Proxy-based Aggregated Synchronization Scheme in Mobile Cloud Computing

Giwon Lee, Haneul Ko, and Sangheon Pack
School of Electrical Engineering, Korea University, Seoul, Korea
Email: {goodkw, st_basket, shpack}@korea.ac.kr

Abstract—**Recently, data sharing applications among mobile users in cloud computing platforms (e.g., Dropbox and Google Drive) are very popular. In such applications, a cloud server is responsible for synchronization among mobile users and thus the synchronization traffic can be significant when the data are frequently updated and shared by many mobile users. To address this problem, we propose a proxy-based aggregated synchronization (PAS) scheme in which the proxy server and the cloud server conduct localized and aggregated synchronizations, respectively. Preliminary simulation results demonstrate that PAS can reduce the synchronization traffic by 83.1%.**

I. Introduction

Data sharing applications among mobile users in cloud computing platforms are very popular [1]. For example, mobile users of Dropbox and Google Drive can access their data in mobile environments and freely share the data with their friends and colleagues. In such applications, a user can add or modify a file in the local folder and this update is then automatically synchronized with a cloud server (CS). When the file is shared with other users, the CS is responsible for synchronizing the shared file with all sharing users. If the file is frequently updated, the synchronization traffic will be significant [2] and thus how to reduce the synchronization traffic is one of the most important issues in data sharing applications.

An update-triggered delta synchronization (UDS) scheme based on delta encoding [3] is a well-known solution for synchronization in data sharing applications. Delta encoding takes old and new versions of a shared file as inputs and computes the difference between them (i.e., delta). When the shared file is updated, only the computed difference is notified to the CS and users for synchronization. Since the size of the delta is much smaller than that of the original file, UDS can significantly reduce the synchronization traffic. Nevertheless, if the shared file is frequently updated and the number of users sharing the file is large, the synchronization traffic of UDS cannot be negligible.

In this work, we propose a proxy-based aggregated synchronization (PAS) scheme to reduce the synchronization overhead when updates frequently occur and the data is shared by many users. In PAS, mobile users are grouped into several domains depending on their geographical locations and each domain is managed by one proxy server (PS). Since the PS can be co-located with an adjacent router or base station, the PS can synchronize mobile users within its domain with low overhead compared with the CS. Note that the domain can be easily extended to include the access pattern or the authority of mobile users. In PAS, when an update occurs in a PS domain, the update is notified to the PS and the CS. After that, the PS synchronizes the update with mobile users in its domain. On the other hand, the CS accumulates updates during the pre-defined timer T and delivers the aggregated one to other PSs (except the originating PS) for synchronization of mobile users in other PS domains[1]. Preliminary simulation results show that PAS can reduce the synchronization overhead by 83.1%.

II. Proxy-based Aggregated Synchronization Scheme

In this work, mobile users are connected to one PS and PSs are connected to the CS. We assume that mobile users share a file F. To avoid simultaneous updates to F, only one node is granted to update F at the same time. For efficient synchronization, the use of delta encoding is assumed. Let d_j represent the jth updated delta of F. Since the PS maintains the list of mobile users in its domain, the PS can multicast the updated deltas to mobile users. To illustrate the operation of PAS, the following messages are defined.

- **Update(**F, d_j**)**: The mobile user sends this message to the PS and the CS to inform an update d_j of F.
- **Sync(**$F, d_j, d_{j+1}, \dots$**)**: The PS and the CS send this message to synchronize updates (i.e., $d_j, d_{j+1}, \dots$) with mobile users. Note that the CS can aggregate the multiple deltas to minimize the synchronization overhead.
- **Access(**F**)**: The mobile user sends this message to access the shared file F.
- **Notify(**F**)**: This message is sent by the CS to notify an update of F to mobile users in PS domains (except the originating domain). By receiving the message, mobile users can know there is an update to file F and thus they should contact the CS for future accesses to F.

If a mobile user updates a shared file, the update is immediately synchronized to other mobile users within the same domain by means of the PS. After that, the PS sends the updated delta to the CS for synchronization with other domains. To reduce the synchronization traffic, the updated delta is not synchronized immediately to mobile users in other domains. Instead, the CS first sends **Notify(**F**)** messages without the updated delta to PSs. The CS aggregates the updated deltas during the timer T and delivers them in a batch manner. As a result, it is possible to reduce the synchronization overhead (due to frequent updates) while providing mobile users with

[1]This proxy architecture is analogous to the cloudlet architecture [4] where the cloudlet locates between mobile users and the CS and takes several tasks from mobile users to reduce the end-to-end response time and the load of the CS. Unlike the existing cloudlet architecture, the proposed proxy architecture focuses on synchronization among mobile users instead of task offloading.

978-1-4673-1017-8/14/$31.00 ©2014 IEEE

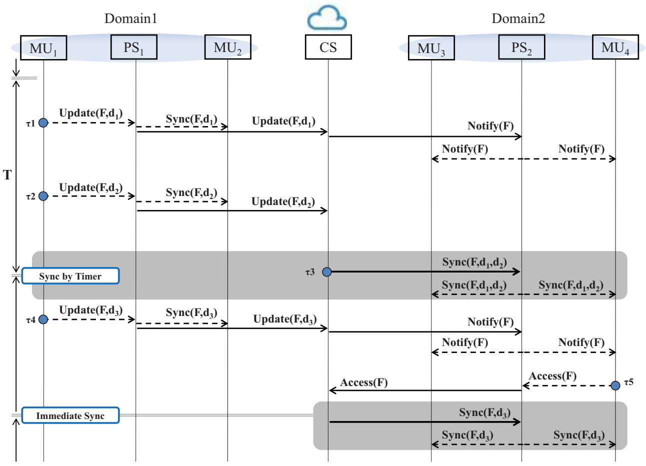

Fig. 1. PAS operation.

consistent access to F. Specifically, if a **Notify(**F**)** message has been received, the mobile user should always contact the CS to access F by sending an **Access(**F**)** message. Otherwise, the mobile user can use the local copy of F that has been informed by **Sync** messages. Of course, the aggregated synchronization may lead to additional latency compared to the immediate synchronization. Therefore, an appropriate timer should be selected to guarantee low latency as well as reduced synchronization traffic, which is one of our future works.

Figure 1 shows the operation of PAS. MU_1 represents a mobile user to be granted to update F. MU_1 sends the **Update(**d_1**)** message to PS_1 at τ_1. If there are mobile users sharing F in the domain of PS_1, PS_1 multicasts **Sync(**F, d_1**)** messages to them (e.g., MU_2). After that, PS_1 sends the **Update(**F, d_1**)** message to the CS. The CS does not send **Sync(**F, d_1**)** messages to other PSs immediately to reduce the synchronization overhead. Instead, the CS aggregates the updated deltas during the timer T while **Notify(**F**)** messages are immediately sent to mobile users in different domains to inform the update of F. As shown in Figure 1, the **Update(**F, d_2**)** message at τ_2 is not notified to mobile users in other domains because the update of F has been already notified owing to the update d_1. After the timer expiration at τ_3, the CS sends the updated deltas to all PSs except the originating PS (i.e., PS_1) in a batch manner. Note that the CS sends the updated deltas to PSs not to individual mobile users because the PS can multicast the deltas to mobile users. For example, the CS sends one **Sync(**F, d_1, d_2**)** message to PS_2 and PS_2 multicasts **Sync(**F, d_1, d_2**)** messages to MU_3 and MU_4. Thus, the gain of PAS increases when more mobile users in one domain share the file. At τ_4, there is another update d_3 of F. Then, the same procedure as d_1 is conducted since this is the first update after the timer expiration. When MU_4 wants to access F at τ_5, it sends the **Access(**F**)** message to the CS because the **Notify(**F**)** message has been received by d_3. After that, the CS sends the **Sync(**F, d_3**)** message to MU_4 immediately, and the timer is initialized.

III. Preliminary Simulation Results

For performance evaluation, we conducted event-driven simulations. Let D and M be the number of PS domains and the number of mobile users in a PS domain, respectively. For tactical analysis, we assume that the size of the delta is constant. The inter-access time of F follows an exponential random variable with the mean $1/\lambda$. The inter-update time of F is also drawn from an exponential distribution with mean $1/\mu$. Let N_s and N_r be the average number of **Sync** messages from the CS within the inter-access time and the average number of the deltas delivered by a **Sync** message, respectively. In our simulations, we compare the performance of PAS and UDS. Note that the CS synchronizes every updated delta to all mobile users in UDS since there is no concept of the PS.

TABLE I
Effect of μ

μ	UDS	PAS (T = $1/\mu$)		PAS (T = $2/\mu$)	
	N_s	N_s	N_r	N_s	N_r
5	25	8.82	0.15	4.22	0.31
10	50	18.74	0.16	9.16	0.32
15	75	29.10	0.17	14.11	0.32

Table I shows N_s and N_r for different values of μ when D = 3, M = 2. λ is normalized by 1. As μ increases, N_s increases because more updates are generated. It can be shown that PAS can reduce N_s (or synchronization overhead) significantly compared to UDS. This is because the synchronization within the originating domain is localized by the PS and the CS aggregates the updates during the timer. In addition, it can be found that N_s decreases with the increase of T, and thus larger T should be used to minimize the synchronization overhead. However, if T is too large, more updates are aggregated by the CS (i.e., larger N_r is obtained as shown in Table I) and longer latency is expected to access the up-to-date F. Consequently, an appropriate T balancing the synchronization overhead and the access latency should be determined to optimize the performance of PAS.

IV. Future Works and Conclusion

In this work, we have introduced a proxy-based aggregated synchronization (PAS) scheme for data sharing applications in cloud computing environments. Preliminary simulation results demonstrate that PAS can significantly reduce the synchronization overhead. In our future work: 1) we will carry out extensive simulations; 2) we will investigate the optimization of timer T by developing the analytical model of PAS.

Acknowledgment

This work was supported by National Research Foundation of Korea Grant funded by the Korean Government (NRF-2012R1A1B4000894 and NRF-2012K2A1A2032693).

References

[1] S. Azodolmolky, *et al.*, “Cloud computing networking: Challenges and opportunities for innovations,” *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 54-62, 2013.

[2] Z. Li, *et al.*, “Coarse-grained cloud synchronization mechanism design may lead to severe traffic overuse,” *Tsinghua Sci. Technol.*, vol. 18, no. 3, pp. 286-297, 2013.

[3] N. Samteladze, *et al.*, “DELTA: Delta encoding for less traffic for apps,” in *Proc. IEEE LCN*, 2012.

[4] M. Satyanarayanan, *et al.*, “The case for vm-based cloudlets in mobile computing,” *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14-23, 2009.