

Data Replication in Online Social Network on Datacenters

Supriya F. Rathod¹, Prof. A.V.Deorankar²

P.G. Student, Department of Computer Science and Engineering, Government College of Engineering, Amravati, India¹

Associate Professor, Department of Information Technology, Government College of Engineering, Amravati, India²

ABSTRACT: In recent years online social network model are having datacenters that are globally distributed which helps to reduce the service latency which causes higher inter-datacenter communication load. For example in facebook, the master datacenter updates all other datacenters, in which each datacenter has a full copy of all data this generates load in the model. Distributed datacenters can store data which is geographically closest to the data storage datacenters. However, numerous interactions between various users lead to numerous inter-datacenter communication and that's why long service latencies. In this paper, we aim to reduce inter-datacenter communications by introducing the concept of data replication in distributed datacenters to achieve low service latency. Thus we first make sure the benefits of the new model and present online social network properties that motivate the basis of our design. We then propose the concept of Selective Data replication mechanism in Distributed Datacenters (*SD3*). Now replicas need inter-datacenter data update and visit rates, therefore datacenters in *SD3* consider update and visit rates of user interaction to select user data for generating replica. *SD3* also atomizes different types of data (e.g., status update, friend post, music) from user interaction and updates for replication, which leads to decrease the inter-datacenter communication.

KEYWORDS: Online Social Networks, Datacenter, Scalability, Data replication, Locality

I. INTRODUCTION

In the past few years, Online Social Networks (OSNs) have completely spread over the world. As of the third quarter of 2016, facebook becomes one of the most popular and largest worldwide OSNs which has 1.79 billion monthly active users, 80% of whom are outside the US. However, Facebook datacenters are deployed sparsely outside the US. We will use inter-network communication load balancing concept. To replicate data in Online Social Network distributed datacenters and to minimize inter-datacenter communication load. An entire user data set is made up of several types of data, including wall posts, music, personal info, photos, videos, and comments. While photos and videos are stored in Facebook's content delivery network (CDN) partners, all other data is stored in Facebook's datacenters. We then propose Selective Data replication mechanism in Distributed Datacenters (*SD3*) for OSNs that embraces the aforementioned general features. *SD3* is proposed for the new OSN model that distributes smaller datacenters worldwide and maps users to their geographically closest datacenters.

Replication of data includes copy of documents, text in the form of XML document on local server which reduces database load and increase performance as time required to fetch data from XML is less than that of database. This increases security as well as availability of documents in case of any attack. Data replica will be stored on the various datacenters according to their geographically nearest server. We propose location based load balancing concept using which user will be redirected to the geographically nearest server automatically. In case of any damage, system will automatically recover the file from stored replica documents in distributed datacenters. Thus, in this paper, we study how to replicate data in online social network using distributed datacenters to minimize network load. The replication of user data can help to provide more data requests locally, leading to lower service latencies. A rarely visited replica provides a little advantage service latency reduction while at the same time it increases network load for updates, leading to higher network load. In online social networks it has been seen that interactions and friendships between local users are more in number than interactions and friendships between distant users [11]. A user's interaction

frequencies with his/her various friends vary widely [12]. Also, for different types of user the visit and update rates [10] of data are different (e.g., posting, status update).

Data Replication: Data replication is the frequent electronic copying of data from a database in one computer or server to a database in another so that all users share the same level of information. The result is a distributed database in which users can access data relevant to their tasks without interfering with the work of others. Data replication is one way of supporting data integration requirements. Data replication provides trusted data synchronization (including change data capture capabilities) and availability, enabling you to efficiently manage data growth while increasing revenue using up-to-the-minute information. Real-time operational and analytical data synchronization enriches big data systems and mobile applications, empowering the event-driven business.

II. RELATED WORK

The basic idea to design data replication datacenters is based on many previous studies on online social network properties. The work in [13] studied online social network architectures and evolution patterns. Online social networks are characterized by the existence of different communities based on user interaction, with a high rate of interaction within communities and low rate of interactions outside [14]. For very large OSNs, the network communities are become untight [33]. This leads to the idea behind the decision of data replication in datacenters to create replicas based on various user interaction and update rates rather than static friend community.

Viswanath *et al.* found that social links can grow stronger or weaker over time, which supports *SD3*'s strategy of periodically checking the necessity of replicas. Also shows that different atomized user data has different visit/update rates, which supports the atomized user data replication in the basic design of data replication in datacenters [6]. Thomson, D. J. Abadi uses the consistency maintenance of replicas over geographically distributed datacenters or within a datacenter. These works focus on providing replicas on the geographically nearest server [2]. Guoxin Liu, Haiying Shen uses the algorithm for selective user data replication. Also provides concept of selective data replication mechanism in distributed datacenters. These works focus on load balancing and document security and availability within the network [1]. Y. Zhou, T. Z. J. Fu, D. M. Chiu [2013] and M. S. Ardekani, D. B. Terry *SD3* also shares the adaptive replication techniques with some works in P2P systems and in clouds, which dynamically adjusted the number and location of data replicas. These works focus on load balancing, while *SD3* focuses on saving network load [3]. Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. C. Lau utilize the geo-distributed cloud to support large-scale social media streaming [5]. Unlike these works, *SD3* focuses on OSN's datacenters other types of user data and distributed small datacenters worldwide, which do not necessarily have full copy of all user data.

III. BASIS FOR DATA REPLICATION IN DATACENTERS

In this section, we verify the benefits of the new OSN model and analyze trace data from a major OSN to verify general OSN properties. *SD3* is particularly proposed for OSNs that embrace these general properties. In order to obtain a representative user sample, we used an unbiased sampling method [18] to crawl user data. If a randomly generated id exists in the OSN and the user with their id is publicly available, we crawled the user's data. We anonymized users' IDs and only recorded the time stamps of events without crawling event contents. All datasets are safeguarded and are not shared publicly. We crawled three OSN datasets for different purposes in our data analysis. The data replication is used to produce various copies of same data that is generated by using various user interactions in online social networks. It was observed that in social networks user interaction decreases with respect to their age and different users have different updates. Thus, user relationships do not necessarily have high data visit or update rates between the friends and the rates vary between different friends and overtime. Around 90% of all friend pairs have an average interaction rate below 0.4, and the average interaction rate of the remaining 10% ranges from 0.4 to 1.8. This result shows that the data visit rate between some users is low. Thus, replication based on inactive friend communities will generate replicas with low visit rates, which leads to waste of storage and inter data centered data updates. Therefore, we need to consider the visit rate of a user's data when determining the necessity of data replication. To improve the data replication data updates are calculated by using variance of interaction rates between each pair of friends by, $\sigma^2 = \sum (x - \mu)^2 / (n - 1)$, where x is the interaction rate, μ is the average of all interaction rates and n is the number of

interaction rates. Now we see that about 10% of friend interactions have high variance in the range of [0.444, 29.66], which shows that update rates differ greatly vary above time. This shows that the visit or update rates of data replica should be regularly checked and replicas with low visit rates and high update rates should be unused in order to save inter-datacenter communications for data updates and resources for storage. *SD3* atomizes a user's database on different types and avoids replicating once in a while visited and regularly updated atomized data in order to reduce inter-datacenter communications. Figure 1 shows an example of *SD3*, where users A, B, C and D are friends in network.

- *Global operation of datacenters:* A new datacenter is added to Japan (JP). Then, the master datacenter of users A and B is switched from CA to their closest datacenter, JP, and they will no longer undergo long service latency from CA.
- *Visit rate in data replication:* Though C and D are friends of JP's users, in this interaction user D's data is rarely visited while user C's data is visited frequently by JP's users, therefore JP only creates a replica of user C which is denoted by C'. As a result, users A and B can read and write their own data in JP and they are also friend of C which interacts with them frequently this saves inter-datacenter traffic. This leads to save network load for updating D's data. When user A interacts with D, JP needs to contact CA, but such visits are infrequent.
- *Considering visit and update rates in data replication model:* Though user A is visited by C and D, A's data is regularly updated that the update load is away from the loads saved by replication in both CA and VA; therefore CA and VA do not generate replicas of A. CA only contains replicas of C and B, while VA only generates replicas of B and D. When user A reads D, JP needs to interact with CA, but such visits are rare.
- *Updating:* When user B updates its status in master datacenter at JP, JP gives the update to CA and VA, as they both have B.

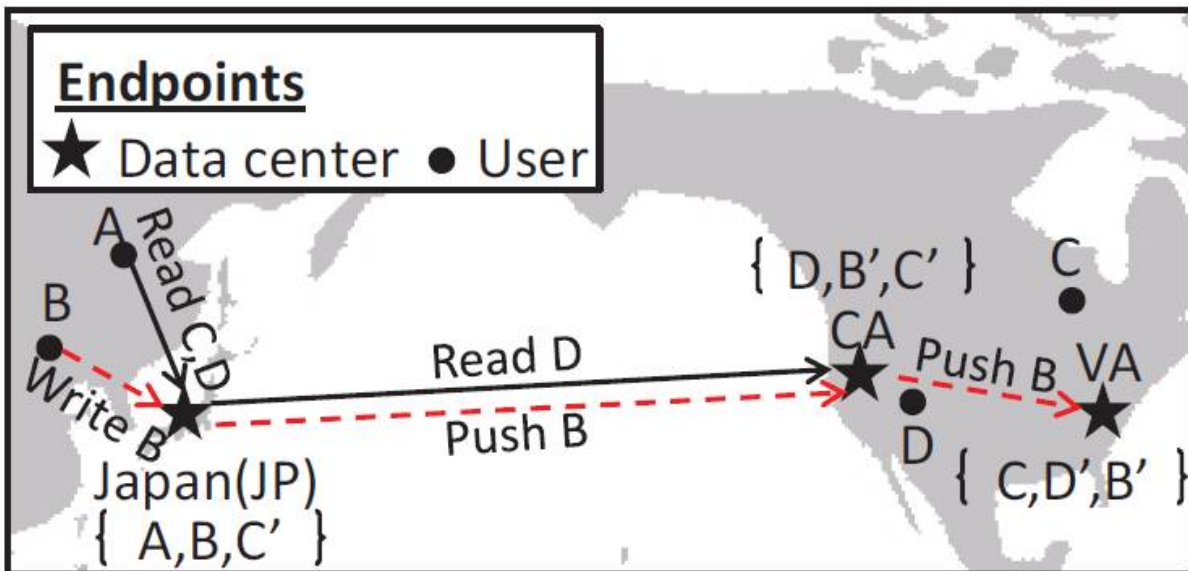


Fig. 17: Inter-datacenter interactions

Inter-datacenter communication occurs when a user reads or writes a friend's data in another nearest datacenter or when a master datacenter updates to slave datacenters. The inter-datacenter communications can be decreased by neighbouring replicas of these outside friends, but this replica also leads to data update load. This work breaks the tie between service latency and network load with some selected replica. We first measure the extra saved network load of all replicas by allowing for both saved visit network load and consumed update network load of all replicas. The network load depends upon message size, as big package requires more bandwidth. This network load also related to its transmission distance as longer distance may cause greater cross ISP network load, which is expensive. The network load of an inter-data center communication, say the k^{th} visit from any user in datacenter c towards a remote user j in datacenter $c(j)$, is measured by $S_{k,j}^v \times D_{c,c(j)}$ Mbkm, where $S_{k,j}^v$ denotes the size of the reply of k^{th} query on user j and

$D_{c,c(j)}$ denotes the distance between datacenters c and $c(j)$. We use $Uout(c)$ to denote the set of remote users to datacenter c and also visited by users in datacenter c , and use $RUout(c)$ to denote the set of remote users replicated in datacenter c . After calculating network load for each replica we measure the total network load of all replicas saved in inter-datatcenter communications within time period in the system (denoted by O_s). We time period denote by T . Then O_s equation becomes:

$$O_s = \sum_{c \in C} \sum_{j \in RUout(c)} \sum_{k \in (1 \dots V_{c,j} \times T)} S_{k,j}^v \times D_{c,c(j)} \dots (1)$$

Where,

C denotes the set of all datacenters of an OSN, $S_{k,j}^v$ denotes the average visit message size, and $V_{c,j}$ denotes the visit rate of datacenter c on remote user j , which is the number of the visits on user j during a unit time interval. In OSNs, users are mostly interested in friend's recent news such as status updates, posts, etc. in the News Feed. Thus, user data tends to be accessed heavily after creation, and then will be accessed rarely. As a result, $SD3$ only focuses on user j 's recent data to make the replica, which may have high $V_{c,j}$ to enlarge the network load savings. AO_s reaches to the maximum value if each datacenter c replicates user data for each visited remote user $j \in Uout(c)$. However, the extra update load is denoted by O_u which carried out by replica. Similar to O_s in Eq. (1), O_u is obtain calculating the summation of network load of each updated message, which is the product of the package size and the update transmission distance. Thus,

$$O_u = \sum_{c \in C} \sum_{j \in RUout(c)} U_j S_j^u \times D_{c,c(j)} \times T \dots (2)$$

Where, for user j 's recent data, U_j denotes its update rate, and S_j^u denote its average update message size. The main purpose of this paper is to minimize the inter-datatcenter network load by alying the benefits (denoted by B) of replicating data while maintaining a low service latency:

$$B_{total} = O_s - O_u \dots (3)$$

To achieve this goal, each datacenter tries to provide the advantage of its replicas by choosing a subset of remote visited users to replicate. Hence, it only replicates that remote users which rovides higher replica benefits than a pre-defined threshold, denoted by δ_{max} . Then each datacenter c keeps track of the each visited remote user j ($V_{c,j}$), and then j 's master datacenter provides j 's update rate, and calculate the benefit of replicating j 's data:

$$B_{c,j} = O_{c,j}^s - O_{c,j}^u = (V_{c,j} S_j^v - U_j S_j^u) \times D_{c,c(j)} \times T \dots (4)$$

Where $O_{c,j}^s$ and $O_{c,j}^u$ are the saved visit network load and update network load of replica j at datacenter c . Datacenter c replicates user j , only if $B_{c,j} > \delta_{max}$. As we know that the interaction rate between various friends varies. Thus, each datacenter regularly checks fo $B_{c,j}$ of each replica, and hence removes low $B_{c,j}$. After removing a replica, replica stops receiving updates. The replica which is removed is not deleted from storage, in order to make easy its creation later. Such replicas will be deleted if there is not enough storage space. To avoid repeated creation and deletion of the same replica, $SD3$ sets another threshold δ_{min} which is less than the δ_{max} . Datacenter c removes replica j , if $B_{c,j} < \delta_{min}$. Therefore,

$$RUout(c) \leftarrow \{j | j \in Uout(c) \wedge (B_{c,j} > \delta_{max} \wedge \neg j \in RUout(c)) \vee (B_{c,j} > \delta_{min} \wedge j \in RUout(c))\} \dots (5)$$

In Eq. (5), if we set δ_{max} and δ_{min} to negative infinity, then $SD3$ becomes the system to simply replicating all previously queried data with a long cache time. Different datacenter c' has different $D_{c,c'}$ for the same update message, as datacenter c sets δ_{max} (δ_{min}) for different remote datacenter c' with different values, which are then denoted by $\delta_{max,c'}$ ($\delta_{min,c'}$). As a result for a specific datacenter c' , there exists a tradeoffs between service latency and update load on datacenter. $SD3$ uses δ_{max} and δ_{min} to achieve an optimal tradeoffs; that is, in Eq. (5) which represent the weights for each respective goal. These goals can be obtained by using various factors such as saved network load, user service latency constraints, user data replication overhead, and so on. For example, if we set a negative value to δ_{max} , then OSN requires very short service latencies for browsing. Therefore, even when a replica promote $B_{c,j}$ has a negative value, which shows that this replica generates more update network load than its saved visit network load, it may still be created in order to gather the low service latency requirement. However, this replica brings more inter-datatcenter communications.

IV. EFFECT OF USER DATA REPLICATION

Firstly, we apply the selective data replication algorithm to see its effect. Figure 2 shows the median, 1st and 99th percentiles of the number of total replicas in all datacenters each day during the 90 days versus the number of users. The Y axis is in the log scale. This figure shows that the median results are SPAR > RS_L > SD3 > RS_S. Also, the median number of replicas of SD3 is about one third of SPAR's. SPAR replicates user data so that all data of a user's friends is in the same datacenter and the total number of replicas is minimized. As we have indicated that every friend relationship is not active at all the time, SPAR wastes system resources on those interactions with few relationships, thus producing the largest number of replicas. Each datacenter in RS replicates previously queried data from other datacenters. RS_L produces fewer replicas than SPAR because RS does not replicate not interacted friend data. SD3 considers the real interactions among datacenters, and only replicates user data that improves network load quality for visits than the generated update load, thus producing fewer replicas than RS_L. RS_S has only a one-day cache timeout, which makes its total number of replicas much smaller than SD3. SD3 constantly maintains replicas with high visit rates, resulting in better data availability than RS_S. This result shows that SD3 needs poorer load to create and continue replicas than the other systems.

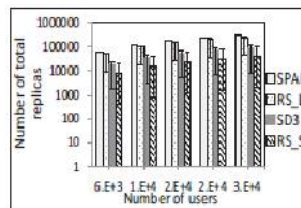


Fig.2 Number of replicas versus number of users

V. CONCLUSION

While a new OSN model with many small, globally distributed data centers will result in improved service latencies for users, a critical challenge in enabling such a model is reducing inter-datacenter communications (i.e., network load). Thus, we propose the Selective Data replication mechanism in Distributed Datacenters (SD3) to reduce interdata center communications while achieving low service latency. We verify the advantages of the new OSN model and present OSN properties from the analysis of our trace datasets to show the design rationale of SD3. Some friends may not have frequent interactions and some distant friends may have frequent interactions. In SD3, rather than relying on static friendship, each datacenter refers to the real user interactions and jointly considers the update load and save dvisit load in determining replication in order to reduce interdatacenter communications. Also, since different atomized data has different update rates, each datacenter only replicates atomized data that saves inter-datacenter communications, rather than replicating a user's entire dataset. SD3 also has a locality-aware multicast update tree for consistency maintenance

REFERENCES

- [1] Guoxin Liu, Haiying Shen, Senior Member IEEE, Harrison Chandler, "Selective Data Replication for Online Social Networks with Distributed Datacenters," IEEE Transactions on Parallel and Distributed Systems, 2015.
- [2] A. Thomson, D. J. Abadi, "CalvinFS: Consistent WAN Replication and Scalable Metadata Management for Distributed File Systems," in Proc. of FAST, 2015.
- [3] Y. Zhou, T. Z. J. Fu, D. M. Chiu, "On replication algorithm in P2P VoD," IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 21, NO. 1, FEBRUARY 2013.
- [4] M. S. Ardekani, D. B. Terry, "A Self-Configurable Geo-Replicated Cloud Storage System," in IEEE/ACM TRANSACTIONS ON NETWORKING, 2014.
- [5] Y. Wu, C. Wu, B. Li, L. Zhang, Z. Li, and F. C. Lau, "Scaling Social Media Applications Into Geo-Distributed Clouds," in Proc. of INFOCOM, 2012.
- [6] B. Viswanath, A. Mislove, M. Cha, K. P. Gummadi, "On the evolution of user interaction in facebook," in Proc. of WOSN, 2009.
- [7] "Facebook." Available: <http://www.facebook.com/>
- [8] "Socialbakers." <http://www.socialbakers.com/facebookstatistics/>
- [9] H. Shen and G. Liu. A geographically-aware poll-based distributed file consistency maintenance method for P2P systems. TPDS, 2012.

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization

Volume 6, Special Issue 1, January 2017

International Conference on Recent Trends in Engineering and Science (ICRTES 2017)

20th-21st January 2017

Organized by

Research Development Cell, Government College of Engineering, Jalagon (M. S), India

- [10] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user behavior in online social networks. In *Proc. of ACM IMC*, 2009.
- [11] M. Wittie, V. Pejovic, L. B. Deek, K. C. Almeroth, and B. Y. Zhao. "Exploiting locality of interest in online social networks," in *Proc. of ACM CoNEXT*, 2010.
- [12] Z. Li and H. Shen. "Social-p2p: An online social network based P2P file sharing system," in *Proc. of ICNP*, 2012.
- [13] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. "Measurement and analysis of online social networks," in *Proc. of IMC*, 2007.
- [14] A. Nazir, S. Raza, and C. Chuah. "Unveiling facebook: A measurement study of social network based applications," in *Proc. of IMC*, 2008.