International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 03, 2020
ISSN: 1475-7192

THE PERFORMANCE ANALYSIS OF OPTIMIZED LOAD BALANCING IN MULTIDIMENSIONAL DISTRIBUTED DATABASE SYSTEM FOR VIDEO ON-DEMAND

[1]Suvendu Kumar Jena, [2]Priyabrata Sahu, [3]Dr. Sasmita Mishra, [4]Umakant Bhaskar Gohatre

Abstract: *This report explores how the efficiency of a complex algorithm for load balancing is influenced by alleged delays (i.e. big and small). Here, we note that the existence of alleged delays induces a substantial reduction in load balancing policy results. Here we use stochastic dynamics via a queuing system to model and optimize the load-balancing algorithm. By compromising the load balancing function, the efficiency of the distributed network may be enhanced. In this basis, we take into consideration the question of optimizing a strategy that has a fixed number (one or two) of juggling momentum, thus optimizing strategy in load flow frequency and periods when preparation is carried out. In this paper we address the efficiency of a one-time preparation approach on a dispersed physical network composed of a WLAN. This paper proposes an algorithm for effective load balancing based on the forecasts of the Cicada end-to-end method. A cloud service simulator or Cloud Sim can be used as a simulation and to obtain a low computing demand algorithm and a better balancing of workload. It is a new analytical model which characterizes the mean for the distributed system of the complete completion of the scheduling to analyze the relationship between delay and load balancing benefit. In order to build an autonomous on-demand (sender initial) load balance system, we then use our optimal one time load balance approach.*

Keywords— *Multidimensional database, distributed system; load balancing; video on demand*

I. INTRODUCTION

Distributed multimedia systems are growing a lot of academic and industrial concern due to developments in

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

the high-speed network, data infrastructure and switching technologies. Distributed video server is an innovative multimedia delivery technology that attracts a lot of coverage from the television, telecommunications, and device industries recently. Distributed video services are commonly recognized as an essential residential tool that can overtake today's home entertainment and knowledge infrastructure.

The centralized framework that helps you to replay videos on request through the networks in real time. The standard of a successful centralized server requires: (a) a software free computer, audio and video synchronization, video replay at at least 30 frames per second, and complete user access. The high bandwidth demand and real-time transmission drawbacks of video replay are better for a LAN file server close replay customers. The economic utility and ability to exchange video for a broad tertiary storage facility are a significant motivator of the usage of the control repository. This means that consolidated computing is cost-effective while retaining good efficiency and the scalability of distributed servers.

A centralized networking framework, user controller desktops, database management software, one or more tertiary information collection servers and one or more video Servers with internet information represent the key components of a centralized device. If a report from a customer is made, it is sent to the administration list. On video servers the administration system controls the requested video file. The Management Server must download the video object from an accessible repository and transfer the video object to the chosen file server if it is not available online. The viewer is then transmitted by the video application. In the case of a video submission, choosing a Web Server and retrieval of the requested web item is important. We concentrate on positioning and reducing the expense of playing game items. We implemented several methods of assigning video on request in this job.

Load balancing is just the distribution of the workload between a varieties of distributed coordinating servers. In broad-based distributed processing structures where the servers are physically or nearly distant. The communication-related delays will affect the planned load-balancing strategies that do not allow for these delays. In environments where individual units are linked via a specific high band (for example, the Telephone, ad-hoc networks, wireless LANs or wireless phone), this is a particularly significant issue. In these situations, the delays fluctuate predictably in addition to being high, rendering their estimation precise once. The reliability of these distributed networks is thus stochastic and should be routinely measured in conjunction with all load management techniques. Currently, load management capability. Policies that better fit such late-infested structures of delivery do have to be applied through a theoretical context.

II. RELATED WORK

In the distributed system [4], load balance is one of the most critical issues. It is planned to boost the efficiency of a distributed network by preparing user activities for processors roughly. This should attempt to equalize the

operating loads of the video server during video entity playback automatically. In the case of video requests, a video server is needed and different methods are suggested for selecting video servers for the playback demanded from video artifacts [10] and [11]. They focus on placing video objects and minimizing their playback costs.

A distributed file system for the distributed video system was defined in the focus group grid calculation [7]. The distributed video file system serves as an intermediary for clients and the distributed media network, and the file system architecture encapsulates downloading, synchronization, load balancing and indexing. The feature of video FS is the awareness of the compatible video format in its file format and can provide it in a streaming manner.

The distributed file system is based on the load on the server and the availability in the replicated copies, which means that the server can either serve the file itself or a client in one or more replicated copies of the requested video available on the network. The distributed file-system adopts a centralization and peer-to-peer networking models. The system will begin caching and replication through network nodes to obtain the optimal redundancy and storage redundancies. The versatility it provides to its viewers is usually characteristic of a streaming video network. A true distributed video server would have full power over the presentation session for its remote subscribers. Engine video on request software monitors the data transmission rate across the network. The mechanism through which a storage media file may be translated and distributed in a byte stream through a computer network, which is called streaming when it is downloaded from the recipient. This strategy advances the traditional method to download and play, as of such users, they do not need to wait until accessing the file until it is downloaded fully from the television platform, but may start watching the media immediately.

The distributed video client will select a video from an accomplishment recognized as a network video square from the digital media library [8] and [9]. The consumer has instant storage media power, which is defined as immersive Video-on - Demand (IVOD), as regards start, first, delay, slow motion, random access and re-wind features. Video on demand is not simple and inexpensive, companies are depending more on demand for lower interactivity. Users will pick frequently scheduled times from many hundred channels already transmitted. If all users simultaneously require a given video, the video on demand infrastructure will block a lot. A. Narasimham [1] demonstrated the way the cloud network slows down video-on - demand traffic to reach network interfaces at a head-end in a centralized video system. Intelligent set-top box (STb) is a small computer which converts and converts user information with a CPU, memory, graphic generation capability, audio or video composition functionality. It also decompresses media components in real time and collects remote info. Video upon request is less than that sent to the setup box distinguished by a symmetric data flow as real-time signals sent to the head end. Selecting, by category, description rather than title, and choosing on a personal preferred basis, should be available for a number of smart video-selection procedures. Choose movie from the website or renting films at a low cost may be important.

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

III. PROPOSED SYSTEM MODEL

The key purpose of this research is to manage the demand between the different video servers and reduce the pause in the production of the video requests. A nearly optimal algorithm for choosing the video server is suggested to accomplish this aim. It shows that our approach can achieve strong load balance efficiency. A cloud services simulation is feasible. The end result would demonstrate the feasibility of a quantitative workload balance solution which can achieve better workload balancing by rising computing resource consumption.

*A. **DSP-based load equilibrium adaptation method***

The controller process obtains the system's output process by means of the Ganglia machine control unit when all computation nodes are modified synchronously and the iterative testing amount from each computer node is determined by the usage of the output model.

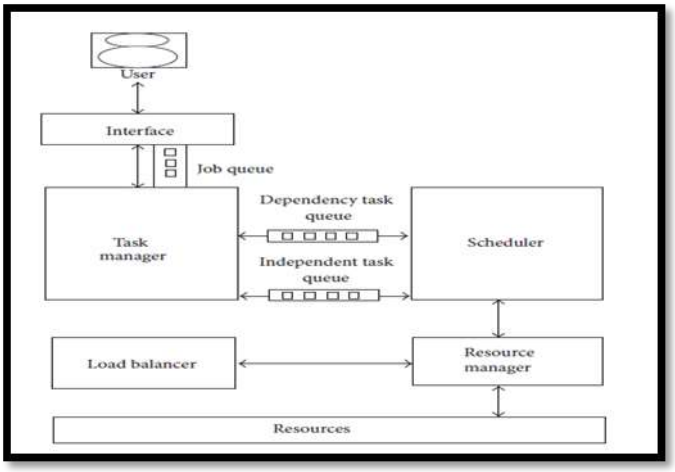


Figure 1: System Architecture Design

The purpose of load balancing is to coordinate computing functions over virtual computers. It just operates in a homogenous setting and doesn't operate on heterogeneous grids.

$$Y_i = f(X_i, \beta) + e_i \quad (1)$$

$$f(X_i, \beta) = \beta_0 + \beta_1 X_i \quad (2)$$

Calculation functions are named "building block" to reduce implementation time. Moreover, in distributed systems, makepan minimization issue is common; we also name it NP-complete. In this way, reducing make-up is not just the duty of balancing loads, but also the need to cope with contact costs.

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

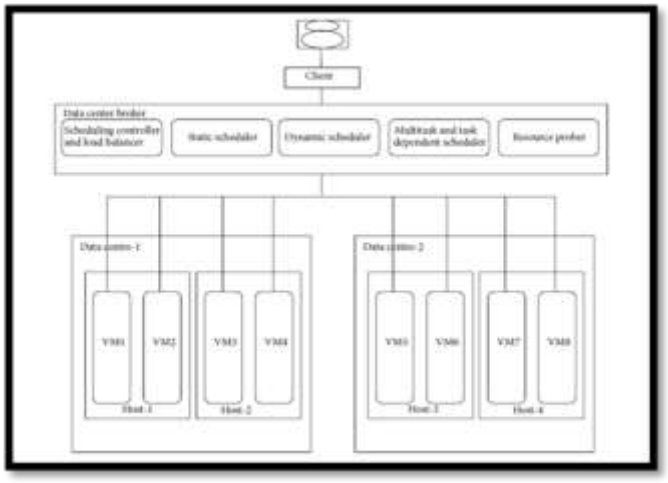


Figure 2: System datacentre loading

A hybrid grid and cloud infrastructure application development technologies [14] lowers the duration of the operating system and the overhead management. In order to solve both the budget and the deadline for scheduling the problem. In a shorter period, this method delivers stronger results. There was consideration of related forms of topics [16–19].

$$\beta_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \tag{3}$$

$$y_i = \beta_0 + \beta_1 x_i. \tag{4}$$

$$\sigma_{\beta_0} = \sigma_\varepsilon \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum(x_i - \bar{x})^2}} = \sigma_{\beta_1} \sqrt{\frac{\sum x_i^2}{n}} \tag{5}$$

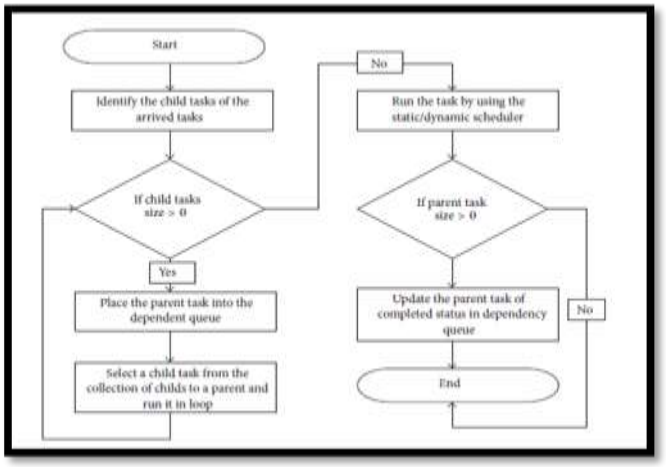


Figure 3: Flow diagram of system

A-DSP introduces DSP-based load equilibrium adaptation method. A-DSP (Parallel Synchronous Adaptive-dynamic). In a cloud system job preparation algorithm, priority task was known as the principal QoS parameter.

$$Y_i = \beta_0 + \beta_1 X_i + e_i \quad (6)$$

$$\sum_i (Y_i - f(X_i, \beta))^2 \quad (7)$$

In fact, this method deals with three key problems such as sophistication, coherence and make-up. This was evaluated to hold the target in mind and the algorithm's recommendation was met. With its quick training and simple to describe interface, as well as its solid learning framework, A-DSP Caffe is well established. Nevertheless, Caffe does not support the transmitted Caffe ML version. This post presents a distributed ML model based on Caffe utilizing the idea of an A-DSP and Parametric Server. There is a relation between network traffic and cloud computing load [3]. Cloud network computer systems typically provide cables for retrieval and transmission such as Ethernet. Bandwidth metrics are known as features of these systems or so-called networks. Thus, the volume of traffic determined by bandwidth-packet transfer determines the necessary capacity for the network connections. Little's Theory, focused on principle of queuing system [3] also illustrates the connection between network traffic and cloud management. The law of the little stuff is that the amount of the average pace and time spent on a list is the cumulative number of items inside a queue system. The Little Theory describes the connection between the overall amount of traffic and the real network usage number [4].

IV. SIMULATIONS LOAD BALANCING TECHNIQUES

The issue of load balancing in distributed systems has gained further interest and relevance with the advancement of computing technologies and the emergence of many distributed systems. This segment addresses some of the main problems involved with complex load balancing in distributed systems. Each of these problems were often discussed in the literature as a product of many specific methods. This portion describes some of the key methods and solutions for load balancing in a distributed network commonly utilized. Load balancing is a method of reassigning the overall responsibility to the specific nodes of the operational machine to the network and usable resources in order to increase the job response time and optimize the system's performance [13]. Estimate load, load contrast, reliability of various networks , network efficiency, interactions between nodes, complexity of work to be transmitted are all core elements in this load balancing In order to significantly boost performance, a contingency plan in the event that the device fails or slightly preserve system reliability, it must be the primary priority to guarantee any system changes. In order to maximize load handling, it is necessary to retain the system stability. There are two main algorithms for load balancing, called static and dynamic. They are both usable. Static load balancing algorithms are an algorithm that fairly separates the traffic on servers and round

robin to despise the traffic on the servers.

It's fast. It would therefore improve the imperfection of the case. It needs previous machine awareness. Even the weighted round robin load is specified to boost the essential issues of the round robin and weight has been allocated to any server. Further contacts are assigned the maximum weight.

$$\sum_{i} e_i^2 = \sum_{i} (Y_i - (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}))^2 = 0 \quad (8)$$

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + e_i \quad (9)$$

$$\sum_{i} e_i^2 = \sum_{i} (Y_i - (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}))^2 = 0 \quad (10)$$

Increasing processor selects other processors in its smaller district in the local load balancing program and uses this local data for a load transfer decision. In order to maintain a local equilibrium, a processor interacts with its nearest neighbors in any move. The key goal is to reduce remote contact and to manage the load of processors efficiently.

However, a certain amount of global information is used in a global balancing scheme to initiate load balance. The DASUD algorithm (Diffusion algorithm) is one of the neighbor's closest groups. The DASUD algorithm performs over triangle, torus and hypercube topology and observes through simulations that this balancing scheme exceeds global balancing strategies in these instances. In the cycle of load balance, the network heterogeneity is reported in the global load balancing step and in the regional load balancing level. Until implementing a regional delivery, transmission costs and machine income must be measured.

Dynamic load balancing does not take the prior device status or action into consideration and relies on the system's current state or behavior. Load balancing choices are made without prior information on the existing condition of the network. It is therefore better to balance load than static approach. Dynamic algorithms to handle load show the correct weights of servers and a lightest node to manage traffic by scanning the entire network. However, it requires real-time coordination with the networks to pick a suitable server to lead to additional device traffic. Cloud load balance relies on a traditional load balance, but varies from classic philosophy on load balancing design and execution by the use of generic servers to handle loads and offer new incentives, economies of scale and problems of its own. The load balancing framework helped to encourage server flexibility and performance [16]. Load balancing in a cloud computer environment is needed in complex and large systems. Till the day where the load becomes a complex metric, it the not stay ideal at the beginning of time as with most real world fields. If such factors are encountered in different structures, we propose the load-balancing issue – if processors obtain work, they may be

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

sub optimally distributed, then a range of activities are moved such that the duration is shortened. The multiprocessor preparation concern or conventional load-balancing allows one to delegate workers to each of the empty processors in a series of differing sizes. Another aim is to reduce the duration of the CPU, which is the maximum powered.

$$\sum_{i=1}^{n} \sum_{k=1}^{p} x_{ij} x_{ik} \beta_k = \sum_{i=1}^{n} x_{ij} y_i, j = 1, \ldots, p. \quad (11)$$

$$(\mathbf{X}^\top \mathbf{X}) \boldsymbol{\beta} = \mathbf{X}^\top \mathbf{Y}, \quad (12)$$

$$\boldsymbol{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}. \quad (13)$$

A job requires the resources of a VM and as a number of tasks arrive at a VM, the resources are drained, meaning that no new work requests are available. The VM is said to have reached an overwhelmed state when such a situation arises. At this stage, activities are either hungry or end in stalemate with no chance of accomplishing them. As a consequence, tasks on other VM will move to another tool.

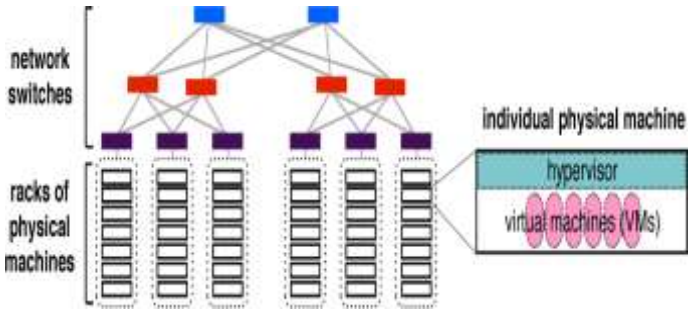


Figure 4: Networks switches and physical machine

The method of moving workloads requires three simple steps: load balancing, which tests the existing system load, resource selection, and another sufficient resources and workload migration. Three systems generally known as the load handling systems, the acquisition of services and the transfer of activities are used for these operations.

V. TO REDUCE REMOTE LOAD OF PROCESSORS EFFICIENTLY

One of the most critical facets of a distributed network is load balancing. It tries to boost the outcomes of a distributed framework by the correct allocation of consumer activities on processors. In a video on demand program

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

more precisely, it aims to equalize the workloads of the data processor dynamically during video object replication, so that only video servers are working for less time or video queries are waiting for an average time. Consequently, the aim of this paper is to find a solution for selecting video servers to manage the load between video servers and to minimize service request time.

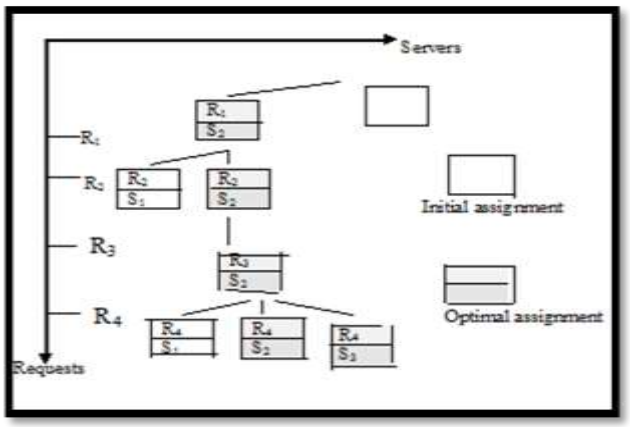


Figure 5: Search Tree

The algorithm of assignment checks all possible pairs of assignments (input client, file server) via solution space. We use a tree in the algorithm to display the search field. The Fig gives an illustration of this branch. 1 Each node in the search tree should reflect a video server and a video request on each point. The partial schedule of each node in this tree consists of servers. A node edge provides an extension of the partial assignment by adding another video application to the assignment of a video server for a partial schedule of the node.

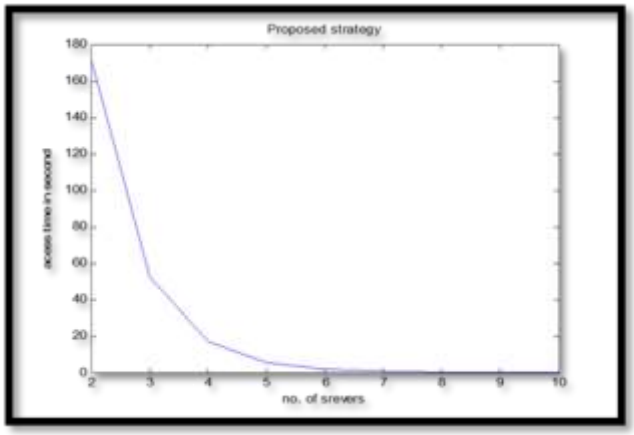


Figure 6: Proposed strategy graph

An expenses function of the nodes leads to the desired solution for the search. The search will be guided. Particularly in any iteration, a video request candidate is chosen to position the lowest cost node (highest value

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

of the remaining accessible load L). When a leaf node was hit, video queries should be delegated to file servers. The algorithm then follows back to search at more alternatives. This method is to continue either I until all potential alternatives have been found or (ii) until the timeline ends (i.e. no leaf node has been reached). In that case, a new search of the remaining requests was then initiated at the next scheduling stage.

The queue of the Work Waiting List includes the pending workers for one specific VM; furthermore, the estimates of the least used VM for each worker will be made after the Task Execution, Job Pause List and Job Wait List have been obtained from each VM. This less used VM data is then transferred to the planner. The Resource Manager meets with all VMs in order to collect all of their resources, including their numbers of processing elements and their processing power. In addition, this resource planner determines the weight for every VM based on its assigned computing power. The optimized memory in each of the VMs is also defined.

$$\rho(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}, \sigma^2) \propto (\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right). \tag{14}$$

$$\boldsymbol{\beta} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{15}$$

The concern with choosing a web server is whether the web item is to be put on a server. The video submission typically utilizes a batch processing methodology. Suppose N video servers are in the network, and a batch of M video artifacts is available. The preference is based on the expense of displaying video items. This expense can be seen by loading between video servers. For every video server the remaining load is used here. If there's a video item on a file server then we claim it's loaded. Because the load of a video server is reduced, the lower the load accessible is with the more data artifacts on a video server. The video server will put more video artifacts if the remaining available load is heavy. In this scenario, we say the cost is small, which implies that the cost is investment relative to the charge that is usable.

$$vs^2 = (\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}) \text{ and } v = n-k, \tag{16}$$

$$\rho(\boldsymbol{\beta}, \sigma^2) = \rho(\sigma^2)\rho(\boldsymbol{\beta} \mid \sigma^2), \tag{17}$$

$$\rho(\sigma^2) \propto (\sigma^2)^{-\frac{v_0}{2}-1} \exp\left(-\frac{v_0 s_0^2}{2\sigma^2}\right). \tag{18}$$

$$\rho(\boldsymbol{\beta} \mid \sigma^2) \propto (\sigma^2)^{-\frac{k}{2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Lambda}_0(\boldsymbol{\beta}-\boldsymbol{\mu}_0)\right). \tag{19}$$

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

In fact, this method deals with three key problems such as sophistication, coherence and make-up. This was evaluated to hold the target in mind and the algorithm's recommendation was met. With its quick training and simple to describe interface, as well as its solid learning framework, A-DSP Caffe is well established. Nevertheless, Caffe does not support the transmitted Caffe ML version. This post presents a distributed ML model based on Caffe utilizing the idea of an A-DSP and Parametric Server

In our cost feature, the remaining load depends on three core elements: the size of your buffer, your loading capability and your network bandwidth. The cost feature plays a specific position increasing portion of the remaining usable load. Since a video object requires its own size and stream capabilities, it consumes a buffer space and a network bandwidth whenever a video object is mounted on a video server which therefore impacts network loading performance. In other terms, the cache source and network bandwidth have loads applied.

$$\rho(\boldsymbol{\beta}, \sigma^2 \mid \mathbf{y}, \mathbf{X}) \propto \rho(\boldsymbol{\beta} \mid \sigma^2, \mathbf{y}, \mathbf{X}) \rho(\sigma^2 \mid \mathbf{y}, \mathbf{X}), \quad (21)$$

$$\boldsymbol{\Lambda}_n = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \boldsymbol{\Lambda}_0), \boldsymbol{\mu}_n = (\boldsymbol{\Lambda}_n)^{-1}(\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\Lambda}_0\boldsymbol{\mu}_0), \quad (22)$$

As load between video servers would have an effect on the service utilization and the waiting time of customers, load balancing is critical and is a challenge in choosing the video service. The issue often aims at reducing the time needed for servicing video requests. They establish waiting times to complete the transition of the submitted video object from a file server to the client when the file request is initiated. The time spent on the assignment of video artifacts to data servers is known as time spent. One solution to which the processing period is to insure that the time limit is not reached. This method has been incorporated in our proposed video server selection algorithm. It communicates with computer nodes by maintaining a thread series and tracks the performance and the number of iterations per process cycle of the computer nodes. The performance control cluster module receives and manages data distributed in real time from all system nodes.

VI. RESULTS

Upon evaluating the performance of each computing node based on techniques, the complex synchronous control and workload allocation module adjust the low threshold w, the stalk thresholds and the distributed operational load mi.

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

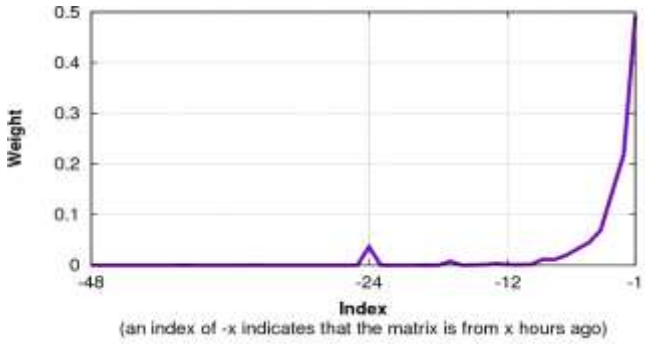


Figure 7: Index with Weight

Computing nodes: data processing panel, calculation panel, stylization module and performance control module primarily contain the following. Nodes for programming.

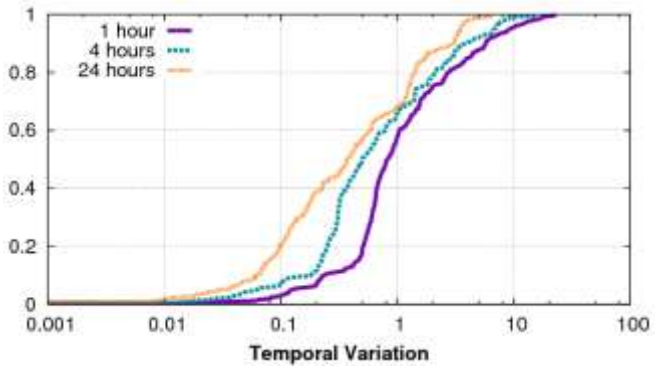


Figure 8: Temporal variation representation

The next iterative training takes the system node repeatedly before a stop condition is reached when a distributed ML model is built on the basis of the iterative converging algorithm. For the traditional research of the distributed ML model, the composition data sets for the replication of any processing node are the same fixed scale.

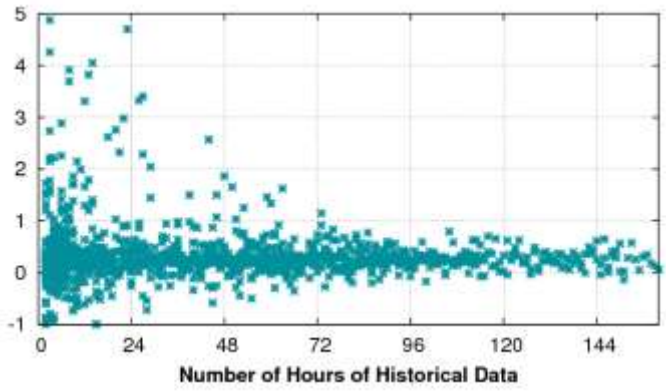


Figure 9: Number of house data

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

$$b_n = b_0 + \frac{1}{2}(\mathbf{y}^T\mathbf{y} + \boldsymbol{\mu}_0^T\boldsymbol{\Lambda}_0\boldsymbol{\mu}_0 - \boldsymbol{\mu}_n^T\boldsymbol{\Lambda}_n\boldsymbol{\mu}_n). \quad (24)$$

$$p(\mathbf{y}|m) = \int p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}, \sigma)p(\boldsymbol{\beta}, \sigma)d\boldsymbol{\beta}d\sigma \quad (25)$$

$$p(\mathbf{y} \mid m) = \frac{1}{(2\pi)^{\frac{n}{2}}}\sqrt{\frac{\det(\boldsymbol{\Lambda}_0)}{\det(\boldsymbol{\Lambda}_n)}} \cdot \frac{b_0^{a_0}}{b_n^{a_n}} \cdot \frac{\Gamma(a_n)}{\Gamma(a_0)} \quad (26)$$

$$p(\mathbf{y} \mid m) = \frac{p(\boldsymbol{\beta}, \sigma|m)p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}, \sigma, m)}{p(\boldsymbol{\beta}, \sigma \mid \mathbf{y}, \mathbf{X}, m)} \quad (27)$$

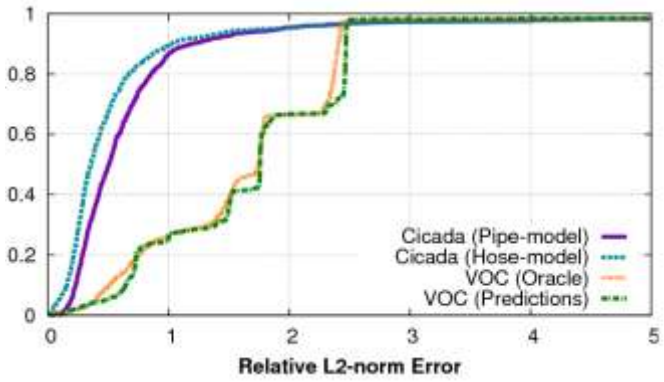
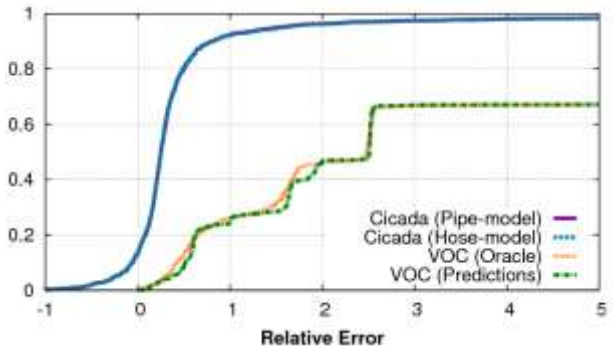


Figure 10: Relative error

Simulation and probabilistic modeling should evaluate the load handling output of the storage strategies. And the techniques as described [2], [13] and [14] do well in the sense that a strong likelihood of a successful load balance is achieved. Here, we observed a compromise between the performance and the time to calculate the retrieval selection (RSP) problem, which of the two disks should be used for retrieval for each data block. The redundancy of data gives us the freedom to obtain a reasonable load and to make use of this freedom, we must solve a question of selection in any time.



Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

Figure 11: Relative Error 2

The storage strategy's load balance efficiency is the disk costs per device, since efficient usage of the limited file space reduces the overall amount of devices,

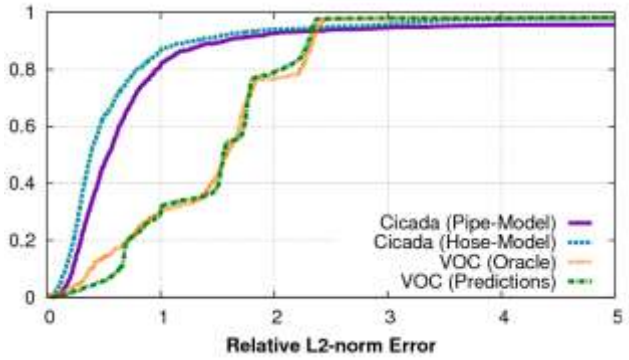


Figure 12: Relative L2 Norm Error results 1

The dynamically chosen random volume capacity balancing; data blocks are placed on two disks at random. The block load balance is often implemented at block level in [2] the goal is to reduce as much block as possible as necessary.

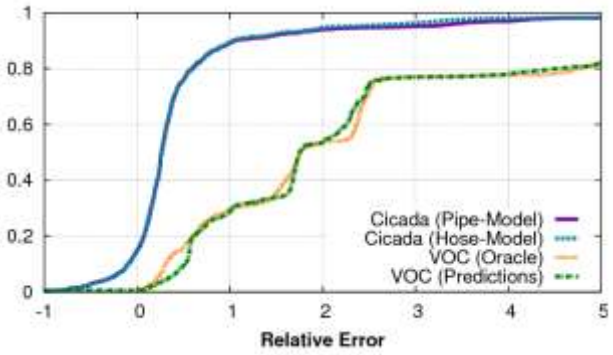


Figure 13: Relative error 3

Introduced an algorithm named the assignment algorithm for video selection in this proposed methodology section. This is based on an incrementally comprehensive quest with backtracking to achieve nearly optimum solutions. This method uses a quest methodology.

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020

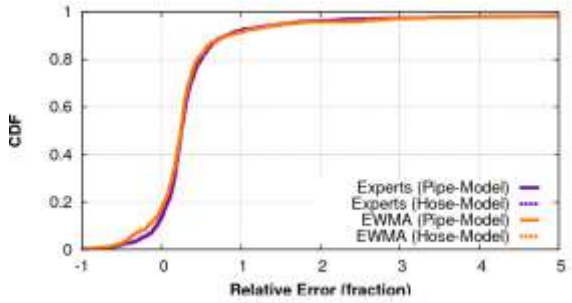


Figure 14: Relative error fraction results 4

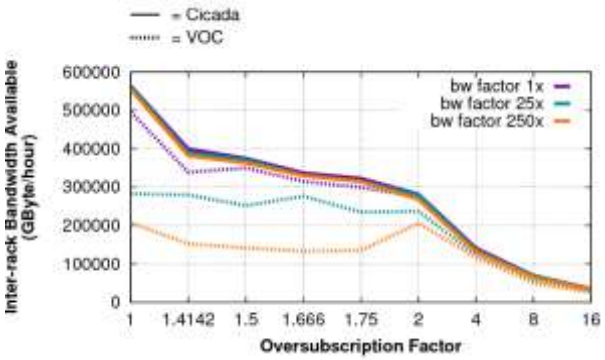


Figure 15: Oversubscriptions factor

The method of selection. The demand on the system must be measured under capacity limits and the time required for a service video request reduced. A list of requested video artifacts is provided that await caching on video file servers, along with a list of accessible video file servers and their limitations.

VII. CONCLUSION

In the distributed video on request program, we suggested an effective approach to solve the video collection issue. The technical approach on which the search procedure, schedules and the data request are centered are restricted in time and allocated to the data file server. The search cost feature is based on the storage capacity factor of the application, which is similar to the disk space dimension, the network bandwidth and the demand on the computer. The stimulation results show improved performance in video requests that reflected an even load balance between video servers, using the methodology.

REFERENCES

[1]. Amruthur, Narasimhan, “proceeding of Intelligent Information System (115-97)”, pp. 455-460, AT & T Labs. Holmolel, NJ, 07733.

[2]. Joep Aerts, Jan Korst and Sebastian Egner, "Random duplicate storage strategies for load balancing in multimedia servers", Philips Research Laboratories.

[3]. Chor Ping Low, Hongatao Yu, Jim Mee Ng, Qingping Lin and Yacine Atif, "An efficient Algorithm for the Video Serveur Selection problem", pp. 1329-1333, School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore-639 798.

[4]. A.Srivastava, A.Kumar and A.Singru, "Design and Analysis of a video- on-Demand Server", Multimedia System, Vol.5, No. 4, pp.238-254, July 1997.

[5]. G.Dammicco, U. Mocci, "Optimal Server Location in VOD Networks", Proceeding of 1997 IEEE Global Telecommunications Conference (GLOBECOM'97), Vol.1, pp.197-201, 1997.

[6]. C.C. Bisdikian, B.V. Patel, "Issues on Movie Allocation in Distributed Video-on-Demand System", Proceedings of IEEE International Conference on Communication. Vol.1 pp.250-255, June 1995.

[7]. C.Z. Xu and F.C.M Lau, Load Balancing in Parallel Computers; Theory and Pracitice, Kluwer Academic Publishers, 1997.

[8]. L.A. Rowe and D.A. Berger, "The Berkeley Distributed Video Server", Multimedia computing-Proceedings of the Sixth NEC Research Symposium, Tokyo, Japan, June 1995.

[9]. J. Aerts, J. Korst, W. Verhaegh. Load balancing for redundant storage strategies: Multiprocessor scheduling with machine eligibility. Submitted to Journal of Scheduling.

[10]. J. Aerts, J. Korst, and W. Verhaegh. Load balancing in multimedia servers. In Proceedings seventh international workshop on project management and scheduling, pages 25-28, April 2000.

[11]. W. Tetzlaff and R. Flynn. Block allocation in video servers for availability and throughput. In Proceeding Multimedia computing and networking, 1996.

[12]. G. Colouris "Introduction to Distributed System", concepts and design. Second edition,

[13]. R. X. T. and X. F. Z, "A Load Balancing Strategy Based on the Combination of Static and Dynamic, in Database Technology and Applications (DBTA), 2010 2nd International Workshop 2010, pp. 1-4, 2010.

[14]. J. Hu, J. Gu, G. Sun, T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment", *Parallel Architectures Algorithms and Programming (PAAP) 2010 Third International Symposium*, pp. 89-96, 2010.

[15]. R.K. Naha, M. Othman, "Evaluation of Cloud Brokering Algorithms in Cloud Based Data Center", *International Computer Science and Engineering Conference (ICSEC)*, pp. 78-82, 2014.

[16]. K. Mahajan, A. Makroo, D. Dahiya, "Round Robin with Server Affinity: A VM Load Balancing Algorithm for Cloud Based Infrastructure", *Journal of Information Processing System*, vol. 9, pp. 379-394, 2013.

[17]. R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A. De Rose, R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *Software: Practice and Experience*, vol. 41, pp. 23-50, 2011.

[18]. S. G. Domanal, G. R. M. Reddy, "Optimal load balancing in cloud computing by efficient utilization of virtual machines", *Communication Systems and Networks (COMSNETS) 2014 Sixth International Conference*, pp. 1-4, 2014.

[19]. S.-C. Wang, K.-Q. Yan, W.-P. Liao, S.-S. Wang, "Towards a load balancing in a three-level cloud computing network", *Computer Science and Information Technology (ICCSIT) 2010 3rd IEEE International Conference on*, pp. 108-113, 2010.

[20]. P. Venkata Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments", *Applied Soft Computing*, vol. 13, pp. 2292-2303, 2013.

[21]. A. P. Florence, V. Shanthi, "A Load Balancing Model Using Firefly Algorithm in Cloud computing",

Received: 03 Dec 2019 | Revised: 21 Jan 2020 | Accepted: 18 Feb 2020