2015 IEEE 35th International Conference on Distributed Computing Systems

Operational Transformation for Real-time Synchronization of Cloud Storage

Agustina and Chengzheng Sun
Nanyang Technological University, Singapore
{Agustina, CZSun}@ntu.edu.sg

***Abstract* — This paper presents an Operational Transformation (OT) technique, named *CSOT* (*Cloud Storage OT*), that supports real-time file synchronization in cloud storage and achieve well-defined consistent combined-effects of concurrent file manipulation operations. We have developed and used a comprehensive suite of concurrency testing cases to derive and compare the results produced by CSOT and three industrial cloud storage systems and made interesting discoveries.**

Keywords—cloud storage; real-time synchronization.

I. INTRODUCTION

Cloud storage systems allow users not only to store and share files on the provider's data centres, but also to replicate them on mutiple devices. With such systems, users can access and modify replicated files in the same way as accessing native file systems on their personal devices; modifications to those files will be automatically synchronized to the provider's storage and among replicas on the users' devices. In such systems, users may freely and concurrently create new files and manipulate existing files via the replicas on their computers. It is well-known that concurrent manipulation of replicated data may conflict and result in inconsistent effects [1]. Real-time synchronization and consistency maintenance of replicated files is a challenge in cloud storage systems.

Existing cloud storage systems have been using proprietary synchronization techniques to achieve a variety of results in the face of concurrency, but none of them has ever explicitly specified what consistency requirements and combined-effects of concurrent operations are guaranteed to end-users. There were some prior researches on consistency maintenance among *back-end storage replicas* maintained by service providers. Consistency maintenance of back-end storage replicas is important for fault tolerance, availability, recovery, and fast access of cloud storage systems, but it is different from and complementary to consistency maintenance of *front-end storage replicas* on end-users' computers, which is what we focus on in this paper. A major difference is: back-end storage replicas and consistency solutions are transparent to end-users, wheras front-end storage replicas are visible to end-users and hence the consistency solutions must meet certain user requirements, such as well-defined, meaningful and desirable combined-effects for concurrent operations [1].

Operational Transformation (*OT*) is an established consistency maintenance technique for distributed real-time collaborative systems [2]. The basic idea of OT is to transform an operation defined on one state into another operation defined on a new state, so that the transformed operation can be correctly executed on the new state and achieve consistency among multiple replicas in the face of concurrent operations [2]. OT is particularly suitable for collaborative applications running over long-latency communication networks like the Internet because of its lock-free, unblocking, and unconstrained collaboration properties, and has been increasingly adopted in industrial applications, including Google Docs[1] and CodoxWord[2] [3]. Nevertheless, OT was never applied beyond within individual documents. In this work, we proposed a new OT technique, named *CSOT* (*Cloud Storage OT*), which extends OT from consistency maintenance within individual documents to a collection of documents in a shared workspace in cloud storage.

II. KEY COMPONENTS IN CSOT

One key insight that inspires the CSOT work is that collaborative manipulation of a collection of replicated files in cloud storage can be modeled as collaborative editing of a shared workspace consisting of those replicated files; user-generated file manipulation operations can be modeled as editing operations over this shared workspace; and file synchronization in cloud storage can be mapped into consistency maintenance of the shared workspace. Despite the similarities between real-time file synchronization in a workspace and real-time collaborative editing of an individual document, there exist important differences between them. One difference is in data and operation granularity: operations in cloud storage are targeting at *coarse-grained* objects like files or folders in a shared workspace, whereas operations in ordinary editors are targeting *fine-grained* objects, such as characters, words, or graphics in an ordinary text or graphics document [1][3]. Furthermore, data models (i.e. general structures of shared workspaces or documents) and operation models (i.e. collection of primitive operations that users may generate to manipulate the workspaces or documents) are all different from each other. These differences have major impacts on consistency requirements, as well as concurrency

[1] https://docs.google.com
[2] http://www.codoxware.com

1063-6927/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDCS.2015.90

control and conflict resolution strategies. Hence, a different OT solution is needed for cloud storage.

The CSOT solution consists of multiple aspects and components. First, we defined the file data model of the replicated workspace and the file operation model in cloud storage. A replicated workspace in cloud storage is modelled as a hierarchical tree of nodes that represent files and folders. The operation model consists of four primitive operations that users may use to manipulate the workspace: (1) *Create* (*CR*) to create a subtree; (2) *Delete* (*DL*) to delete a subtree; (3) *Rename* (*RN*) to rename a file or folder; and (4) *Update* (*UP*) to update the content of a file. Real-world cloud storage systems may provide other operations for users to manipulate files, but the effects of those operations can be achieved by combinations of the primitive operations. For example, a *move* can be achieved by a *Delete* followed by a *Create*.

Furthermore, we defined conflict and compatible relations among operations. A pair of concurrent operations conflict if they produce inconsistent workspace states when executed in different orders. We conducted comprehensive analysis on the effects achieved by executing every pair of operations (with all possible target pathname relations) in different orders; and found conflicts may occur among six pairs of operations under specific target pathname relations: two *CRs* conflict if they create subtrees with the same pathname; a *CR* conflicts with an *RN* if the *CR* creates a subtree that is either underneath the folder to be renamed by the *RN* or has the same pathname as the renamed node; two *RNs* conflict if they rename the same node, or they rename different nodes under the same folder to the same name, or one *RN* renames a node that is on the upstream of a node to be renamed by the second *RN*; an *RN* conflicts with a *DL* or an *UP* if the *RN* renames the node on the target pathname of the *DL* or *UP*; and two *UPs* conflict if they update the same file.

From general consistency requirements for collaborative editing systems [2], we derived four consistency requirements for replicated workspaces in cloud storage: (1) *Causality preservation*: operations with causal relationships must be executed in their causal orders. (2) *Convergence*: after executing the same group of operations at all sites, the replicated workspace states must be identical. (3) *Intention-confined effect*: changes made to the workspace must be confined to those produced by user-generated operations. (4) *Aggressive effect preservation*: effects of compatible operations are preserved fully; effects of conflicting operations are preserved as much as possible. Guided by these general requirements and conflict relations among operations, we defined desirable *combined-effects* – integrated effects of concurrent operations, independent of their execution orders. Explicit description of combined-effects for concurrent operations is important for end-users to understand the behaviour of a collaborative application, and for system-builders to devise suitable techniques for supporting the required effects in the face of concurrency.

Finally, we devised the CSOT system consisting of a set of new transformation functions for cloud storage and a generic OT control algorithm COT or POT [4]. We formally verified the correctness of CSOT in terms of the general consistency requirements and special combined-effects, and built a proof-of-concept implementation of the CSOT system.

III. CONCURRENCY BENCHMARK STUDY

We have also developed a suite of general benchmark testing cases for deriving what combined-effects can be achieved by a cloud storage system in concurrent operation executions, independent of whatever internal file synchronization techniques used. The benchmark suite is comprehensive and cover all possible combinations of file manipulation operation types, operation target pathname relations, and operation target types.

We have applied the same benchmark testing cases to the CSOT prototype and three industrial cloud storage systems: Dropbox, Microsoft OneDrive, and Google Drive (GDrive), to derive and evaluate the combined-effects they produce under the same benchmark. From benchmark testing, we have experimentally confirmed that the CSOT prototype can achieve the combined-effects as defined in this work and preserve the *convergence* property under all cases. From the results produced by Dropbox, OneDrive, and GDrive, we found none of these systems is able to achieve desirable combined-effects and consistent results under all benchmark cases, but they collectively cover all consistent combined-effects proposed in this work, which can be regarded as a real-world validation of the proposed combined-effects from this research. We also discovered that these three systems often produce different results under the same benchmark cases, some of which are undesirable or confusing to the users, and even inconsistent among multiple replicas. Detailed discussion of the concurrency benchmark design and discoveries will be included in an extended publication.

IV. CONCLUSIONS AND FUTURE WORK

In this work, we have contributed a novel CSOT solution to real-time synchronization of replicated files in cloud storage. This work is the first to extend OT from consistency maintenance in real-time collaborative editing within individual documents/files to across a collection of documents/files in a shared workspace in cloud storage. We have also contributed a suite of general concurrency benchmark testing cases and applied them to three industrial cloud storage systems to derive and evaluate the combined-effects of concurrent operations under these systems. This work is still ongoing and more comprehensive discussion of CSOT and concurrency benchmark study will be reported in future publications.

REFERENCES

[1] Agustina and C. Sun, “Dependency-Conflict Detection in Real-time Collaborative 3D Design Systems,” Proc. ACM CSCW 2013*,* pp. 715–728, doi:10.1145/2441776.2441856.

[2] C. Sun and C. Ellis, “Operational Transformation in Real-time Group Editors: Issues, Algorithms, and Achievements,” Proc. ACM CSCW 1998*,* pp. 59–68, doi: 10.1145/289444.289469.

[3] C. Sun, S. Xia, D. Sun, D. Chen, H. Shen, and W. Cai, “Transparent Adaptation of Single-user Applications for Multi-user Real-time Collaboration,” ACM TOCHI, vol. 13, no. 4, 2006, pp. 531–582, doi: 10.1145/1188816.1188821.

[4] Y. Xu and C. Sun, “Conditions and Patterns for Achieving Convergence in OT-based Co-editors,”, IEEE TPDS, no. 99, 2015, doi: 10.1109/TPDS.2015.2412938,