Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 4, April 2013, pg.193 – 196

RESEARCH ARTICLE

Sharing File System for Campus Wide-Workgroups

Hangad Abdulkadar J[1], Avantkar Amol D[2], Chougule Amit B[3]

[1] abbu2015@gmail.com; [2] avantkaramol@yahoo.in; [3] amit.bvceok@gmail.com

Abstract— This paper describes the implementation of a file system based distributed authoring system for campus-wide workgroups .In this we overcome the difficulty in changing a single document by different group members. Prior approaches operate on only specific documents but our system operates on any type of document. Each group member maintains one updatable copy of shared document. We also hoard read-only copies of each of these updatable copies in any other group member's node. All these copies are propagated to other group members at a rate that is solely dictated by the wireless user availability. Our user space file system prototype exhibits acceptable file system performance. A subjective evaluation showed that the moderation operation was intuitive for students.

I. INTRODUCTION

Just as in real life, a Moderator is someone who mediates; usually it's a person who presides over a meeting, forum, or group conversation in world. Our system works on two different mechanisms i.e. Centralized mechanism and Distributed Mechanism. In Distributed mechanism again there are two approaches Centralized approaches and Distributed approaches. Centralized approaches offer good availability but it required infrastructure before deployment of system. In centralized system all computing is controlled through a central terminal server, which centrally provides the processing, programs and storage. In distributed system every user has their own PC or laptop for processing, programs and storage. Storage is often mixed over the network between the local PC, shared PCs, or a dedicated file server. It do not required infrastructure before deployment of system. We implement distributed approach for our system.

II. GROUP AUTHORING SYSTEM

2.1 Structure of Group Authoring system

The systems such as Google Docs2, MS Office 11 for Mac are designed for co-authoring. Traditional editors (e.g., MS Office 2010 and earlier) are not group-aware. File system based mechanisms such as AFS and

NFS are application agnostic but they must rely on the limited interactions of editing applications. We use a file system based approach in order to operate on any document type.

Consider the example as shown in figure; here

We consider three groups 'Group A', 'Group B', and 'Group C'. In Group A there are two users C1 and C2. These users can change single document at the same time. The users from Group B and Group C have read only authority for document modified by users of Group A. We provide a dummy node for authentication of users. It stores all the information about any documents. When the any user is offline then we can see the all information about client at server side that is on dummy node

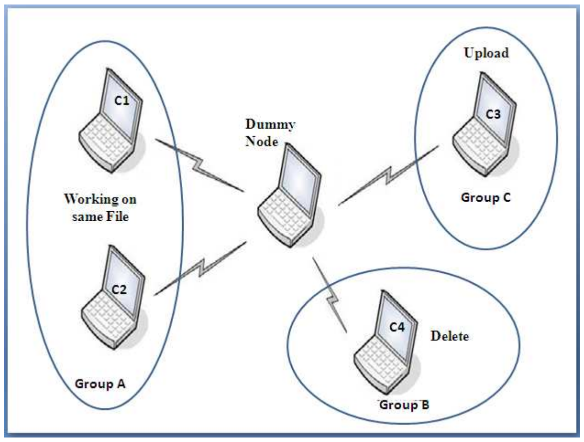

Fig. Group Authoring System

2.2 Proposed System Architecture

In our system, group members maintain a local copy of the shared contents. An Individual user contains one copy of his/her own documents in which he/she can update. As well as he/she contain the copies of other group members documents which are only readable. This means that he cannot modify the other's document. Each local update is reconciled with those of other group members using a pair wise reconciliation process. We require at least two group members to be simultaneously updating the documents.

As soon as the two members completed their own updating, their updated copies are stored to server. Then these updated documents are transmitted to other nodes by the server.

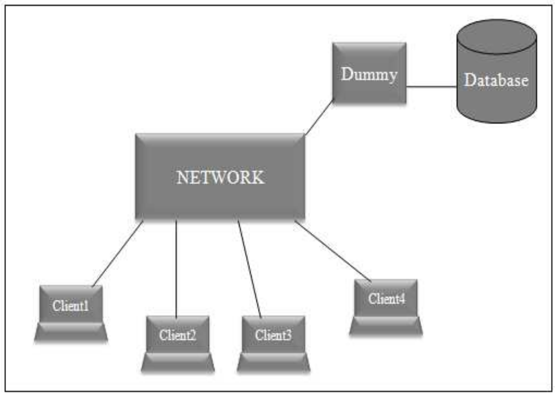

Fig: Architecture of distributed system

2.3 Updates

When user wants to modify document of the group member, the changes made by him or her are saved in temporary file and user has to upload that file. User can also delete file according to his/her authority. Two users can also update the same document at the same time. Changes made by these two members are automatically reconciled into single file. For example, suppose User A is working on file "Auther.txt" from 11.00 AM to 11.30 AM. User B is working on same file at 11.05 AM to 11.25 AM. When User A closed the file "Auther.txt" at 11.30 AM the changes made by the User A and User B is automatic reconciled into the file "Auther.txt".

III. CHARACTERISTICS OF DISTRIBUTED SYSTEM

i. No Fixed Infrastructure:-

The ad-hoc network in independent of fixed infrastructure, so it doesn't need any such kind of systems.

ii. Limited capacity of bandwidth:-

Each node act as host and router, this is called autonomous behavior.

iii. Energy constrained:-

As the each node in system is a moving, the most important factor in system is energy consumption by nodes.

iv. Nodes can join and leave network any time:-

It is wireless network, so to get connected to this network there is no need of any physical device to host. So it's easy to node to join or leave network any time.

v. Nature of operation for security, routing and host configuration is of distributed type which makes it easy to do so.

vi. High number of user capability.

IV. DISTRIBUTED MECHANISM

There are two ways to implement's distribution mechanism for group authoring systems that are centralized or distributed.

4.1.1 **Centralized approach**

A centralized synchronous system allows all group members to simultaneously modify the shared document. The system performance depends on the network latency; as the latency to the server increases, it becomes difficult to coordinate the shared modifications.

Exclusively locking the contents can address the server latency by avoiding simultaneous modifications. The exclusivity can be limited to the duration when the group member is online or until the lock was explicitly released. Systems such as Google Docs and MoonEdit8 allow non-locking and non-blocking access using operational transformation.

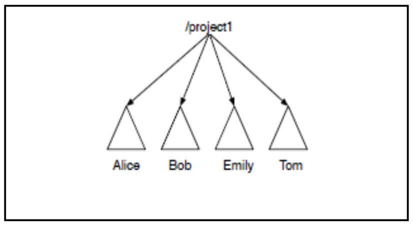

Fig: Centralized approach

Figure shows the shared copies for Alice, Bob, Emily and Tom are stored on the server. Alice is only allowed to modify contents under her directory ~/project1/alice, Bob under his own directory name, etc.

The examples of Centralized Systems are

http://knol.google.com

http://skydrive.live.com

4.1.2 Distributed approach

Distributed approaches do not require an infrastructure before collaboration systems can be deployed. However, they require mechanisms to locate the definitive version of the shared contents.

In an asynchronous system, group members maintain a local copy of the shared contents. Local updates are asynchronously reconciled with updates from other users using a pair wise distributed protocol.

Each user Figure b: ignore the user Dummy maintains their own copy while also hosting read-only copies of others' contents and thereby increasing the storage requirements in each of the users' laptops. Each group member manages the local storage overhead by selectively maintaining copies from specific members. For example, Emily only hosts her personal copy, while Alice also has a read-only copy of Emily's contents.

The examples of Distributed Systems are

http://moonedit.com

http://n-brain.net

V. Security Implications

Group management is fully distributed. System that generates password for users automatically by using username and random digits. We also provide the security to the dummy node. We provide a dummy node for authentication of users. It stores all the information about any documents.

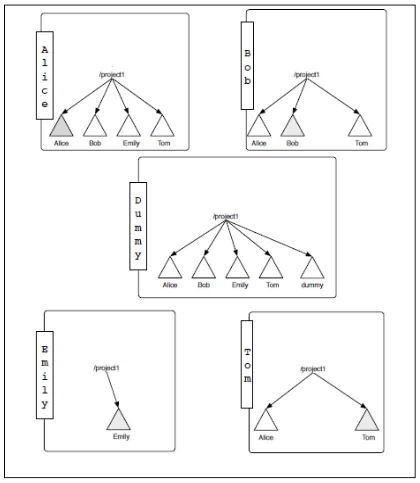

Fig Distributed approach

VI. Implementation Details

Systems such as Google Docs2, MS Office 11 for Mac and 3653 are designed for co-authoring. They are aware of each modification to the shared document and can resolve any conflicts. However, traditional editors (e.g., MS Office 2010 and earlier) are not group-aware. They cache all updates, eventually writing the entire updated document to a temporary copy and then atomically exchanging the temporary copy for the shared document. Systems such as Docx2Go extend Word for group authoring. On the other hand, file system based mechanisms such as AFS and NFS are application agnostic and allow the users to use shared documents in any format. They must rely on the limited interactions of editing applications with the file system to identify and resolve conflicts. We use a file system based approach in order to operate on any document type.

We implement the file system based distributed authoring system for campus-wide workgroups. We provide the login window for both sides for uses and for dummy node. In this password generated automatically using the combination of username and random digits. User modifies the documents of his/her group members only. User can modify any type of document (.doc,.txt,.ppt etc). Dummy node stores login information of all users and all updated documents of users.

VII. Acknowledgement

I express my deep sense of gratitude and appreciation towards my research guide Prof. A.B. Chougule for his continuous inspiration and valuable guidance in throughout project work.

References

[1] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in MobiCom '04, 2004, pp. 187–201.

[2] C. A. Ellis and S. J. Gibbs, "Concurrency control in group ware systems," SIGMOD Rec., vol. 18, no. 2, pp. 399–407, 1989.

[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.