ISSN 2319-2720
Volume 7, No.2, April – June 2018
International Journal of Computing, Communications and Networking
Available Online at http://www.warse.org/ijccn/static/pdf/file/ijccn22722018.pdf
https://doi.org/10.30534/ijccn/2018/22722018

ARTIFICIAL INTELLIGENCE BASED SECURE CONTENT PROCESSING AND DISTRIBUTION BY COOPERATIVE INTERMEDIARY CLOUDS

Rakshitha R[1], Rakshita R[2], Ramya S[3], Ranjitha Shivamatta[4], Prof. Kiran Mensinkai[5]

[1]Student in CSE, EWIT, India, rrakshithasona@gmail.com
[2]Student in CSE, EWIT, India, rakshitar11@gmail.com
[3]Student in CSE, EWIT, India, ramyaumas96@gmail.com
[4]Student in CSE, EWIT, India, ranju0496@gmail.com
[5]Assistant Professor, Department of CSE, EWIT, India, kiranm@ewit.edu

ABSTRACT

Web applications traffic request fluctuates generally and capriciously. The basic routine with regards to provisioning a fixed limit would either bring about unsatisfied clients (under provision) or waste profitable capital venture (overprovision). The System is such that whole processing is completely dependent on the server and causes high latency to complete the request. Subsequently there is a requirement for low idleness investigation over high speed information streams, motivating the need for distributed intermediaries. We permit multiple intermediaries to simultaneously perform content service on different portions of data using cloud for storage and processing the information. We use "Genetic Algorithm" for processing the request and Multiagent algorithm for enabling the agents to perform tasks of other agents and hence acts as an Alternatives. Our approach is efficient and minimizes the amount of re-transmissions and congestion in network.

Keywords: Artificial Intelligence, Cloud, Content Distribution, Genetic Algorithm, Load Balancing.

1. INTRODUCTION

Today a developing number of organizations need to process gigantic measures of information in a cost-effective way. Exemplary agents for these organizations are administrators of Internet web crawlers, similar to Google, Yahoo, or Microsoft. The huge measure of information they need to manage each day has made customary database arrangements restrictively costly[3].These organizations have promoted a structural worldview in light of a substantial number of ware servers. Issues like preparing slithered records or recovering a web list are part into a few autonomous subtasks, conveyed among the accessible hubs, and figured in parallel. The essential furthest reaches of web server are to store, process and pass nearby pages to customers. The correspondence among customer and server happens utilizing the Hypertext Trade Tradition (HTTP). Pages passed on are most reliably HTML records, which may join pictures, arrangements and substance not withstand the substance. A customer administrator, as rule a web program or web crawler, begins correspondence by making an interest for a specific resource using HTTP and the server responds with the substance of that advantage or a bumble message if unfit to do accordingly. The advantage is customarily a certifiable record on the server's helper accumulating, however this isn't generally the case and depends upon how the web server is executed. While the basic limit is to serve content, a full utilization of HTTP in like manner fuses techniques for getting content from clients. This part is used for submitting web shapes, including exchanging of records.

Congestion in information organizing and queuing hypothesis is the decreased nature of administration that happens when a system hub or connection is conveying a larger number of information than it can deal with. Run of the mill impacts incorporate queuing delay, parcel misfortune or the hindering of new associations. An outcome of clog is that an incremental increment in offered stack leads either just to a little increment or even an abatement in organize throughput.

Framework traditions that usage mighty retransmissions to compensate for package disaster as a result of stop up can grow blockage, even after the fundamental weight has been lessened to a level that would not commonly have affected sort out blockage. Such frameworks show two stable states under a comparable level of load. The unfaltering state with low throughput is known as congestive fall.

Artificial Intelligence is knowledge showed by machines or programming. It is likewise the name of the scholastic field which examines how to make PCs and PC programming that are fit for wise conduct. Evolutionary computation is a family of algorithms for global optimization inspired by biological evolution, and the subfield of artificial intelligence and soft computing studying these algorithms. We will allow numerous cloud servers to perform content administration on various segment information. Our convention bolsters decentralized cloud and key administration and adaptable appointment of administrations. Our usage result demonstrates that our approach is productive and limits the measure of information transmitted over system.

Keeping in mind the end goal to upgrade the execution of cloud servers, a few methodologies have been created in light of the utilization of substance administration administrations gave by cloud. In the greater part of these methodologies, content reserving is the primary administration gave by cloud. That is, rather than approaching a substance server for substance upon every customer ask for, a cloud first checks if these substance are privately stored. Just when the asked for substance are not reserved or outdated are the substance exchanged from the substance server to the customers. In the event that there is a reserve hit, the system transfer speed utilization can be diminished. A reserve hit likewise diminishes get to inactivity for the customers. Framework execution along these lines enhances, particularly when a lot of information is included. Other than these changes, reserving makes the framework vigorous by letting storing cloud give content dispersion administrations when the server isn't accessible.

2. RELATED WORK

Client-side Load Balancer [1] offers the following: To start with, Amazon EC2 just others 5 kinds of virtual servers and application proprietors can't redo the specification of them. Second, again because of its ware plan of action, a cloud ordinarily just gives product Virtual Machines (VM). The calculation control and the system data transmission is regularly not as much as top of the line servers. Third, not at all like in a venture, application proprietors have practically no control of the fundamental cloud framework.

DNSLoadBalancing calculation is utilized to adjust customer – side load and has its drawbacks– stack adjusting granularity and adaptiveness– that are not specific to the cloud. In the first place, it completes a poor employment in adjusting the heap. For execution reasons, neighborhood DNS server reserves the IP address data. In this way, all programs reaching the same DNS server would get a similar IP address. Since the DNS server could be in charge of a substantial number of hosts, the heap couldn't be effectively smoothed out. Second, the neighborhood DNS server reserves IP address for a set timeframe, e.g., for a considerable length of time. Until the point that the reserve terminates, the nearby DNS server guides demands from programs to a similar web server. At the point when traffic fluctuates at once scale considerably littler than days, tweaking DNS server settings has little effect. Nephele, [2] another information handling system for cloud conditions. Nephele takes up numerous thoughts of past preparing structures yet refines them to better match the dynamic and misty nature of a cloud. Characterizing a Nephele work includes three obligatory advances:

To start with, the client must compose the program code for each undertaking of his handling work or select it from an outside library. Second, the program must be allotted to a vertex. At long last, the vertices must be associated by edges to characterize the correspondence ways of the activity.

In conventional registering conditions of circulated figuring, parallel processing [3] and network figuring, scientists in and abroad have proposed a progression of static and dynamic and blended booking systems [6]. In static booking calculation, ISH [7], MCP [8] and ETF [9] calculations in view of BNP are appropriate for little disseminated situations with high web speed and insignificant correspondence delay while MH [10] and DSL [11] calculation in light of APN mull over of the correspondence postponement and execution time so they are reasonable for bigger conveyed conditions. In unique planning calculation, a few calculations ensure the heap adjusting and stack partaking in errand dissemination through self-adjusting circulation and shrewd dispersion. In blended planning calculation, it essentially stresses square with circulation of doled out figuring undertaking and diminishment of correspondence cost of disseminated registering hubs and in the meantime it understands adjusted booking as indicated by the processing volume of each hub. Specialists have additionally directed examinations on calculations of autonomic booking, focal planning, savvy planning and operator arranged booking. There are numerous similitudes and furthermore contrasts between conventional planning calculations and the booking of VM assets in distributed computing condition. To begin with, the greatest contrast between distributed computing condition and conventional processing condition is the objective of planning. In conventional figuring condition, it chiefly plans process or errand so the granularity is little and the exchanged information is little; while in distributed computing condition the booked target is VM assets so the granularity is vast and the exchanged information is expansive too. Second, in distributed computing condition, contrasted and the arrangement time of VMs, the season of planning calculation can nearly be dismissed.

3. METHODOLOGY

The client need not wait for long time if cloud is failed; it gets assured response from a cloud or from service provider if cloud is failed. The cloud are authenticated while giving client request, hence we can securely process client request using valid cloud. It allow numerous go-betweens to all the while perform content administration on various segment information. Advantages are: Minimizes bandwidth congestion, supports decentralized proxy and key management and flexible delegation of services, provides efficient QoS, minimizes the amount of data transmitted across the network and reduces access latency.

There are four modules:

3.1 Client Module:

Login portrays the interface that must be actualized by validation innovation suppliers. It is connected to applications

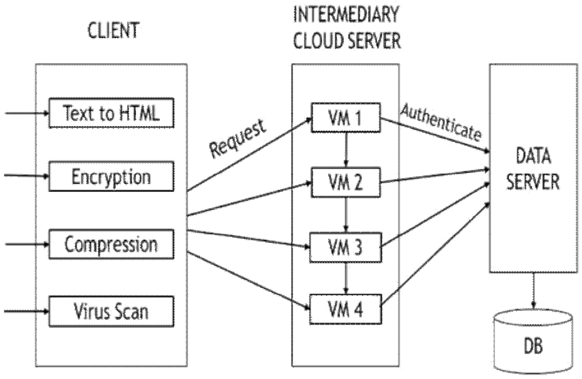

Fig. 3.1 System Overview

to give verification. An Arrangement determines the Login Module(s) to be utilized with a specific login application. Diverse Login Modules can be connected to under the application without requiring any alterations to the application itself. Each Login Module is instated with a Subject, a Callback Handler, shared Login Module state. The Subject speaks to the client or administration right now being verified and is refreshed by a Login Module with significant Principals and qualifications if verification succeeds. Login Modules utilize the Callback Handler to speak with clients (to provoke for client names and passwords, for instance), as depicted in the login technique portrayal. Algorithm used is **Status Update**

Algorithm: Status Update

```
a. START
b. Send Authentication Key from Cloud to Service
     Manager
c. If key is valid
   Update cloud server Login status as valid
d. Else
   Update Login status as invalid
e. END
```

This is an entity that requests data from the data server. The client process initiates the connection to the server and makes a service request to the server. Some of the requests are Text to Html, Text Encryption, File Compression and spam filter

.Gets back the response from the proxy servers or the main server itself.

Load balancing can be formulated as allocating N={N1,N2,N3,N4} number of requests submitted by user to M number of Intermediary processing units in the Cloud, where M={M1,M2,M3,M4}.Each of the intermediary processing unit will have a processing unit vector (PUV) indicating current status of unit utilization. This vector consists of MIPS, indicating how many million instructions can be executed by that processing unit per second, α is cost of execution of instruction and delay L.

PUV = f (MIPS, α, L) ………..Equation (1)

Similarly each request submitted by user can be represented by a request unit vector (RUV). Thus the attribute of different requests can be represented by equation (2).

RUV = f (t, NIR, AT, wc)…………Equation (2)

Where, t represents the type of request such as text to html, encryption, compression and Virus scan, NIR represents the number of instructions present in the request, AT is Request arrival time, wc is worst case completion time. To allocate these N requests among M number of machines, the Fitness function ζ is as indicated in equation 3, is minimized.

$\zeta = w1 * \alpha (NIC \dagger MIPS) + w2 * L$……...Equation (3)

Where w1 and w2 are predefined weights.

The ***proposed algorithm*** is as given below:

i. Randomly initialize a population of processing units after encoding them into binary strings [Start].
ii. Evaluate the fitness value of each population using Equation 3 [Fitness].
iii. While either Algorithm maximum number of iterations are Exceeded or optimum solution is found.
Do:
(a): Consider chromosome with lowest fitness twice and eliminate the chromosome with highest fitness value to construct the mating pool [Selection].
(b): Perform single point crossover by randomly selecting the crossover point to form new off spring [Crossover].
(c): Mutate new off spring with a mutation probability of (0.05) [Mutation].
(d): Place new off spring as new population and use this population for next round of iteration [Accepting].
(e): Test for the end condition [Test]
iv. End

3.2 Server1 Module:

Text file is converted to HTML format that can be placed on web page. We use **Text to Html** algorithm

Algorithm:Text to Html
Input: text file
Output: html file

1: declare a string variable "filename" and get the input by using inbuilt class scanner
2: declare a string variable "line" and initialize it to null
3: declare an object of inbuilt class FileReader and pass the input through the constructor and wrap the object in BufferedReader class
4: read the contents of file unit line =null and create an object of StringBuffer class and compute the length of each line at a time
5: convert each line into array of characters and check each character to convert it into the formatted html tags and return to the string buffer
6: create an object of class FileWriter and wrap the object into the BufferedWriter class
7: the processed contents are written into another file using the FileWriter object
8: close BufferedWriter and BufferedReader

3.3 Server2 Module:

The plaintext is encrypted using Data Encryption Standard algorithm and a symmetric key. Data Encryption Standard Algorithm is as follows:

Input: Plain Text

Output: Cipher Text

1: Divide the Plain Text into 64-bit blocks

2: Take blocks one by one and implement DES

3: DES involves two operations on Plain Text and Key that run simultaneously

4: Initial operation on Key [In 64-bit original Key, after discarding every 8th bit gives 56-bit Key]

5: Initial operation on first block of Plain Text 64-bit Plain Text, on initial permutation gives Initial Permutation Text

6: Divide Initial Permutation Text into LPT (32-it) and RPT (32-bit)

7: Divide Key into Ckey and Dkey (Inside 16 rounds blocks)

8: Circular left shift on Ckey and Dkey gives shifted Ckey (28-bit) and shifted Dkey

9: Combining the shifted Ckey and shifted Dkey gives shifted key (56-bit)

10: Expansion permutation on RPT (32-bit) gives Expanded RPT (48-bit) and Compression Permutation on shifted key (56-bit) gives Compressed Key (48-bit)

11: Performing XOR operation on Expanded RPT (48-bit) and Compressed Key (48-bit) gives XoredRPT (48-bit)

12: Performing SBox substitution on Xored RPT (48-bit) gives SBox RPT (32-bt)

13: Performing PBox permutation on Sbox RPT gives PBox RPT (32-bit)

14: Perform Pbox RPT XOR LPT and Swap

15: Continues for 15 more time

16: Combining LPT and RPT and performing Final Permutation gives 64-bit Cipher Text.

3.4 Server3 Module:

File Compression is a process of „packaging‟ file or files to use less disk space. Zip Compression Scheme is used to compress files with no loss of information. It keeps related files together and makes downloading easy and faster.

Input: file

Output: zip file

Compression steps are:

1. The matching and replacement of duplicate strings with a back reference, linking to previous location of that identical string.
2. Replacing symbols with new, weighted symbols based on frequency of use.
3. An end of Central Directory Record is placed at the end of a zip file, which identifies what files are in the zip and where in the zip the files is located.
4. Zip Files keeps related files together and makes downloading faster and easy.

Example: Suppose there is a line in the word "Ask not what your country can do for you – ask what you can do for your country". This line occupies 79Bytes of memory. Repeated words or patterns are picked and put into numbered index which are included in dictionary. The compressed line would be "1 not 2345 – 12354".This line occupies 59Bytes of memory [18 unit‟s memory of sentence + 41 units of dictionary].

3.5 Server4 Module:

Recognizes the spontaneous and undesirable messages and keep those messages from entering the customer inbox utilizing a Whitelist strategy.

Input: folders

Output: virus free folders

1. Scans the filenames in the folder comparing the specific bits of the filenames extension against information in the database.
2. Follows Blacklisting Method, where the .exe and .imi files are been blacklisted.
3. If the filename matches with the blacklisted files in the database, it is considered as virus and that particular file is deleted from folder and returned back to client.

3.6 Service Provider

This is an entity that originally stores the data requested by the client. The server waits for the request from the client and communicates with other intermediary modules in order to serve the client request.

The two sub-modules are:

a) Process Requesting:

It provides methods for processing of the task. It contains all the activities done by the intermediary modules. The proxies request the server, if it fails to process the task.

b) Process Delegation:

It involves the server and the intermediary servers to whom the task is assigned using the Genetic algorithm and if in case the intermediaries fail, the alternative is assigned using multi-agent Algorithm.

3.7 Multiagent Algorithm

(Self-organized**)**

1. ***Start***
2. During processing, check if processing unit is available go to Step3, if not available go to Step4
3. Process the request by using Genetic algorithm
4. If suppose PU fails and it is unavailable it requests its alternative VM to complete the request. (directing the task to its alternative)
5. Suppose if the alternative PUs also fails, then the server completes the request.
6. ***End.***

Genetic algorithms differ from traditional search and optimization methods in four significant points:

1. Hereditary calculations look parallel from a populace of focuses. In this way, it can abstain from being caught in nearby

ideal arrangement like customary strategies, which seek from a solitary point.
2. Hereditary calculations utilize probabilistic determination rules, not deterministic ones.
3. Hereditary calculations chip away at the Chromosome, which is encoded variant of potential arrangements' parameters, rather the parameters themselves.
4. Hereditary calculations utilize wellness score, which is acquired from target capacities, without other subsidiary or helper data

3.8 Advantages of Genetic Algorithm:

1. Does not require any subordinate data (which may not be accessible for some true issues).
2. It is quicker and more productive when contrasted with the conventional techniques.
3. Has great parallel abilities.
4. Enhances both ceaseless and discrete capacities and furthermore multi-target issues.
5. Gives a rundown of "good" arrangements and not only a solitary arrangement.
6. Continuously finds a solution to the issue, which shows signs of improvement over the time.
7. Helpful when the pursuit space is vast and there are a substantial number of parameters included.

4 CONCLUSION

The need for Content Distribution between the intermediaries is to eliminate requests for re-transmissions and to avoid infinite loop of request-response that causes duplication of data in turn leading to congestion. We have proposed the genetic algorithm and Multi-agent algorithm, in which each intermediary‟s acts as intelligent-agents and process the requests.

REFERENCES

[1] “Client-side load balancer using Cloud”, Sewook Wee Accenture Technology Labs 50 W. San Fernando Street, Suite 1200 San Jose, California, USA sewook.wee@accenture.com Huan Liu Accenture Technology Labs 50 W. San Fernando Street, Suite 1200 San Jose, California, USA huan.liu@accenture.com

[2] “Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud”, Daniel, Warneke, Member, IEEE, and Odej Kao. IEEE transactions on parallel and distributed systems, vol. 22, no. 6, June 2017

[3] “A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment”, Jinhua Hu, Jianhua Gu, Guofei Sun, Tianhai Zhao.

[4] “AI efficient & secure content Cloud based Architecture for Opinion Mining”, IEEE International Workshop on Management in Cloud Computing (MCC 2017), July 2017, Washington Stella Gatziu Grivas , Marc Schaaf , Michael Kaschesky , Guillaume Bouchard . Keywords: Cloud Services, Operations and Management, Event-driven Architecture, Policy-based Service Management.

[5] “Collaborative Proxy System for Distributed Web Content Transcoding”, Proc. Ninth ACM Int‟l Conf. Information and Knowledge Management (CIKM ‟00), Nov. 2000, V. Cardellini , P.S. Yu, and Y.W. Huang.

[6] “A reliable dynamic scheduling algorithm based on Bayes trust model”, Wei Wang, Computer Science, 2007.

[7] “Task Scheduling in parallel and Distributed System”, Englewood Cliffs, Rewinin H E, Lewis T G, Ali H H New Jersey: Prentice Hall,1994, pp. 401-403.

[8] “A programming aid for message passing system”, Wu M, Gajski D, IEEE Trans Parallel Distributed Systems, 1990, pp. 330-343.

[9] “Scheduling Precedence Graphics in Systems with Interprocessor communication times”, Hwang J J, Chow Y C, Anger F D, Siam J Comput, 1989, pp. 244-257.

[10] “Scheduling Parallel Programs onto Arbitrary Target Machines”, Rewinin H E, Lewis T G, J Parallel Distributed Compute,1990 , pp.138-153.

[11] “A Compile Time Scheduling heuristic for Interconnection- constraint heterogeneous processor Architectures”, IEEE Trans Parallel Distributed Systems, 1993, pp.175-187.

[12] “Caching on the World Wide Web”, IEEE Trans. Knowledge and Data Eng., vol. 11, no. 1, pp. 94- 107, Jan. 1999, C. Aggarwal, J.L. Wolf, and P.S. Yu. https://doi.org/10.1109/69.755618

[13] “Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing” ,Qian Wang, Student Member, IEEE, Cong Wang, Student Member, IEEE, Kui Ren, Member, IEEE, Wenjing Lou, Senior Member, IEEE.

[14] “An Intelligent Load Balancing Algorithm Towards Efficient Cloud Computing” , in Proc. of AI for Data Center Management and Cloud. Yang Xu, Lei Wu, Liying Guo, Zheng Chen,Lai Yang Zhongzhi Shi. Computing: Papers, from the 2011 AAAI Workshop (WS-11- 08), pp. 27–32, 2008.

[15] “The Existential Pleasures of Genetic Algorithm”, E Goldberg, In: Genetic Algorithms in Engineering and Computer Science, Winter Ged, New York: Wiley, 1995, pp.23-31.

Rakshitha R: pursuing B.E in CSE, EWIT(VTU), Bengaluru. Her areas of Interest are Artificial Intelligence, Programming the Web, Cloud Computing, Computer Networks, Big Data, Java programming etc.

Rakshita R: pursuing B.E in CSE, EWIT(VTU), Bengaluru. Her areas of Interest are Artificial Intelligence, Programming the Web, Cloud Computing, Computer Networks, Databases, Java programming etc.

Ramya S: pursuing B.E in CSE, EWIT(VTU), Bengaluru. Her areas of Interest are Artificial Intelligence, Programming the Web, Cloud Computing, Computer Networks, Software Engineering, Java programming etc.

Ranjitha Shivamatta: pursuing B.E in CSE, EWIT(VTU), Bengaluru. Her areas of Interest are Artificial Intelligence, Programming the Web, Cloud Computing, Computer Networks, Computer Security, Java programming etc.

Prof Kiran M: Assistant professor, Department of Computer Science and Engineering, EWIT(VTU), Bengaluru. Qualification: B.E, M.Tech. His areas of Reasearch are Artificial Intelligence, Wireless adhoc Networks, Computer Networks, Software Engineering, Genetic Algorithms, Machine Learning, IoT and Cloud Computing.