Scientific Journal Impact Factor (SJIF): 5.278
International Journal of Modern Trends in Engineering and Research
IJMTER
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

IMPLEMENTATION OF : COMMUNICATION OVERHEAD REDUCTION FOR REMOTE DATA SHARING TO PREVENT DATA PRIVACY ON CLOUD STORAGE

Kajal V.Dabhade[1], Shridevi Ganacharya[2], Manjusha Borade[3] and Prof.Madhuri Mali[4]
[1,2,3,4]computer department sknsits,lonavala

Abstract- Distributed file searching in delay tolerant networks formed by mobile devices can potentially support various useful applications. In such networks, nodes often present certain social network properties of their holders in terms of contents (i.e., interests) and contacts. However, current methods in DTNs only consider either content or contact for file searching or dissemination, which limits the file sharing efficiency. In proposed System, we develop an efficient file sharing system based on a mixture distribution model working in the BitTorrent like p2p networks. The Bit- Torrents built-in unchunking mechanism delays the initial file sharing process for newly joined peers as well as brings the problem of free-riding that peers only download from others but never contribute to the network. We demonstrate a file sharing mechanism for BitTorrent. File will be shared in chunks using BSW(Basic Sliding Window) algorithm also providing the support of Multi- threading which allows load balancing. By multi-threading we will be exploiting all the available CPU cores to gain the X4(in case of good core) increase in the chunking performance. The load balancer is a technology that uses multiple machines on a network. This technology is used to prevent client access to resources during traffic on a network. The process multiple machines on a network (MMN) on network is used to solve the problem of server failure.

Keywords- Big sensing data**,** Cloud computing**,** Data chunk**,** Data compression**,** Data flow concurrency**.**

I. INTRODUCTION

For proposing an online resource management framework that maximizes profit ratio while minimizing energy expenses by exploiting the distributed task elasticity and price heterogeneity. This is done by reducing the duration during which servers need to be left ON and maximizing the monetary revenues when the charging cost for some of the in elastic tasks depends on how fast these tasks complete, while meeting all resource requirements. The power supply and the core speed are increased when there are more tasks in server, such that tasks can be processed faster and the average task response time is reduced. It is possible to design a multicore server processor with workload dependent dynamic power management, such that its average task response time is shorter than a multicore server processor of constant speed (i.e., without workload dependent dynamic power management). Byte Rotational Algorithm(BRA) provides more security and takes smallest amount of time for transfer file . This algorithm can apply on different types of files like text, image, audio, video files.

Comparative studies conducted using UCI repository data traces show the effectiveness of our proposed framework in terms of improving resource utilization, reducing energy expenses, and increasing profits ratio by calculating memory and bandwidth with increasing speed. The process elasticity is exploited on heterogeneous environment in a distributed system. Elasticity is the degree to which a system is able to adapt to workload changes by provisioning and deprovisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible. Various approaches are considered like serial, parallel and hybrid approaches. Accordingly, profit ratio is calculated. Process mining is taken as a task to calculate the profit ratio. Resources that are considered are CPU, Bandwidth, Time and Temperature and Memory. Tools used to calculate the profit ratio of CPU, Bandwidth, Time and Temperature and Memory is CPU-Z and HW-Monitor. The tasks involved are independent on each other. Profit ratio is calculated of factors i.se CPU, Bandwidth, Memory, Time and Temperature of systems with different processors.

II. LITERATURE SURVEY

Computational efficiency of data centers and cloud infrastructures has been considered together with energy-efficient and environment-conscientious resource scheduling and optimization. Recent survey of green data centers and energy-efficient cloud computing systems can be found[7].The proposed scheme reduces the queuing delay of the cloud tasks by accounting for their execution time lengths. We also derive bounds on the average queuing delays , and evaluate the performance of our proposed scheme and compare it with those achievable under existing schemes by relying on real Google data traces[4].

Modern multicomputer interconnection networks offer the delivery of messages with very low latency. However the message in-flight time is only a small portion of the total time that is required to send a message from source to destination. For fine to medium grained message sizes, the majority of time is spent in overheads for setting up and managing message transmission. It is often

possible for compilers/programmers to separate inter-processor communication traffic into messages that exhibit communication locality and messages that do not[9].

Cloud Computing has transformed the software support for large systems from server to service oriented paradigm. This drift has evolved new challenges for design and delivery of services over heterogeneous requirements and environments. This brings about risks and challenges for systems. The system over internet is vulnerable to performance and security risks. The performance is a composite evaluation but risks that are related to privacy can be handled at different levels of abstraction in cloud model [10].

IaaS providers offer their customers the illusion of unlimited compute, network, and storage capacity often coupled with a frictionless registration process where anyone with a valid credit card can register and immediately begin using cloud services. Some providers even offer free limited trial periods. By abusing the relative anonymity behind these registration and usage models, spammers, malicious code authors, and other criminals have been able to conduct their activities with relative impunity. PaaS providers have traditionally suffered most from this kind of attacks; however, recent evidence shows that hackers have begun to target IaaS vendors as well. Future areas of concern include password and key cracking, DDOS, launching dynamic attack points, hosting malicious data, botnet command and control, building rainbow tables, and CAPTCHA solving farms[11].

Cloud computing providers expose a set of software interfaces or APIs that customers use to manage and interact with cloud services. Provisioning, management, orchestration, and monitoring are all performed using these interfaces.

The security and availability of general cloud services is dependent upon the security of these basic APIs. From authentication and access control to encryption and activity monitoring, these interfaces must be designed to protect against both accidental and malicious attempts to circumvent policy. Furthermore, organizations and third parties often build upon these interfaces to offer value- added services to their customers. This introduces the complexity of the new layered API; it also increases risk, as organizations may be required to relinquish their credentials to third parties in order to enable their agency[11].

Volume: 5, Issue: 05, [May– 2018] **ISSN (Online):2349–9745 ; ISSN (Print):2393-8161**

The problem of scheduling batch jobs for multiple geographically distributed data centers [9],and proposed a provably efficient online scheduling algorithm, which optimizes the energy cost and fairness among different organizations, subject to queuing delay constraints, while satisfying the maximum server inlet temperature constraints[8].A novel energy-aware scheduling algorithm for real time, a periodic, independent tasks, and proposed two strategies in terms of resource scaling up and scaling down to make a good trade-off between task schedulability[6] and energyconservation

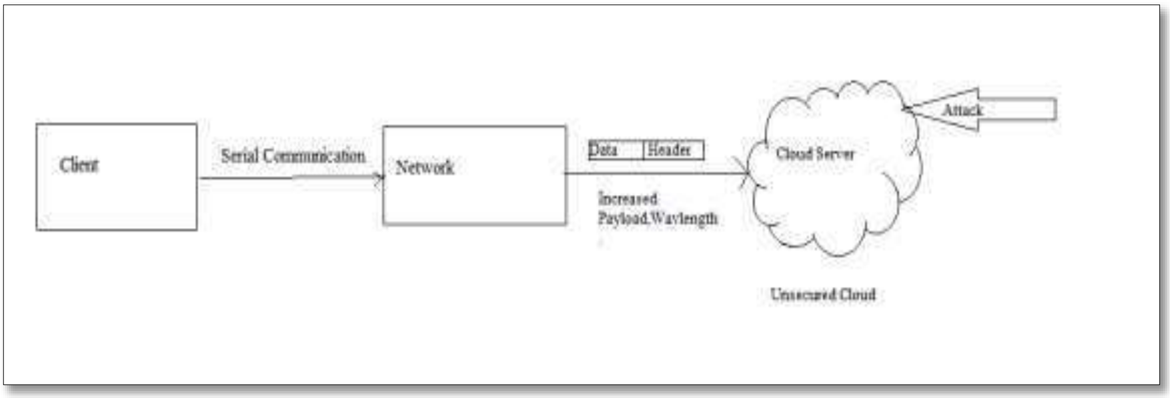


Figure 1:-Existing System

III. EXISTING SYSTEM

As OSI Layer is responsible for data sharing and communication for now it increases the overhead. So the communication time is also increased. It makes delay in communication. Ex. Huge data sharing(Google data sharing). The packet is the whole data that is to be transmitted. This packet consists of the actual data which is the payload and the source/destination information of the packet is in the header. So a packet consists of header and payload. So what is this overhead?. is overhead a part of the header?. "Packet overheard includes all the extra bytes of information that are stored in the packet header" the header already contains source/destination info. What are the extra bytes of information that this packet overhead has?. The "payload" which is the data itself it needs to transfer (usually the user's data), the "header" contains various things depends on the protocol you are using, for example UDP contains just simple things in the header like Destination and Source IP/PORT, TCP on the other end contains more things like the sequence number of the packet to ensure ordered delivery, a lot of flags to ensure the packet actually received in it's destination and checksum of the data to make sure it didn't get corrupted and received correctly in it's destinations. Now, the "overhead" part is actually the additional data that you need in order to send your payload. In the cases I talked about above it's the header part, because you need to add it to every payload that you want to send over the internet. TCP has bigger overhead than UDP because it needs to add more data to your payload, but you are guaranteed that your data will be received in it's destination, in the order you sent it and not corrupted. UDP does not have this features so it can't guarantee that. The *overhead* of a packet type is the amount of wasted bandwidth that is required to transmit the payload. The packet header is extra information put on top of the payload of the packet to ensure it gets to its destination. The overhead is variable because you can choose a different type of packet (Or packet protocol) to transmit the data. Different packet protocols give you different features. The two key type of packet protocols that exist today are TCP and UDP.One can say UDP has a lower overhead than TCP because its packets have a smaller header and therefore take less bandwidth to send the payload (The data).The reasons for this are a deep subject but suffice to say that TCP provides many very useful features that UDP does not, such as ensured delivery of the packets and corruption detection. Both are very useful protocols and are chosen based on what features an application needs (Speed or reliability).

Volume: 5, Issue: 05, [May– 2018] ISSN (Online):2349–9745 ; ISSN (Print):2393-8161

A. Reliable vs. Unreliable Communication

The terms reliable and unreliable don't refer to whether it works or not. It refers to whether something is done to guarantee.

i. RELIABLE

End stations running reliable protocols will work together to verify the transmission of data to ensure accuracy and integrity of the data. A reliable system will set up a connection and verify that: all data transmitted is controlled in an orderly fashion, is received in the correct order and is intact. Reliable protocols work best over physical medium that loses data, and is prone to errors. The error correction, ordering and verification mechanisms require overhead in the data packets and increase the total amount of bandwidth required to transmit data. Transmission Control Protocol (TCP) is a typical reliable protocol. TCP often usually adds an average of 42-63 bytes of overhead to datagram‟s. For a Telnet connection which transmits each keystroke individually, this is horribly inefficient because up to 64 bytes of data are transmitted to communicate just 1 byte of useful information.

ii. UNRELIABLE

Unreliable protocols make no effort to set up a connection, they don't check to see if the data was received and usually don't make any provisions for recovering from errors or lost data. Unreliable protocols work best over physical medium with low loss and low error rates. User Datagram Protocol (UDP) is an example of an unreliable protocol. UDP makes no provisions for verifying whether data arrived or is intact. However, UDP adds a minimum of overhead when compared to TCP and is thus much faster for data transfers over high quality physical links that are high speed and exhibit little or no errors in communication.

IV. PROPOSED SYSTEM

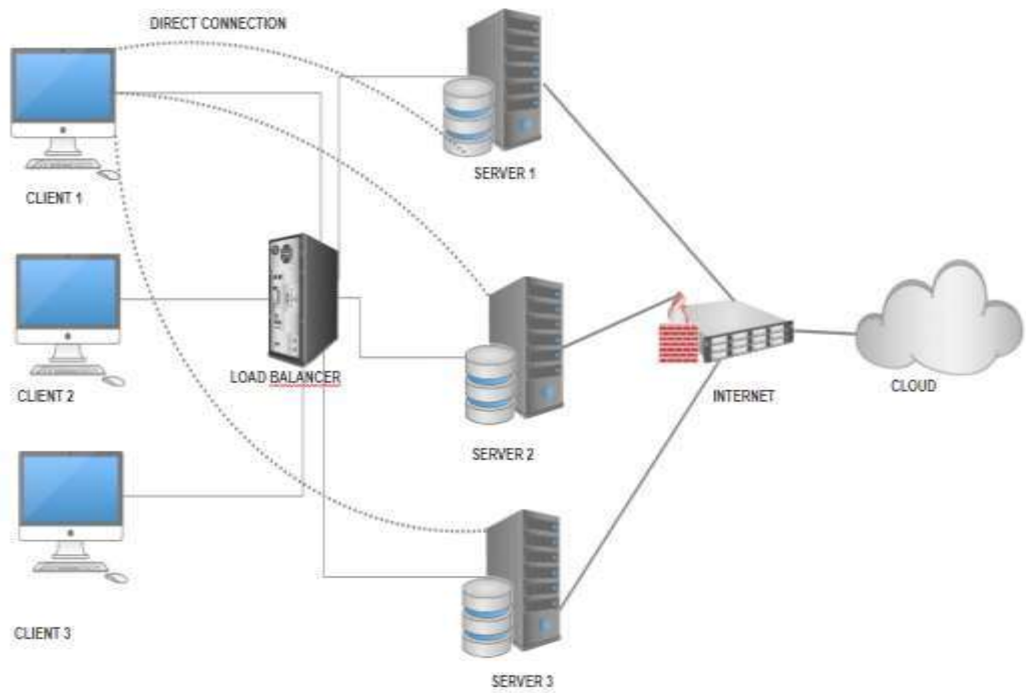


Figure 2:-Proposed System

As there is a large amount of data present for data sharing fields like marketing, sales, Banks customer support, e-commerce such database having a size in GB and TB need fast processor. Serial approach can consume time and reduce performance. To solve this issue's the proposed model of parallel optimize system to reduce time, increase performances and fast processing based on OSI layer Approach. For fast processing multicore processor are used. For finding useful knowledge an algorithm is required. Byte rotational algorithm is fast Cryptosystem approach which complex to hackers In the Byte Rotation Algorithm involve two techniques. One is key generation technique is used. And second is data processing .We plan to implement cryptosystem in single threaded approach as well as multi-threading technique.

The main issue in existing was the overhead here we will be diagnosing the problem of communication overhead at the data link layer in OSI Model used in the network for data sharing. As the each layer adds the overhead in the frames while travelling from source to the destination so it increases the size and overhead which will increase the wavelength, payload, delay time, etc.So the Proposed system will be the faster approach of data sharing in the network. As now we are using the cloud for storage purpose. Hence they are not yet that much secured. They may hacked by the hackers using various malicious attacks on the cloud. So the essential data may be lost. So here we will be providing Remote data integrity checking (RDIC) enables a data storage server, say cloud servers, to prove to a verifier that it is actually storing a data owner's data honestly. Here we also add the Feature like Load balancer as shown in the Figure2.Which will balance the load on the servers. Which will be efficient for the network to handle the requests? The Client will send the request serially/parallely to store/share the data in the network which will be in the sent in the chunks format with the secure key encapsulated in the frame while sending so in the middle of the network attack will get prevented. In each of the frame there will be security key so it will preserve the data security. At receiver end he/she will get to know the security key for decrypting the data. Now after broadcast request it will go to the load balancer which will check the load present at each servers and will inform which servers load is less. So the request will go further towards that server which will receive the data avoiding the delay of receiving the data. The shared data may at risk in the cloud so the cloud will be also preventing the attacks by the unauthorized users to preserve the data privacy and security.

V. METHODOLOGY

In Chunking mechanism, it break a file into fixed size pieces (typically 256 KB each), and exchange pieces with each other. Once chunks are downloaded they are grouped together in a file at both the ends. In order to transfer data over TCP connection at full capacity, pipelining technique is used. To achieve this, a piece is broken further into sub-pieces, typically of 16KB in size, which is named chunks or blocks. A number of request for sub-pieces, typically five, are kept pending in a pipeline. Each chunk arrival will trigger a request for another chunk. By this way, the file transfer connection is kept busy most of the time and the each piece is downloaded fast. When some chunks (sub-pieces) are received from a specific piece, then the rest chunks of that piece are requested before chunks from any other pieces. In other words, peers will not request another piece until the current piece is downloaded completely. The reason of strict priority is that only a full copy of piece can be traded between peers. Once peers finish downloading the current piece, it will select the next piece which is the fewest among its neighbors. This mechanism is named rarest first. At the end of download, a requested piece is sometimes transferred with a very slow rate. This is not a problem in the middle of download, but could be a matter to delay a peer's finish. To solve this problem, a peer sends request to all of its neighbors for all chunks of the last piece. As soon as a chunk is received, the request of that chunk is cancelled to prevent redundant downloads. In this way, the last piece of a file is downloaded quickly.

In the context of voice over IP, jitter is the variation in delay of packets received, caused by network congestion or route changes. Jitter is a vital quality of service (QoS) factor in evaluation of network performance. It is one of the significant issues in packet based network for real time applications. The variation of interpacket delay or jitter is one of the primary factors that agitates voice quality .Jitter plays a vital role for the measurement of the Quality of Service of real time applications. A voice message is mostly tightly coupled within a group of consecutive packets. Therefore, the proposed algorithm suggests chunking of the packets at the destination in the buffer. Thus, playing of a chunk, mostly, conveys a complete message without something missing. Hence, the QoS is improved while playing it out. All chunks are of the same size.

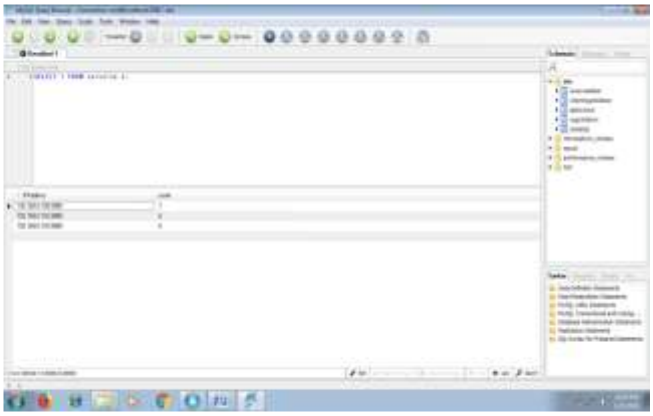
Load Balancing: - load balancer a technology uses multiple machines on a network. This technology is used to prevent client access to resources during traffic on a network. In this system to solve the problem of a server failure the process is used is multiple machines on a network (MMN)

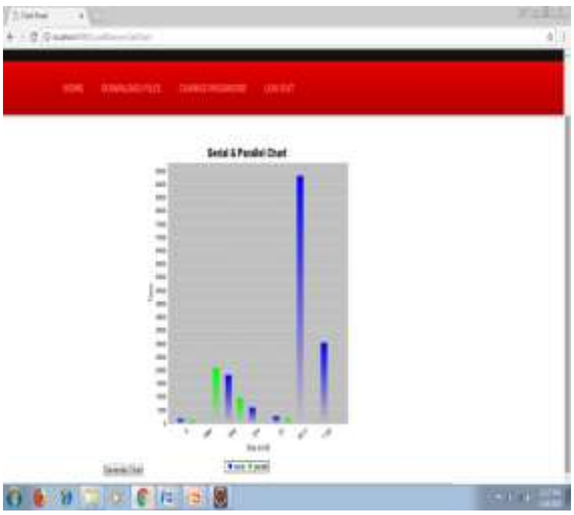
VI. WORKING

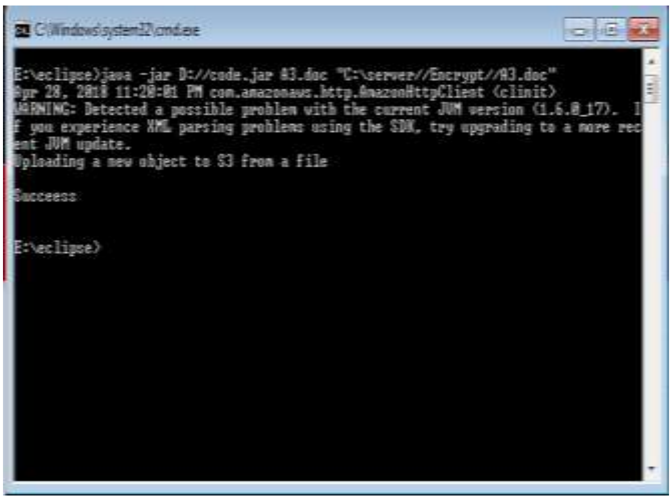
Basically, working of our model is divided in three parts Loadbalncer, Master-slave based architecture and data privacy, integration on cloud storage. Loadbalancer works on server to avoid the server crash, whenever the huge amount of data is shared over the network the chances of server failure/crash is there so loadbalancer will be the key feature to overcome this server failure, when client downloads any file the request is forwarded to the loadbalancer.

It will check the load on each server and the load on which server is minimum that server will address that request and the server will synchronizes the data on each server. Then via internet the data is forwarded to the cloud while forwarding the data it maintains the integrity,privacy of data using encryption technique, here we used AES encryption technique so that security of data is maintened.

VII. RESULTS







VIII. FUTURE WORK

Future work of this project will be an android app which will be upload the data automatically.

IX.CONCLUSION

A proft-driven online resource allocation framework for elastic task requests is proposed. The framework exploits the elasticity and the varying charging costs among the submitted requests and decides where to place the heterogeneous submitted task requests, and how much resources should be allocated to the elastic ones such that the cloud profits are maximized while meeting all tasks demand. The various algorithms like chunking algorithm, byte rotational algorithm and TTTD algorithms are discussed. TTTD algorithm, not only successfully achieves the significant improvements in running time and average chunk-size, but also obtains the better controls on the variations of chunk-size by reducing the large-sized chunks. Byte rotational algorithm is fast encryption algorithm. Byte Rotational Algorithm is complex to hackers but it's a strong algorithm in case of security.

X. ACKNOWLEDGMENT

It gives us great pleasure in presenting the preliminary Survey Paper on "Survey on:Communication Overhead Reduction for Remote Data Sharing to Prevent Data Privacy". I would like to take this opportunity to thank my internal guide Prof. Madhuri Mali for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.

REFERENCES

[1] IEEE TRANSACTIONS ON COMPUTERS, VOL. 65, NO. 8, NOVEMBER 2016 "Reducing Communication Overhead of the Subset Difference Scheme". Sanjay Bhattacherjee and Palash Sarkar
[2] IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 12, NO. 4, APRIL 2017 Identity-Based "Remote Data Integrity Checking With Perfect Data Privacy Preserving for Cloud Storage". Yong Yu, Man Ho Au, Member, IEEE, Giuseppe Ateniese, Xinyi Huang, Willy Susilo, Yuanshun Dai, and Geyong Min
[3] J. Doyle, R. Shorten, and D. OMahony, "Stratus: load balancing the cloud for carbon emissions control", IEEE Transactions on Cloud Computing, vol. 1,no. 1, pp. 116-128, 2013.
[4] M. NoroozOliaee, B. Hamdaoui, M. Guizani, \Online multiresource scheduling for minimum task completion time in cloud servers," in Computer Communications Workshops, 2014 IEEE Conference
[5] X. Zhu, L. T. Yang, H. Chen, J. Wang, S. Yin, and X. Liu, "Real time tasks oriented energy-aware scheduling in virtualized clouds"
[6] V. Maitri, Dattatray S. Waghole, Vivek S. Deshpande, IEEE Senior Member, \Low latency for _le encryption and decryptionusing BRA algorithm in network security", 2015 International Conference on Pervasive computing.
[7] Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems "
[8] M. Polverini, A. Cianfrani, S. Ren, and A. V. Vasilakos, "Thermal-aware scheduling of batch jobs in geographically distributed data centers".
[9] B. Vien Dao ; S. Yalamanchili ; J. Duato" Architectural support for reducing communication overhead in multiprocessor interconnection networks High-Performance Computer Architecture, 1997., Third International Symposium on
[10] Tunisha Saxena ; Vaishali Chourey "A survey paper on cloud security issues and challenges". IT in Business, Industry and Government (CSIBIG), 2014 Conference on
[11] Cloud Security Alliance. (2010). Top Threats to Cloud Computing.[Online]. Available http://www.cloudsecurityalliance.org

