Moderated group authoring among weakly connected workgroups

Surendar Chandra (surendar@acm.org) and Nathan Regola (nregola@nd.edu)
University of Notre Dame, Notre Dame, IN 46556, USA

Our goal is to develop a practical groupware system that allows any group member to modify a shared file. Each user can modify any part of the document; the documents cannot be easily decomposed into sections that can be independently modified. Updates from different users will conflict and the goal is to resolve these conflicts and eventually produce a consistent document. Consider a group of users: Alice, Bob and Tom who are modifying a single document (illustrated in Figure 1). Each user creates updates on a shared document: Alice creates updates a_1, a_2, and a_3, Bob creates b_1 and b_2 and Tom creates updates t_1, t_2 and t_3. Traditionally, groupware systems ordered these updates using their causality relationships [1] in order to achieve consistency. The updates are then applied in this particular order. However, the system performance and the duration for all the updates to be applied to produce a consistent version depends on the availability patterns of the group members.

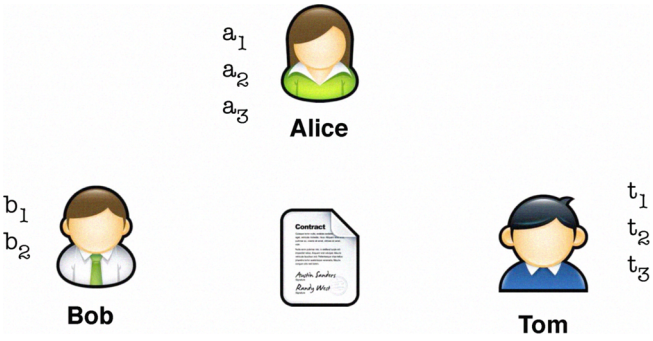

Fig. 1. Shared modification among Alice, Bob and Tom

Contemporary users are wireless. Wireless networks are ubiquitous, allowing users to operate from a variety of locations. Our empirical analysis of the availability patterns of wireless LAN users in a variety of locations showed that the user sessions are becoming smaller. The users also exhibited significant node churn. Further analysis of prior group collaboration systems [2] using wireless user availability traces from a variety of locales showed the practical limitations of prior collaboration systems. Systems such as Coda [3] and Ficus [4] had assumed fewer conflicting updates and better user availability than was observed among wireless users. We observed that a mandatory locking scheme will reject many updates because of simultaneous requests. Similarly, AFS [5] like *last writer wins* semantics causes a large number of inconsistent updates with an observable loss of update causality. Also, asynchronous propagation systems such as Bayou [6] experiences a large number of update conflicts and transaction roll-backs. More importantly, the worst performance in all these systems was observed during the peak availability durations; these systems fail when all the group members were available. We highlighted the inadequacy of prior systems that attempt to maintain a single copy of the shared document.

Instead, we develop a moderated collaboration mechanism that maintains multiple copies of the shared object. We maintain one updateable copy of the shared content on each group member's node. We also hoard read-only copies of each of these updateable copies in any interested group member's node. Our system introduces additional storage and conflict resolution overhead. We show this behavior for our group in Figure 2. In our illustration, each group member can potentially maintain three copies of the shared document. Depending on the user availability patterns, the hoarded copies on each user might not be current. For a group of size n, each user can potentially maintain one updateable copy (the author) and $(n-1)$ read-only copies (from other users in the collaboration group). Given the vast improvements in laptop storage (a 500GB 2.5" hard disk retails for less than USD$120), extra copies are a reasonable overhead. Group members can reduce the replica maintenance overhead by explic-

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.7011
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.7011

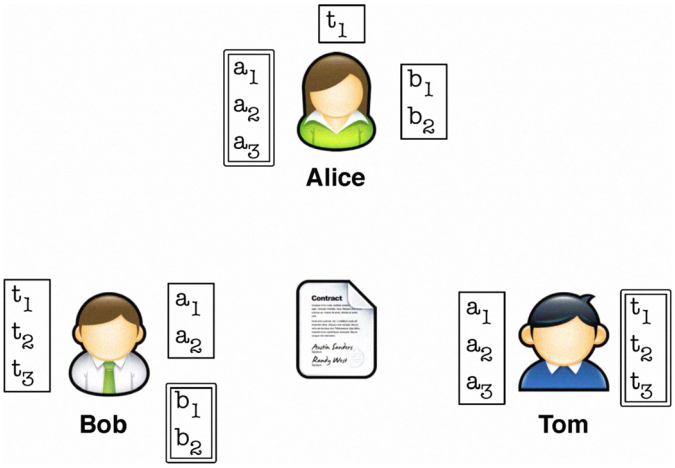

Fig. 2. Moderated collaboration among Alice, Bob and Tom

itly specifying the group members whose contents are replicated. Also, since the document versions are similar, deduplication can achieve good storage savings.

To resolve the update conflicts, each author manually moderates and incorporates modifications from other group members using these read-only replicas. For groups of size two, moderation operations are similar to reconciliation operations in Coda [7] and Ficus [8]. For larger groups, the moderator simultaneously incorporates updates from all the other group members into their copy. The groupware system can use the hoarded copies and help the user in identifying parts of the document that were modified by other group collaborators. Moderation operation is likely not scalable for large group sizes. The challenge is to evaluate the ease of moderation for typical shared documents.

The various versions of the shared document will eventually converge through independent moderation operations. The system automatically logs the provenance of all causal reads of contents from other replicas into the author versions. We assume that if an user opened a file from their colleagues, then they have incorporated the relevant updates into their own version. These provenance records allows any users to independently decide whether their updates had been incorporated into the final version of a particular file.

Using the wireless access traces, we show that our system avoids many of the problems of prior systems. Except for small groups, our distributed approach achieves similar performance as a server based system. We have built a prototype of our system using Git[1], an open source distributed version control system and the Fuse[2] userspace file system. Git is designed to be fast and efficient (space and time) and we inherit these advantages. We use our middleware, Yenta, to propagate updates across the group participants. Our analysis showed that pull mechanisms naturally randomized the times when updates are propagated and thus achieved better performance than push based mechanisms. The fundamental impediment to the system performance is the wireless user availability behavior and node churn characteristics. Update propagation is improved when members with good availability *shares* all copies regardless of whether they themselves were interested in the shared documents. Benchmarks show that our system achieves performance similar to a baseline fuse file system. This system is in regular use within our group. We are in the process of releasing it to a larger audience.

ACKNOWLEDGMENT

This work was supported in part by the U.S. National Science Foundation (CNS-0447671).

REFERENCES

[1] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Commun. ACM*, vol. 21, no. 7, pp. 558–565, 1978.

[2] S. Chandra and N. Regola, "*flockfs*, a moderated group authoring system for wireless workgroups," in *Mobiquitous 09*, Toronto, CA, Jul. 2009.

[3] M. Satyanarayanan, J. J. Kistler, P. Kumar, M. E. Okasaki, E. H. Siegel, and D. C. Steere, "Coda: A highly available file system for a distributed workstation environment," *IEEE Transactions on Computers*, vol. 39, no. 4, Apr. 1990.

[4] T. W. Page, R. G. Guy, J. S. Heidemann, D. Ratner, P. Reiher, A. Goel, G. H. Kuenning, and G. J. Popek, "Perspectives on optimistically replicated peer-to-peer filing," *Software—Practice and Experience*, vol. 28, no. 2, pp. 155–180, February 1998.

[5] J. H. Morris, M. Satyanarayanan, M. H. Conner, J. H. Howard, D. S. Rosenthal, and F. D. Smith, "Andrew: a distributed personal computing environment," *Commun. ACM*, vol. 29, no. 3, pp. 184–201, 1986.

[6] A. Demers, K. Petersen, M. J. Spreitzer, D. Terry, M. Theimer, and B. Welch, "The bayou architecture: support for data sharing among mobile users," in *Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, Dec. 1994, pp. 2–7.

[7] P. Kumar and M. Satyanarayanan, "Flexible and safe resolution of file conflicts," in *USENIX Technical Conference*, New Orleans, Louisiana, Jan. 1995, pp. 8–8.

[8] P. Reiher, J. S. Heidemann, D. Ratner, G. Skinner, and G. J. Popek, "Resolving file conflicts in the Ficus file system," in *USENIX Technical Conference*, Boston, MA, Jun. 1994, pp. 183–195.

[1] http://git.or.cz/

[2] http://fuse.sourceforge.net/

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.7011
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.7011