ISSN No. 0976-5697

Volume 6, No. 6, July-August 2015

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Enhanced Replication Strategy using Proportional Replication

Sujata I Mane
M. Tech (CNE) Dept.
PG Studies
Belgaum, India

Dr.S.L.Deshpande
Professor Computer Network Engineering Dept.
PG Studies
Belgaum, India

Abstract: Replication is used to improve the performance of data access in terms of increased availability, load balancing and fault tolerance. There are three aspects which need to be considered are target replica, frequency of replica and location of replica. As the number of replica increase, data consistency & maintenance overhead are major challenges in replication of data. To ensure data availability and consistency in order to deal with fast updates, optimal data replication can be used. To determine optimal number of replica during initial placement phase, uniform and proportional replica distribution strategy can be used. Proportional replication strategy replicates items proportional to their popularity but harder to find less popular items. To resolve this problem the optimised distribution strategy can be implemented which proactively replicate the data as per user request. By using analytical model, this paper shows increased availability, improved performance and reduced system overhead in the form of number of hits to access data.

Keywords: data replication; replica distribution strategy; optimal number of replica; proportional replication

I. INTRODUCTION

The challenging task for distributed system is to ensure data availability and consistency in order to deal with fast updates. To improve the performance of data access in distributed systems, optimal replication can be used. Replication is a strategy in which multiple copies of same data are stored at different locations. Replicating the data and moving it closer to where it is required. This helps to resolve the scalability problem as well as improves efficiency of data access. If replication is used without proper synchronization mechanisms, consistency of system shall be violated [1]. Replication of data in distributed system can improve performance of system in terms of:

1. Increased availability, reliability and fault tolerance

Data replication creates multiple copies of same data at different locations which can increases availability of data. Thus service disruption can be avoided due to system node failures.

2. Scalability

As server loads are decreased, the service capacity in terms of response time and QOS requirements can be improved.

3. Increased performance

Due to efficient data accessibility, increases performance of system especially when user requests only for read only queries.

A. Setting a Replication Plan

Prior to replication, there are three aspects which need to be considered as follows:

1. Target replica: This depends upon popularity of data and importance of content to meet user requirement on waiting time and to speed up the data access. Also indentifies type of data to be replicated.
   - Full objects, fine grained objects, chunks or blocks can be replicated.
   - Whether data could be static or dynamic.
   - Evaluation of homogeneity and heterogeneity degree of data.
   - Whether replication could be for read only queries or update transactions.
2. Frequency of replica: If the number of newly created replicas increases, the maintenance of system cost will be increases. Thus large number of replicas may not increases availability, but increases unnecessary system overhead.
   - This should address the necessity of: how much data is to be replicated to improve the system performance?
3. Location of replica: If the number of replicas and user requests are distributed correctly then the consumption of network bandwidth and execution rate of system will improve.
   - Identifies the heterogeneity of computing environment.
   - Analyses storage capacity, type of storage available in the network.

B. Replication Techniques in distributed system

Depending upon the user requirements different replication techniques are used in distributed system [1]. The criteria for the choosing of which data to be replicated and selection of locations or sites where the replica should be stored, these are major features of replication algorithm in distributed system. The replica should be placed at locations which are close to source nodes.

Replication techniques are classified based on [5]:

1. Replica creation strategy
2. Replica distribution
3. Erasure coded replication
4. Schemes for super peer architecture

This paper concentrates on replica distribution which is described as follows:

1. Replica distribution

The replica distribution strategy distributes replicas in terms of number of replicated data copies to be distributed in the network. Storage capacity and access bandwidth of peers affect the number of replicas. Also availability of system, popularity of data, number of sites and frequency of node failures affects the number of replicas [4] [2]. To fix the number of replicas during initial placement phase, we will use the static replica distribution, uniform and proportional, as given in [2].

Considering each data file is replicated on r_i nodes. The total number of files in a network is R.

$$R = \sum_{i=1}^{m} r_i \tag{1}$$

(where m is the number of individual files / objects)

i. Uniform Replication

The uniform replication strategy is where all items are equally replicated. Thus by using the above equation, replica distribution for uniform replication strategy can be represented by,

$$r_i = R/m \tag{2}$$

ii. Proportional Replication

The number of replicas is proportional to their popularity of data access frequency. Thus, if a data item is more popular then it has more chances of finding the data more close to the sites where query was submitted.

$$r_i \propto q_i \tag{3}$$

where, q_i = relative popularity of the file/ object (in terms of number of queries issued for i^{th} file).

$$\sum_{i=1}^{m} q_i = 1 \tag{4}$$

If all objects were equally popular, then

$$q_i = 1/m \tag{5}$$

But, results have shown that the object shows Zipf-like distribution in a system. Thus query distribution is as follows:

$$q_i = 1/m^{\alpha} \text{ (where } \alpha \text{ is close to unity).} \tag{6}$$

In Uniform replication strategy, the system creates fixed number of replica copies when the item first enters the system and in proportional replication strategy, the system creates fixed number of copies every time when the item is requested by user. Uniform replication and proportional replication strategies are not the same. Popular items are easier to find in proportional replication strategy and the data objects which have less popularity takes more time to find. Proportional replication requires large in number limit on maximum search size while uniform minimizes this limit. Uniform and Proportional strategy shows same average search size on successful query execution. To optimize the distribution strategy, random distribution of replicas in which proactively replicate the data has been used. Uniform and Proportional strategy shows uniform distribution and optimized strategy shows exponential distribution which will give better hit rate than the traditional one.

C. Challenges in replicated environment

Although replication strategy can be used to improve performance of system, some issues of replication as follows:

1. Data Consistency: Replication should be transparent to the user. Replication can be used to increase availability of data but maintaining consistency of data might be difficult. Consistency can be defined as same data is available at any time to all the users. When original data and all its replicas are identical then full consistency can be maintained. In partial consistency some differences are present between original data and its replicas. The degree of consistency dependence on whether the system or application can tolerate partial consistency. There are multiple solutions implemented to maintain consistency of data such as strict consistency, sequential consistency, causal consistency, eventual consistency, pipelined random access memory consistency and weak consistency [3]. The applications which have high precision may require strict data consistency of the updates made by transactions.

2. Downtime during new replica creation: When new replica is to be created, performance is severely affected because some application requires strict data consistency is to be maintained. Due to consistency requirements, requests are not fulfilled by sites.

3. Maintenance overhead: When the data replication is done at many sites then replica occupies more space which creates system overhead.

4. Lower write performance: Performance of write operations can be significantly lower in applications which require high updates in replicated environment, because the transaction may need to update multiple copies [1].

II. PROPOSED WORK AND METHOLOGY

As number of replicas increases, maintaining consistency and system overhead becomes more difficult. To resolve this problem, enhanced replication strategy can be used. Proportional replication and uniform replication shows same average search size on successful query. The proportional replication strategy creates number of replicated data copies which is proportional to number of requests. The proposed approach deals with replicating the data as per user request. When user sends a request for data to the server, search engine searches the data in replicated database. If data not found then request goes to main database and retrieves the data. This architecture is shown in figure 1.

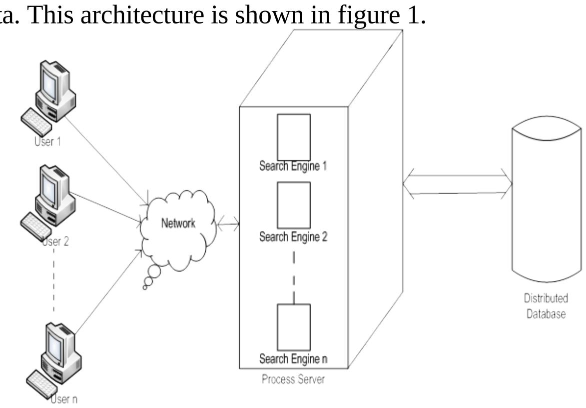

Figure 1. Block diagram of Enhanced Replication Strategy using Proportional Replication

By analyzing previous proportional replication results, proactively replicating the data will improve the system performance.

In enhanced replication strategy, owner replication is used which replicates data objects at requester node once the data is found. The proposed replication strategy will proactively replicate the data which is generated as per user requests for data. This strategy uses FIFO principle. In

FIFO, if replicated data queue length becomes full, this will remove first data item from the replicated data queue and replaced by a new data. When user request for data, first it is checked in replicated data. If data is found gives response to user, otherwise request goes to main center node and gives response to the user by replicating the data at intermediate node.

The enhanced replication strategy is represented by a main process which consists of data set. This main process is connected to three child processes with variable queue length where the replicated data is stored as per user request. Initially queue is empty. The steps followed by enhanced replication strategy as follows:

Step 1: When N number of randomly generated users sends a request for a data, initially request goes to main process and retrieves the data. This data is replicated at child process or intermediate node and gives response to user.

Step 2: For next request, first searches in replicated data at child process. If data is found, gives response to user otherwise request goes to main process and retrieve the data which is replicated in child process.

Step 3: This process is tested for given amount of time in seconds which replicates the data at second level by using FIFO principle.

Step 4: In enhanced replication, proactively replicate the data which is generated by above proportional replication process.

Step 5: Repeat the step2 to step3 for a given time duration.

The simulation model for enhanced replication strategy is shown in figure 2. The simulation model experimental results are shown in next section.

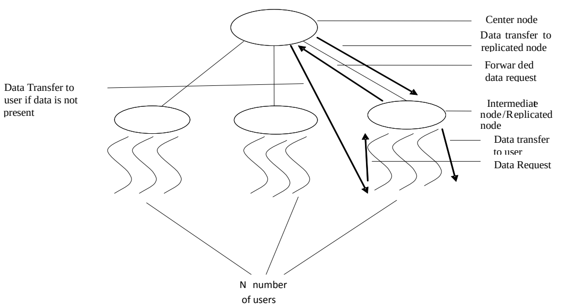

Figure 2. Simulation Model for Enhanced Replication strategy

III. EXPERIMENTAL RESULTS AND ANALYSIS

A. Proportional replication strategy

In proportional replication strategy, as per user request it replicates the data object at intermediate node. This is represented by a main process which consists of data set 1 to 100,1 to 1000 and 1 to 1000 numbers. Main process creates three child processes which are initially empty of queue length 50, 75, 500,750, and 5000. When number of users generated at third level, make a request for a data. Initially request goes to main process, copies the data from main process to child process and gives response to user by retrieving the data. For next request, first data is searched in child process (intermediate node) where the data is replicated. If data is found gives response to user, otherwise request goes to main process and copies the data and gives reply to user. This simulation is executed for given amount of time duration. This is shown in the following Table I, Table III, and Table V which calculates number of hits, number of misses occurred at second level (replicated data) and number of replacements by applying the FIFO principle.

B. Enhanced replication strategy

In enhanced replication strategy, we proactively replicate the data which is generated by proportional replication. By applying the same process as proportional replication, results are shown in the following Table II, Table IV, and Table VI for data set 100,1000,10000.

By observing the results shown in Table I and II, indicates the increased hits % by 5 and 4 for queue length 50 and 75 respectively which shows improved availability of data. The miss % decreases by 5 and 4 which shows reduced access latency. Also shows correlation between number of request and number of hits is 0.99, number of request and number of misses is 0.99 and number of request and number of replacements is 0.99.

By observing the results shown in Table III and IV, indicates the increased hits % by 9 and 14 for queue length 500 and 750 respectively which shows improved availability of data. As the data set increases, the data availability will be increased.

By observing the results shown in Table V and VI, indicates the increased hits % by 8 and 30 for queue length 5000 and 7500 respectively which shows improved availability of data. Also shows correlation between number of request and number of hits is 0.99, number of request and number of misses is 0.99 and number of request and number of replacements is 0.99.

Table I. Experimental Results for Proportional Replication for data set 100

Data set	Number of users generated	Total number of requests	Queue length	Number of request goes to main process	At Second level		Total number of replacements	hits %	Miss %	Correlation bet^n number of request and number of hits	Correlation bet^n number of request and number of misses	Correlation bet^n number of request and number of replacements
					Total number of hits	Total number of misses						
1-100	83	2131	50	1107	1024	1106	1057	45	55	0.99	0.99	0.99
1-100	83	3250	75	872	2378	871	797	70	30	0.99	0.99	0.99

Table II. Experimental Results for Enhanced Replication for data set 100

Data set	Number of users generated	Total number of requests	Queue length	Number of request goes to main process	At Second level		Total number of replacements	hits %	Miss %	Correlation bet^n number of request and number of hits	Correlation bet^n number of request and number of misses	Correlation bet^n number of request and number of replacements
					Total number of hits	Total number of misses						
1-100	46	3587	50	1805	1782	1805	1805	50	50	0.99	0.99	0.99
1-100	37	3885	75	993	2892	993	993	74	26	0.99	0.99	0.99

Table III. Experimental Results for Proportional Replication for data set 1000

Data set	Number of users generated	Total number of requests	Queue length	Number of request goes to main process	At Second level		Total number of replacements	hits %	Miss %	Correlation bet^n number of request and number of hits	Correlation bet^n number of request and number of misses	Correlation bet^n number of request and number of replacements
					Total number of hits	Total number of misses						
1-1000	683	2949	500	1625	1328	1624	1125	42	59	0.99	0.99	0.99
1-1000	683	3102	750	1193	1191	1192	443	61	39	0.99	0.99	0.99

Table IV. Experimental Results for Enhanced Replication for data set 1000

Data set	Number of users generated	Total number of requests	Queue length	Number of request goes to main process	At Second level		Total number of replacements	hits %	Miss %	Correlation bet^n number of request and number of hits	Correlation bet^n number of request and number of misses	Correlation bet^n number of request and number of replacements
					Total number of hits	Total number of misses						
1-1000	615	4615	500	2274	2340	2274	2274	51	49	0.99	0.99	0.99
1-1000	646	4339	750	1066	3273	1066	1066	75	25	0.99	0.99	0.99

Table V. Experimental Results for Proportional Replication for data set 10000

Data set	Number of users generated	Total number of requests	Queue length	Number of request goes to main process	At Second level		Total number of replacements	hits %	Miss %	Correlation bet^n number of request and number of hits	Correlation bet^n number of request and number of misses	Correlation bet^n number of request and number of replacements
					Total number of hits	Total number of misses						
1-10000	4350	18203	5000	10628	7575	10627	5628	42	58	0.99	0.99	0.99
1-10000	4350	17347	7500	7360	9987	7359	2134	37	63	0.99	0.99	0.99

Table VI. Experimental Results for Enhanced Replication for data set 10000

Data set	Number of users generated	Total number of requests	Queue length	Number of request goes to main process	At Second level		Total number of replacements	hits %	Miss %	Correlation bet^n number of request and number of hits	Correlation bet^n number of request and number of misses	Correlation bet^n number of request and number of replacements
					Total number of hits	Total number of misses						
1-10000	8330	20860	5000	10453	10407	10453	10453	50	50	0.99	0.99	0.99
1-10000	5856	19512	7500	5965	13547	5965	3782	67	33	0.99	0.88	0.93

IV. CONCLUSION

The enhanced replication strategy analyses the effect of replica distribution on the system performance in terms of data availability and consistency of data. We build a mechanism by means of analytical model to illustrate the working of enhanced replication strategy. This utilizes the proportional replication strategy based on replica distribution to analyze the strategy. We find that the number of hits increases by using enhanced replication strategy than proportional replication, if proactively replicate the data. This shows the increased availability of system.

V. REFERENCES

[1] Sushant Goel , Rajkumar Buyya Grid, "Data Replication Strategies in Wide Area Distributed Systems" ,Enterprise Service Computing: From Concepts to Deployment, 2006.

[2] Cohen E., Shenker S. "Replication Strategies in Unstructured Peer-to-Peer Networks", Proceeding of Special Interest Group on Data Communications (SIGCOMM), (pp: 177-190), Pittsburg, USA, August 2002.

[3] David Bermbach and Jorn Kuhlenkamp, “Consistency in Distributed Storage Systems”, Networked systems-Springer,2013.

[4] Modou Gueye, Idrissa Sarr, Samba Ndiaye, “Database Replication in Large Scale Systems: Optimizing the Number of Replicas”, Proc. Of EDBT/ICDT Workshops, pages 3-9, 2009.

[5] Sabu M. Thampi, Chandra Sekaran. K “Survey of Search And Replication Schemes In Unstructured P2P Networks”, Network Protocols and Algorithms, ISSN 1943-3581,Vol.2, 2010.