replicated services is been proposed by Wendell et al. [9]. Ramasubramanian et al. [8] dispense the Cooperative Domain Name System (CoDoNS). CoDoNS gain its scalability, decentralization, self-organization, and failure flexibility by using DHT.

VI. CONCLUSIONS

This paper dispenses a request dispatching technique that depicts client requests to data centers where the electric price is relatively smaller. Electric costs are not only Key insight behind our technique but also the application's latency, demands, and datacenter loads by considering these aspects, mapping nodes forward client requests to relevant data centers. Our simulation-based experimental results exhibit that our dispatching technique can victoriously reduce the electric cost of data centers.

REFERENCES

[1] T. Brisco. DNS Support for Load Balancing. RFC 1794 (Informational), 1995.
[2] R. Buyya, R. Ranjan, and R.N. Calheiros. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In Proc. of the Int'l Conf. on High Performance Computing Simulation, pages 1–11, 2009.
[3] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. ElasticTree: Saving energy in data center networks. In Proc. of the USENIX NSDI '10, pages 249–264, 2010.
[4] R.H. Katz. Tech titans building boom. IEEE Spectrum, 46(2):40–54, 2009.
[5] E. Kayaaslan, B.B. Cambazoglu, R. Blanco, F.P. Junqueira, and C. Aykanat. Energy-Price-Driven Query Processing in Multi-center Web Search Engines. In Proc. of the 34th int'l ACM SIGIR Conference on Research and development in Information, pages 983–992, 2011.
[6] R. Nathuji and K. Schwan. VirtualPower: coordinated power management in virtualized enterprise systems. In Proc. of the ACM SOSP '07, page 265, 2007.
[7] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for internet-scale systems. In Proc. of the ACM SIGCOMM '09, pages 123–134, 2009.J
[8] V. Ramasubramanian and E.G. Sirer. The Design and Implementation of a Next Generation Name Service for the Internet. In Proc. of the ACM SIGCOMM '04, pages 331–342, 2004.
[9] P. Wendell, J.W. Jiang, M.J. Freedman, and J. Rexford. DONAR: decentralized server selection for cloud services. In Proc. of the ACM SIGCOMM '10, pages 231–242, 2010.