ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 5, Issue 2, pp: (12-17), Month: April - June 2017, Available at: www.researchpublish.com

# Bloom Filter Check: Integrated Network Construction with Dynamic Data Retrieval

K.Sangeetha[1], P.Sathiya[2], S.Vinitha[3], R.Visalakshi[4]

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

***Abstract:*** **The main problem nowadays we facing today is retrieval of significant data and with less consumption of time. So, it's not an efficient system. Hence the system designed with cluster, nodes and master site. Cluster with the number of nodes and the Network with the number of clusters. Moreover user firstly updates the data to their respective nodes then to the master site this maintains of data is called bloom filter. Whereas, search will be done by the replication to the closer node if not then proceed to the cluster if not then to the master site based on popular file replicate strategy (PFRF). Instead of storing the data in all the nodes system uses hash value and then retrieved. Retrieved data packets will be encrypted.**

***Keywords:*** **less consumption of time, Bloom filter check, integrated network construction.**

## 1. INTRODUCTION

Network grid comprises of the storage and information about the data in usage and are distributed in wide area network (WAN). This network grid provides the data into the requested node. This network grid provides the collection of data even at remote area where the users needs to access the large volume of data from the farther node. This will consumes large amount of bandwidth. So, create an replication of the same file and its replicated to all the nodes so the user can access the file from its own storage because of replication, this will reduce the time and bandwidth. Many researchers proposed that how to select the best nodes for replication. Nevertheless, none of these strategies are round-based and they also do not take into account the variation that might occur in user behavior. Dividing the time into rounds usually leads to a better decision on which files to replicate because this decision is made after a large number of file requests and therefore the users will determine more accurately which files need to be kept in their storages. In real situation, the user behavior might change since users sometimes change their interests in files. PFRF strategy fills those gaps by dividing the time into rounds and at the end of each round it calculates the popularity of different files. Then, it only replicates a percentage of the most popular files to different clusters. On the other hand, PFRF strategy has a number of drawbacks. First, this strategy does not determine to which cluster node the file is replicated. Therefore, a number of factors have to be used in that determination such as number of requests, free storage space, and node centrality. Second, this strategy only considers the number of requests to determine the file popularity (importance). However, there are other important factors to determine the popularity of a file such as how many times it was requested in the last round and the file size. Third, in PFRF, the average popularity for a file is defined as the sum of the popularities of the file only in the clusters having it divided by how many clusters having this file. However, some clusters might not have the file but have a high request rate for it. Therefore, the average popularity for a file is better to be defined as the sum of the popularities of the file in all clusters divided by the number of clusters in the Data Grid. Eventually, in case of replicating a file to a cluster node with no enough free storage space, PFRF compares the popularity of the new file to the popularity of each stored file separately, while in our proposed strategy, the popularity of the new file is compared to the sum of popularities of a group (one or more files depending on the space still needed to store the new file) of files at once. In this paper, a data replication strategy named Improved PFRF (IPFRF) is proposed. This strategy is based on PFRF but overcomes its drawbacks.

## 2. LITERATURE SURVEY

**Dynamic Replica Management for Data Grid:**

K. Sashi and Dr. Antony Selvadoss Thanamani proposed that Data grids provide distributed resources for dealing with large scale applications that generate huge volume of data sets. Data replication, a technique much discussed by data grid researchers in past years creates multiple copies of file and stores them in conventional locations to shorten file access times. One of the challenges in data replication is creation of replicas, replica placement and replica selection. Dynamic creation of replicas in a suitable site by data replication strategy can increase the systems performance. When creating replicas a decision has to be made on when to create replicas and which one to be created. This decision is based on popularity of file. Placement of replicas selects the best site where replicas should be placed. Placing the replicas in the appropriate site reduces the bandwidth consumption and reduces the job execution time. Replica selection decides which replica to locate among many replicas. This paper discusses about dynamic creation of replicas, replica placement and replica selection. It is implemented by using a data grid simulator, Optorsim developed by European data grid projects.

**A New Dynamic Replication Strategy for Data Grids:**

Feras Hanandeh1, Mutaz Khazaaleh2, Hamidah Ibrahim3, and Rohaya Latip3 proposed that Data grids are currently proposed solutions to large scale data management problems including efficient file transfer and replication. Large amounts of data and the world-wide distribution of data stores contribute to the complexity of the data management challenge. Recent architecture proposals and prototypes deal with dynamic replication strategies for a

high-performance data grid. This paper describes a new dynamic replication strategy called Constrained Fast Spread (CFS). It aims to alleviate the main problems encountered in the current replication strategies like the negligence of the storage capacity of the nodes. The new CFS strategy enhanced the fast spread strategy by concentrating on the feasibility of replicating the requested replica on each node among the network.

**A Novel Data Replication Policy in Data Grid:**

Y. Nematii proposed that Data grid aims to provide services for sharing and managing large data files around the world. The challenging problem in data grid is how to reduce network traffic. One common method to tackle networktraffic is to replicate files at different sites and select the best replica to reduce access latency. This can generate many copies of file and stores them on suitable place to shorten the time of getting file. To employ the above two concepts, in this paper we propose a dynamic data replication strategy. Simulation Results with Optorsim shows better performance of our algorithm than former ones.

**An Improved Dynamic Data Replica Selection and Placement in Hybrid Cloud:**

A.Rajalakshmi, D.Vijayakumar, Dr. K .G. Srinivasagan proposed that Cloud computing platform is getting more and more attentions as a new trend of data management. Data replication has been widely used to speed up data access in cloud. Replica selection and placement are the major issues in replication. In this paper we propose an approach for dynamic data replication in cloud. A replica management system allows users to create, register and manage replicas and update the replicas if the original datasets are modified. The proposed work concentrates on designing an algorithm for suitable optimal replica selection and placement to increase availability of data in the cloud. Replication aims to increase availability of resources, minimum access cost, shared bandwidth consumption and delay time by replicating data. Our approach based on dynamic replication that adapts replica creation continuously changing network connectivity and users. The proposed systems developed under the Eucalyptus cloud environment. The results of proposed replica selection algorithm achieve better accessibility compared with other methods.

**Data Replication Strategies In Wide Area Distributed Systems:**

Sushant Goel Rajkumar Buyya proposed that Effective data management in today's competitive enterprise environment is an important issue. Data is information; and information is knowledge. Hence, fast and effective access to data is very important. Replication is one such widely accepted phenomenon in distributed environment, where data is stored at more than one site for performance and reliability reasons. Applications and architecture of distributed computing has changed drastically during last decade and so has replication protocols. Different replication protocols may be suitable for different applications. In this manuscript we present a survey of replication algorithms for different distributed storage and content management systems ranging from distributed Database Management Systems, Service-oriented Data Grids, Peer-to-Peer

(P2P) Systems, and Storage Area Networks. We discuss the replication algorithms of more recent architectures, Data Grids and P2P systems, in details. We briefly discuss replication in storage area network and Internet.

**Optimal Replica Placement in Data Grid Environments with Locality Assurance:**

Pangfeng Liu Yi-Fang Lin, Jan-Jan Wu proposed that Data replication is typically used to improve access performance and data availability in Data Grid systems. To date, research on data replication in Grid systems has focused on infrastructures for replication and mechanisms for creating/deleting replicas. The important problem of choosing suitable locations to place replicas in Data Grids has not been well studied. In this paper, we address three issues concerning data replica placement in Data Grids. The first is how to ensure load balance among replicas. To achieve this, we propose a placement algorithm that finds the optimal locations for replicas so that their workload is balanced. The second issue is how to minimize the number of replicas. To solve this problem, we propose an algorithm that determines the minimum number of replicas required when the maximum workload capacity of each replica server is known. Finally, we address the issue of service quality by proposing a new model in which each request must be given a quality-of-service guarantee. We describe new algorithms that ensure *both* workload balance and quality of service simultaneously.

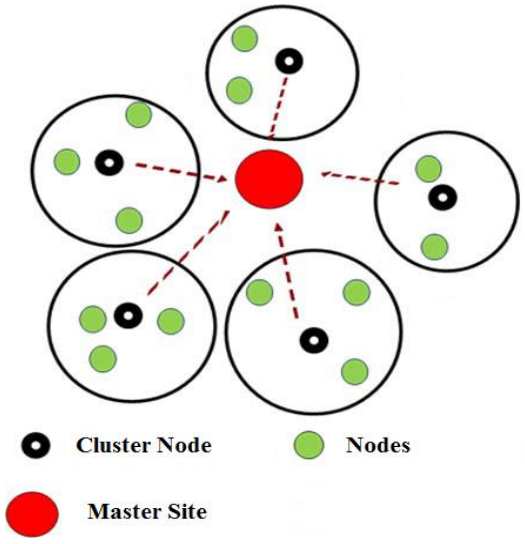
**PROPOSED SYSTEM:**

Every cluster comprises a number of nodes. Moreover, there is a master site that has all the files in the data grid. The storage of each cluster node is small therefore cannot accommodate all the files in the data grid. So files need to be brought from other nodes. The requested node checks if the closest node does not have the file, it searches the next closest node. Based on requested file popularity, master site replicate the file to cluster node otherwise clear files from the closest node based on Popular File Replicate Strategy (PFRF). we are adding Bloom Filter algorithm in order to capture the data in a short forms. This system will avoid the whole data storage in all the nodes same time every cluster node can maintain the Bloom filter index of the entire data stored. This process is very useful in order to fetch the data quickly. Packets are encrypted.

## 3. ALGORITHM / METHODOLOGY

PFRF strategy, Encryption

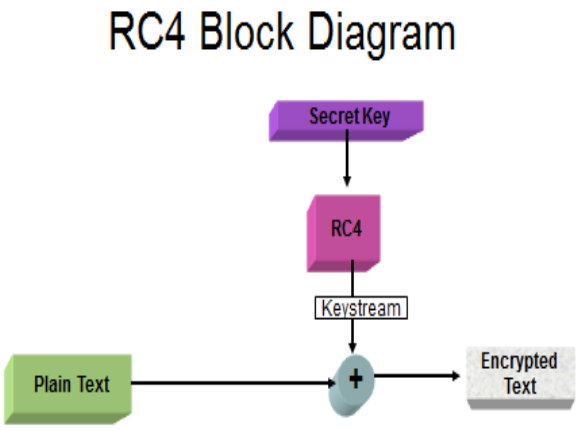
**ARCHITECTURE DIAGRAM**




The attackers are not able to guess and hack the passwords. It provide high security to data owner.

Encryption is yet another process by which information is protected from unauthorized access.It is normally accomplished by rendering the original information unreadable by using a reversible technique known only to the authorized entities. A symmetric key encryption algo. Invented by Ron Rivest. Normally uses 64 bit and 128 bit key sizes. Most popular implementation is in WEP for 802.11 wireless networks and in SSL. Cryptographically very strong yet very easy to implement. Consists of 2 parts: Key Scheduling Algorithm (KSA) & Pseudo-Random Generation Algorithm.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 5, Issue 2, pp: (12-17), Month: April - June 2017, Available at: www.researchpublish.com




RC4 like as a streaming cipher encrypts plaintext one byte at a time, but also can be designed to encrypt one bit a time or even units larger than a byte at a time. In this structure a key is input to a seudorandom bit generator that produces a stream of 8-bit numbers that are supposed to be truly random, the pseudorandom stream can't be predicted without knowledge of the input key. The output of the generator is called a key stream .It is combined one byte a time with the plain text stream using the bitwise exclusive-OR (XOR) operation. The RC4 is simple and easy to be explained. The algorithm is based on the use of random permutation. A variable length key K [ ] of from 1 to 256 bytes (8 to 2048 bits) is used to initialize a 256-byte state vector S [ ], with elements S [0] S [255]. At all times S [ ] contains a permutation of all 8-bit numbers from 0 to 255. For encryption and decryption, a byte K is generated from Journal of ELECTRICAL ENGINEERING 60, NO. 3, 2009 157 S [ ] by selecting one of the 255 entries in a systematic fashion. As each value of K is generated, the entries in S [] is again permuted. Figure 1 shows the block diagram of the RC4 two phases. To encrypt, XOR the value K with the next byte of plaintext, To decrypt, XOR the value K with the next byte of ciphertext, it is clearly here that if we use different keys for encryption and decryption we will never restore our plain text again, even if we use different keys for encryption the resulted ciphertext would not be the same. RC4 like any other stream cipher depends on the strength of its key stream, which in turns depends onthe degree of randomness of its pseudo random bit generator (throughout this paper the term pseudo random number generator refers to PRBG).The output of such generator needs to meet stronger requirements than for other applications. In particular, their output must be unpredictable in the absence of knowledge of the inputs.

**MODULE DESCRIPTION:**

The proposed system has the following modules,

1. **Network Construction**
2. **Bloom cast**
3. **Bloom filter**
4. **Distribution Of Bloom Filter Among The Nodes**
5. **Ranking Process**
6. **Retrieval Of Data**

**Network Construction:**

The overall connected network consists of number of nodes and it maintains the details of each node and it establishes the direct connection between every node and exchanges the data. And that network node called as server protects the details about the node IP address, port details and status. The other nodes request the service to server node for their updates.

**Bloom cast:**

In an peer-peer network every peer is centralized and each contains the details of every node and the network is totally unstructured and these bloom cast is an significant full text retrieval method. Peer-peer protocol will replicate the data which is searched by the user to all the peer nodes. Bloom cast structures the replicated data into lightweight DHT to support random node sampling and network size estimation.

**Bloom filter:**

Bloom filter is the technique of hashing where the user's need will be done by hashing it removes the stop words performs stemming then it creates the hash value for the keywords. The bloom filter gets the Query from the node, it performs multiple hashing in the query and as a result it converts the query into URLs.

**Distribution of Bloom Filter Among The Nodes:**

Once the searched information is converted into hash values, the hash values are updated to bloom filter. Then during the search by the user it searches the information using the hash values. After the match found it results the data.

**Ranking Process:**

Using the chord algorithm, the server peer node will forwards and backwards the searches and that will lead to the ranking of the documents based on the number of user and then the best document is filtered out to the user.

**Retrieval of Data:**

After ranking the documents, the user can retrieve their needs based on their views. Using bloom filter it's significant and efficient way to retrieve the data for an unstructured peer-peer network.

## 4. CONCLUSION

In this paper, a round-based data replication strategy called IPFRF has been implemented. IPFRF is based on PFRF but overcomes the shortcomings of PFRF. IPFRF is superior to PFRF in terms of average file delay per request, average file bandwidth consumption per request, and percentage of files found. IPFRF strategy achieved a reduction in average file delay per request up to 19.38 and 60.74 percent in scenarios 1 and 2, respectively, while it achieved a reduction in average file bandwidth consumption per request up to 18.00 and 55.84 percent in the same scenarios. Additionally, IPFRF strategy achieved an improvement in percentage of files found up to 46.69 and 217.81 percent in scenarios 1 and 2, respectively.

## REFERENCES

[1] S. Figueira and T. Trieu, Data Replication and the Storage Capacity of Data Grids. Berlin, Germany: Springer-Verlag, 2008, pp. 567–575.

[2] D. G. Cameron, A. P. Millar, C. Nicholson, R. Carvajal-Schiaffino, K. Stockinger, and F. Zini, "Analysis of scheduling and replica optimisation strategies for data grids using Optorsim," J. Grid Comput., vol. 2, no. 1, pp. 57–69, 2004.

[3] H. Lamehamedi, B. Szymanski, Z. Shentu, and E. Deelman, "Data replication strategies in grid environments," in Proc. 5th Int. Conf. Algorithms Architectures Parallel Process., 2002, pp. 378–383.

[4] M. Bsoul, "A framework for replication in data grid," in Proc. 8th IEEE Int. Conf. Netw. Sens. Control, Delft, The Netherlands, 2011, pp. 234–236.

[5] K. Ranganathan and I. Foster, "Identifying dynamic replication strategies for a high-performance data grid," in Proc. GRID '01: Proc. 2nd Int. Workshop Grid Comput.. London, United Kingdom: Springer-Verlag, 2001, pp. 75–86.

[6] M. Tang, B. Lee, C. Yeo, and X. Tang, "Dynamic replication algorithms for the multi-tier data grid," Future Generation Comput. Syst., vol. 21, no. 5, pp. 775–790, 2005.

[7] Q. Rasool, J. Li, G. Oreku, and E. Munir, "Fair-share replication in data grid," Inform. Technol. J., vol. 7, no. 5, pp. 776–782, 2008.

[8] R. Chang and H. Chang, "A dynamic data replication strategy using access-weights in data grids," J. Supercomput., vol. 45, no. 3, pp. 277–295, 2008.

[9] S. Park, J. Kim, Y. Ko, and W. Yoon, "Dynamic data grid replication strategy based on internet hierarchy," in Proc. 2nd Int. Workshop Grid Cooperative Comput., 2003, pp. 838–846.

[10] J. Wu, Y. Lin, and P. Liu, "Optimal replica placement in hierarchical data grids with locality assurance," J. Parallel Distrib. Comput., vol. 68, no. 12, pp. 1517–1538, 2008.

[11] M. Bsoul, A. Al-Khasawneh, E. E. Abdallah, and Y. Kilani,"Enhanced fast spread replication strategy for data grid," J. Netw.Comput. Appl., vol. 34, no. 2, pp. 575–580, 2011.

[12] M. Bsoul, A. Al-Khasawneh, Y. Kilani, and I. Obeidat, "A threshold-based dynamic data replication strategy," J. Supercomput.,vol. 60, no. 3, pp. 301–310, 2012.

[13] K. Sashi and A. Thanamani, "Dynamic replication in a data grid using a modified BHR region based algorithm," Future Generation Comput. Syst., vol. 27, no. 2, pp. 202–210, 2011.

[14] N. Mansouri and G. Dastghaibyfard, "A dynamic replica management strategy in data grid," J. Netw. Comput. Appl., vol. 35, no. 4,pp. 1297–1303, 2012.

[15] Z. Wang, T. Li, N. Xiong, and Y. Pan, "A novel dynamic network data replication scheme based on historical access record and proactive deletion," J. Supercomput., vol. 62, no. 1, pp. 227–250, 2012.