Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 3, March 2017, pg.228 – 235

Effective Resource Allocation for Video Service through Load Balancing in Distributed Cloud Data Centers

N.R.RejinPaul[1], Iswariya P[2], Kanaka M[3], Keerthana T[4]
Student, Dept. of Computer Science and Engineering, Velammal Institute of Technology[2,3,4]
Asst.Professor, Dept of Computer Science and Engineering, Velammal Institute of Technology[1]

Abstract -- Cloud computing provides efficient platform for Video Service Providers to running multimedia applications in a cost effective manner. The main problem is Load balancing among various large distributed server systems. Existing algorithm focus only on high queuing overhead and difficulty in predicting the user demand, hence there is a delay in processing. In this paper, we propose a method called Dynamic Request Redirection and Resource Provisioning to address the load balancing problem. Video Service Providers live various virtual machines from multiple geographical distributed datacenters that are close to video requestors to run their services. Our proposed method reduce the long-term time average cost of renting cloud resource while maintaining the user quality of service and experience

Index Terms – Video Service Provider, Load Balancing, Data Centers

1. INTRODUCTION

Multiple heterogeneous applications concurrently run in distributed cloud data centers (CDCs) for better performance and lower cost**.** The data center is used to store a data within an organization of local network. A cloud service provider (CSP) uses datacenters to host cloud services and cloud based resources through load balancing on video services. In First stage, we create Content Delivery Network(CDN) by admission control. The Cloud Data Center handle a resource renting from multiple CSPs and load balance the user requests to these resources in a nearly virtual machine. And then virtual machine has been created in a different location by admin controller. The content delivery network(CDN)is a system of distributed servers (network) that deliver web pages and other Web content to a user based on the geographic locations of the user, the origin of the webpage and a content delivery server. CDN providers use additional techniques to optimize the delivery of files in optimal data centers. And then CDN admin can login into his account with this credentials to view the CDN architecture.

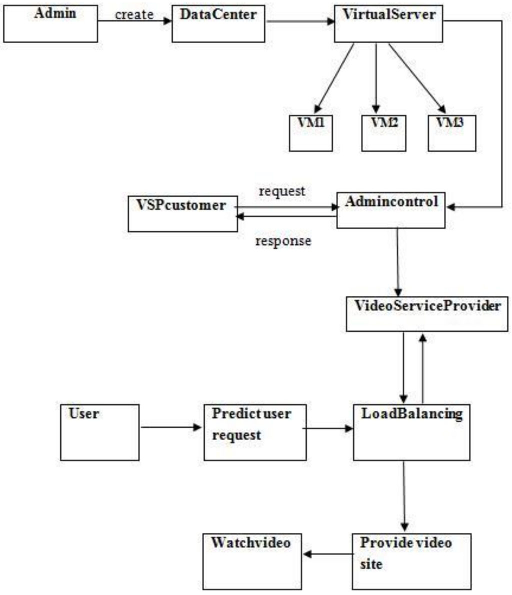
They are 3 additional technique for CDN are 1)HTTP redirection 2)IP (internet protocol) redirection 3)DNS redirection.In this paper we use a HTTP redirection technique to optimize the delivery files.Then the admin can Configure add, delete, modify virtual instances in various data centers. Policy file will be generated for user request for dynamic request redirection and enabling good quality of service.

In second stage we process a CDN request and response, here we approach our Implemented idea on video application using Dynamical Request Redirection and Resource Provisioning (DYRECEIVE) Algorithm. The video service provider(VSP) request for the cloud service provider to host their application in the cloud. The video service provides choose the Virtual instances on various data centers and request the CDN to host their Services.The video service provider application has the various types of videos such as the high quality, medium quality and the low quality videos. The rent for data center usage will virtual instances. Now he can deploy his own video service application in the CDN by packaging the content and sending to various data center. Then the services as started and made available to all user through CDN.

Finally, In the third stage is the work scheduling process, resource allocation in the cloud can be classified based on different perspectives of cloud providers and cloud users. There are many efforts on designing Scheduling strategies for cloud providers. For single datacenters, improving resource utilization and fairness are often the focus. For multiple datacenters, some work propose scheduling strategies to minimize the cost of electricity use through balancing load among geographically located datacenters.

Users from different regions obtain various services like video streaming from CDN by the policy the video service provider already generated. Once the VSP receives a request, the request will be dynamically redirected to an optimal datacenter like that High quality, Medium quality, Low quality, based on previous user survey and workloads in geographical location.

With our approach the video service provider is able to provide an efficient, cost effective and quality service to any number of clients.



Process of CDN

2. RELATED WORK

Here, we discuss the related work and presents the contribution of the workload admission control and workload scheduling in comparison to existing works.

A. Performance Modeling

There are several works focusing on the performance modeling and analysis of cloud infrastructure by considering Virtual Machines (VMs).The author in **[3]** to evaluate the performance by creating several replicas of each job and sending each replica to a different server. Upon the arrival of a replica to the head of the queue at its server, the latter signals the servers holding replicas of that job, so as to remove them from their queues.

Indeed, upon the job's arrival, no data collection is performed.However there is delay in executing the required signal when the job begins processing and there is a queuing overhead. If the replicas arrives at the heads of the servers at the same time, then flaw may occur. The author in **[6]** proposes a **Join-Idle-Queue** algorithm for large-scale load balancing with distributed dispatchers by decoupling the discovery of lightly loaded servers from job assignment. The basic version involves idle processors informing dispatchers at the time of their idleness, without interfering with job arrivals. Here the request is directed to randomly chosen dispatcher. Informing a large number of dispatchers will increase the rate at which jobs arrive at idle processors, but runs the risk of inviting too many jobs to the same processor all at once and results in large queuing overhead. On the other hand, informing only one dispatcher will result in wasted cycles at idle processors and assignment of jobs to occupied processors instead, which adversely affects response times. In this paper, we model the system to reduce the delay and queuing overhead by redirecting the user request to the idle server

B. Admission Control

The objective of admission control is to protect servers from overload and to guarantee the performance of applications.

In **[8]** the author provide an efficient spectrum allocation problem that is attracting a lot of attention and they use a distributed algorithm for selecting channel, so the load is distributed among them. The author used a load balancing algorithm that includes Compare and Balance algorithm. In this algorithm they use a nash equilibrium concept for making allocation in precise manner.The another algorithm is avoid contention which is used for providing efficient system. But these algorithms only supports games theoretic approach. Generally, the channel allocation approach in wireless network field. In this paper, we use the admission control scheme for content delivery network (CDN) to allocate a datacenters because the admission control is important in wireless network area. Admission control scheme is used to avoid a network congestion.so that performance is maintained based on user requirement.

C. Traffic overhead

There are several works focusing on reducing the traffic overhead. The author in **[14]** provides an **idealized process** by analyzing the supermarket model. The task is sent to the randomly chosen server among n servers. Customers are served based on FIFO policy. However there is a locality problem. The author in **[7]** proposes an ***Extended Supermarket Model (ESM)*** to focus on quantifying the utility of monitoring for self-adaptive load sharing, where a stream of jobs arrives at a collection of n identical servers. In the centralized ESM model, all client request arrives at a centralized load-balancing device. Then the device selects servers uniformly at random with minimal queue length to process the request. Sometimes the smaller task needs to wait for larger task to complete, hence delay in processing. In this paper, we propose an DYRECEIVE algorithm to solve this problem.

D. Task Dependency

The author in **[13]** proposes an extension of LFF (Least Flow-time First) task assignment policy , called LFF-PRIORIY. LFF-PRIORITY dynamically computes two priorities, namely task size and task size priorities, and put them in a priority based on multi-section queue. However there is some limitations with LFF-PR1ORITY. One of them is that if the system load is very high and task variation is not high, the policy performs unsatisfactorily, because it tries to balance task size and deadline. This may cause some smaller tasks to be delayed by larger ones. Another limitation is that the policy does not consider the issue of task interdependency

If some tasks are dependent on other tasks, the order of dependency must be maintained in spite of task sizes and task deadlines. In this paper, there is no task dependency because of redirecting the user request.

E. Resource Allocation

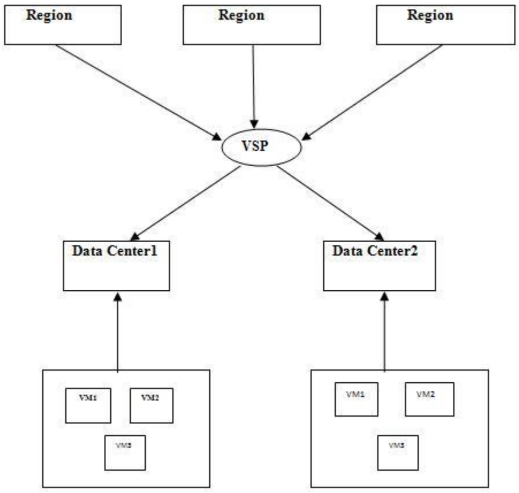
In **[9]** the author provide a automatic datacenter management infrastructure used for automating a software provisioning, system monitoring and it deal with a faulty hardware and software. The author use a cluster computing and reduce the load (load shedding) concept is tedious .In **[5]** the author propose a dynamic capacity management policy for the problem of managing a large application using a capacity inference algorithm.This algorithm meet with a SLA(service level agreement) for robustness. But in SLA difficult to design a good looking product and also long term curing can lead to warping. In this paper we use a DYRECEIVE algorithm concept for minimizing VMs through load balancing scheme in cloud computing for faster scheduling and also easy to provide a service to the customer in effective manner based on user requirement.

3. SYSTEM ARCHITECTURE

This architecture explain about overall structure of resource allocation of video services through load balancing. a framework that systematically handles resource renting from multiple CSPs and schedules user requests to these resources in a nearly optimal manner. In particular, the framework is capable of handling heterogeneous types of user requests, workloads and QoE requirements. VMs in the cloud are of different types and are priced dynamically. We propose an algorithm to solve the jointed stochastic problem to balance the cost saving.

We leverage the existence of content delivery network (CDN) to host video services on their various datacenters distributed in various regions. We give a systematic method called Dynamic Request Redirection and Resource Provisioning (DYRECEIVE) to address this problem. With our approach the video service provider is able to provide an efficient, cost effective and quality service to any number of clients.

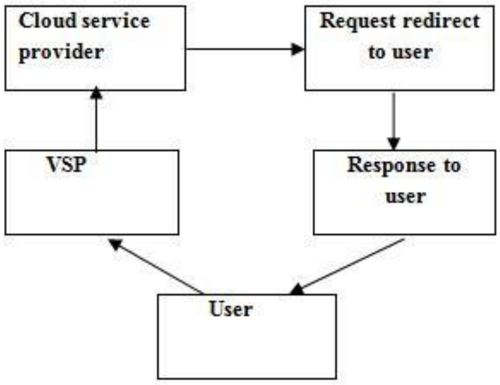
Users from different regions obtain several of services like video streaming and transcoding from VSPs which do not possess their own datacenters but actually rent the infrastructure (VMs) from CSPs. Once the VSP receives a request, the request should be dynamically redirected to an optimal datacenter according to its QoE requirements and the execution cost, considering the different prices of datacenters over different regions.



SYSTEM ARCHITECTURE

This flow diagram explain about the how the user get a response from CDN (Content Delivery Network).User provide a request to VSP(video service provider) then the VSP request to CDN for allocating the VMs (Virtual machine) in nearby datacenter in different geographical location based on user demand. When the CDN finish the process of it user it will redirect the request using DYRECIVE algorithm to the corresponding user.

As user demands are difficult to predict and the prices of the VMs vary in different time and region, optimizing the number of VMs of each type rented from datacenters located in different regions in a given time frame becomes essential to achieve cost effectiveness for VSPs.



FLOW DIAGRAM

4. DYRECEIVE ALGORITHM

In this paper we use a DYRECEIVE (Dynamic Request Redirection and Resource Provisioning) Algorithm for redirect the user request in efficient manner. Our task is to make following decisions: 1) Request redirection, once the request arrives and 2) Resource procurement, every *m* time slots. The ultimate goal is to minimize the resource procurement cost as well as guarantee the user QoE in the long run.

We leverage the existence of content delivery network (CDN) to host video services on their various datacenters distributed in various regions. We give a systematic method called **Dynamical Request Redirection and Resource Provisioning (DYRECEIVE)** to address this problem.

Allocate resources for cloud based video services on user request from multiple regions to distributed data centres and dynamically computes the near and optimal virtual machine.Video Service Providers (VSP) for running compute-intensive video applications in a cost effective manner. VSP may rent virtual machines (VMs)

Multiple geo-distributed datacenters that are close to video requestors to run their services.

Optimizing the number of VMs of each type rented from datacenters located in different regions in a given time frame becomes essential to achieve cost effectiveness for VSPs.

PROCEDURE

1. Input: sk, ωc, Wmax, ℓc a c r, Amax rc, pk d(τ), ρk d, drent, V, a, b, u, v (∀c ∈ C, ∀d ∈ D, ∀r ∈ R, ∀k ∈ K);
2. Output: n c, k d (τ), λc rd(τ) (∀c ∈ C, ∀d ∈ D, ∀r ∈ R, ∀k ∈ K);
3. Initialization step: Let τ = 0, st = cputime, and set Q c d(0) = 0, Hc d(0) = 0, (∀c ∈ C, ∀d ∈ D), ddec(0) = 0;
4. while the service of VSP is running do
5. calculate time slot τ ,τ = (curtime − st)/60s;
6. estimate the decision overhead ddec(τ) based on ddec(t), t ∈ [τ − 5, τ − 1];
7. **Resource provisioning:**
8. For each datacenter d ∈ D do
9. if (τ mod md) == 0 then
10. Observing the queue backlogs Q c d (τ), H c d (τ) and the VM price p k d (τ) at current time;
11. Getting the VM provisioning strategy (n c, k d (τ)) by solving the problem using CVX tool;
12. **Request redirection:**
13. if request arrives at system then
14. for each r ∈ R, c ∈ C do
15. Observing the queue backlogs Q c d (τ), Hc d(τ) ,the network delay drd and estimating the

computation delay dcomp(τ) at current time;

16. Getting the request redirection strategy λ c rd (τ)
17. Update the queues Q c d (τ), Hc d(τ) according to queue dynamic equation respectively.
18. Record the decision-making time consumed at current time slot ddec (τ).

Notations

D	Set of datacenter distributed over multiple region
C	Set of all service classes
R	Set of user regions
K	Set of VMs types
M	Time interval to decide a resource provisioning
p_d^k	The availability of the type-k VM in datacenter –d
ω_c	Workload of type-c request
W_{max}	Max workload of each type of request
l_c	Tolerable delay of type-c service
a_c^r(t)	Number of VMs of type-k in datacenter d
N_{max}	Max number of VMs of each type over all datacenter
A_{rc}^{max}	Max number of request for type-c in region-r
p_d^k	Price to provision to type-k VM in d at t
s_k	Compute capacity of type-k VM
Q_0	Minimal QoE level should be guaranteed for user
Q_{max}	Max QoE level the user can achieve
$H_d^c(t)$	Unprocessed workload of type-c request in d at t
$Q_d^c(t)$	Virtual queue to satisfy the constraint

EFFECTIVENESS OF ALGORITHM:

We run our dynamic algorithm for T = 2, 880 time slots, with parameter V = 2 × 104, m = 10. Present the cost occurred in each time slot. We observe that the monetary cost curve is fluctuating synchronously with the variation of requests, which means that our algorithm can adaptively lease and adjust VMs resources to meet dynamic user demands, without forecasting the future workload information. In detail, the cost comparison of each type of VM is illustrated in (b), in which we use the metric CR for comparison.

It can be observed that, under the variation of workload, the cost ratio of each VM type is relatively stable in the whole sense especially within crowd flash period. It may attribute to the fact that, within crowd period, resources are inadequate to the system and all type of the VMs will be rented to guarantee the user QoE, which cause a stable cost ratio near to the price ratio. Also the Extra Large is shown to have the highest ratio.

5. CONCLUSION AND FUTUER WORK

Thus we proposed a novel method called **DYRECEIVE** (Dynamical Request Redirection and Resource Provisioning) for request redirection and resource procurement from the perspective VSPs. We showed that DYRECEIVE is capable of reducing the cost of providing video services in the cloud and achieving satisfactory user QoE level simultaneously. Thus we allocated resources for cloud based video services on user request from multiple regions to distributed data centers and dynamically computed the near and optimal virtual machine.

The video service application deployment is done on various data centers. This method provides an efficient way to run video services in a general and heterogeneous environment consisting of dynamic user workload, dynamic resource price, multiple services with heterogeneous QoE requirements and heterogeneous Datacenters.

FUTURE ENCHANCEMENT

- Banking
- Transcoding

REFERENCES

[1] Amir Nahir, Ariel Ordaand Danny Raz,"Replication –based Load Balancing "IEEE transactions on parallel and distributed systems, vol. 27, no. 2, february 2016.

[2] J. Dean and L. A. Barroso, "The tail at scale," Commun. ACM,vol. 56, no. 2, pp. 74–80, Feb. 2013

[3] A. Nadir, A. Orda, and D. Raz. (2012). Schedule first, manage later: Network-aware load balancing, Dept. Electr. Eng., Technion,Haifa, Israel, Tech. Rep.[Online].Available:www.ee.technion.ac.il/Sites/People/ArielOrda/Info/Other/NOR12JR.pdf

[4] A. Nahir, A. Orda, and D. Raz, "Distributed oblivious load balancing using prioritized job replication," in Proc. 8th Int. Conf. Netw. Service Manage, 2012, pp. 55–63.

[5] A. Gandhi, M. Harchol-Balter, R. Raghunathan, and M. A.Kozuch, "Autoscale: Dynamic, robust capacity management for multi-tier data centers," ACM Trans. Comput. Syst., vol. 30, no. 4,p. 14, 2012.

[6] Y. Lu, Q. Xiao, G. Kliot, A. Geller, J. R. Larus, and A. G. Greenberg,"Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services," Perform. Eval, vol. 68, no. 11,pp. 1056–1071, 2011.

[7] D. Bridgend, R. Cohen, A. Nahir, and D. Raz, "On cost-aware monitoring for self-adaptive load-sharing," IEEE J. Sel. Areas Commun.vol. 28, no.1, pp. 70–83, Jan. 2010.

[8] S. Fischer, "Distributed load balancing algorithm for adaptive channel allocation for cognitive radios," in Proc. 2nd Conf. Cognitive Radio Oriented Wireless Netw. Commun., 2007, pp. 508–513.

[9] M. Isard, "Autopilot: Automatic data center management,"SIGOPS Oper. Syst. Rev., vol.41, no. 2, pp. 60–67, 2007.

[10] V. Gupta, M. Harchol Balter, K. Sigman, and W. Whitt, "Analysis of join-the-shortest-queue routing for web server farms," Perform.Eval., vol.64, no. 9-12, pp. 1062–1081, Oct. 2007

[11] T. Xie and X. Qin, "Scheduling security-critical real-time applications on clusters," IEEE Trans. Comput., vol. 55, no. 7, pp. 864–879, Jul. 2006.

[12] J. Moon and M. H. Kim, "Dynamic load balancing method based on DNS for distributed web systems," in Proc. 6th Int. Conf. Ecommerce Web Technol., Copenhagen, Denmark, 2005, pp. 238–247.

[13] B. Fu and Z. Tari, “A dynamic load distribution strategy for systems under high task variation and heavy traffic,” in Proc. ACM Symp.Appl. Comput., 2003, pp. 1031–1037.
[14] M. Mitzenmacher, “The power of two choices in randomized load balancing,” IEEE Trans.Parallel Distrib. Syst., vol. 12, no. 10, pp. 1094– 1104, Oct. 2001.
[15] M. Mitzenmacher, “How useful is old information?” IEEE Trans.Parallel Distrib.Syst., vol.11, no. 1, pp. 6–20, Jan. 2000.
[16] K. Park and W. Willinger, Self-Similar Network Traffic and Performance Evaluation, 1st ed. New York, NY, USA: Wiley, 2000