The research and design of real-time collaborative document management system

Xiao Lv[a] ,LiangZhong Cui[b] ,Juan Li

College of Computer Engineering,Naval University of Engineering,Wuhan, 430033, China

[a]email: lvxiaoluxu@126.com, [b]email:13871162531@163.com

Keywords: component; real-time document management system; concurrncy control; consistency maintenance

Abstract. Different from traditional document management system, the real-time collaborative document management system pays more attention to support multiple users in different places to manage a document simultaneously . In order to support natural and harmonious human to human interactions in real-time collaborative document management system, a fully-replicated architecture is necessary to be adopted but a great challenge for the consistency maintenance . In this paper, the whole framwork of real-time documnet management system is proposed , then a concurrency control algorithm is adopted to ensure consistency in collaborative editing. At last, the real-time collaborative document management system is implemented and will be put into practical use .

Introduction

Document management systems can support to store all kinds of electronic document, view the document, edit the document, scan the document, upload and download the document and other functions. Specially, it is most important to design the total document library for editing and archiving . Most traditional document management systems adopt the centralized database to manage the documents,which need a costly central service . And most traditional document management systems can't support different users from different places edit documents simultaneously, which can't improve the efficiency.Moreover, most existing document management are not suitable to massive real-time collaborative distributed environment.

Computer Supported Cooperative Work can greatly improve the way people communicate and work[1]. Real-time collaborative work system is a classic theme in this area, and can support natural and harmonious human to human interactions[2]. A fully-replicated architecture is adopted to permit many users from different places to deal with the same object simultaneously[3].

Based on this background, we propose the design idea of real-time collaborative document management systems, and will pay more attention to discuss the consistency maintenacne technology in real-time collaborative document editing. At last, we select a typical concurrent control algorithm to ensure consistency in a real-time document editing, which can be a good guidence to acheive a real-time collaborative document management system.

The Whole Framework

This section we introduce a scheme composed of design and achievement of real-time collaborative document management system .

In order to support different users to mannage the documents at the same time, we adopt a fully replicated database and document management system. The whole framework is shown in Fig.1.

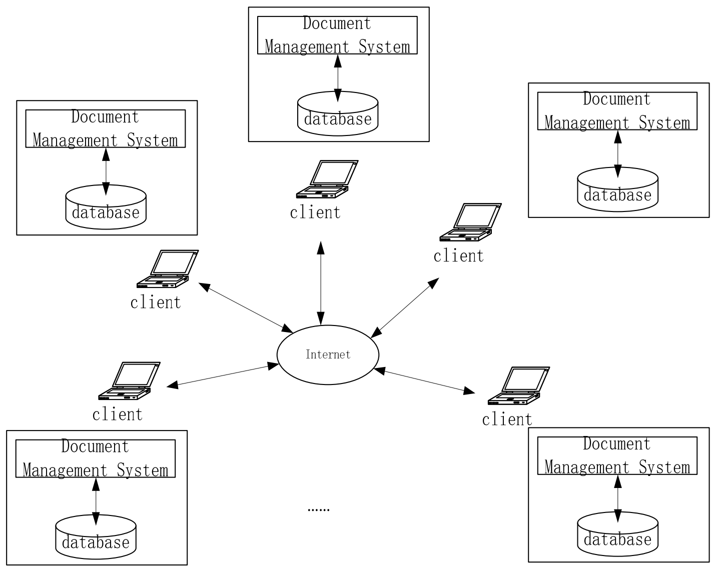

Fig.1. The framework of a real-time collaborative document management system.

Many clients can concurent mannage the document without imposing any restrictions to users' actions. In other words, users can add, delete, modify and view the same document or any document in any manner at any time in the real-time collaborative document system.

Local changes by users' any actions are applied to the local system and database immediately, which can support a harmonious human to computer interaction.

Local changes are propagated to other remote clients through Internet to keep all clients consistent. But the propagation brings benefit and disadvangtage. Propagation of changes can make use of network resource utilized efficiently, but also can bring communication latency. Replicated arichitecture can hide communication latency and support real-time collaborative management,but it may cause inconsistency problems.

Therefore when remote clients receive the changes from other clients, they will apply these changes to the their system and database in accordance with some concurrent control mechanisms.

Consistency maintenance is an important problem in collaborative management with replicated architecture and is a hot research point in collaborative computing. In next section, we will take collaborative document editing as an example,select a concurrent control algorithm to achieve the consistency of all clients.

Consistency Maintenance in Collaborative Document Editing

A. Related technology

Replication is necessary in collaborative editing to achieve high responsiveness. Many methods dealing with replication have been proposed. In the domains of database repilcation, pessimistic methods are usually used, which cannot ensure high rensponsiveness for real-time collaboration. Because local changes should aquire an exclusive access, changes cannot be applied immediately before resovling conflicts.

Optimistic repilication is a family of methods to be suitable to real-time collaboration. Operational Transformation(OT) is an optimistic concurrency control algorithm which has been seen as a proper technology to achieve nonblocking real-time collaboration[1,4-7]. Local operations are executed immediately at local peers. Remote operations are transformed against concurrent operations uopn their reception at other copies. In order to achieve convergence, OT adopt two approaches : one is to design specific transformation functions which can be capable of preserving TP1/TP2; another is to design a total order transformation path capatable of avoiding TP1/TP2[8, 9]. However, more research has found the first method is very difficult to achieve, the second method is always to ensure the convergence.

The main problem of OT is scalability. OT algorithms use version vectors or central timestamps to detect concurrent operations which cannot scale well in massive peer to peer envrionments.

Recently, a new class of mechanisms called CRDT(commutatibe replicated data types) have been proposed[10-16]. Concurrent operaions are designed to be commutative by using the characteristics of abstract data types. By associated each object with an unique and totallly ordered identifier, let all objects totally store in the data types and can ensure convergence for all copies.

Experiments results show that CRDT algorithms outperform OT algorithms by a factor between 25and 1000.

B. RGA (Replicated Growable Array) algorithm

RGA is a CRDT that can support not only insertion and deletion but alse update operations[16]. All sites have a linked list and a hash table. A linked list represent the total order of objects. A hash table reserve the pointer to the node in linkedn list. The data structue is shown in Fig. 2.

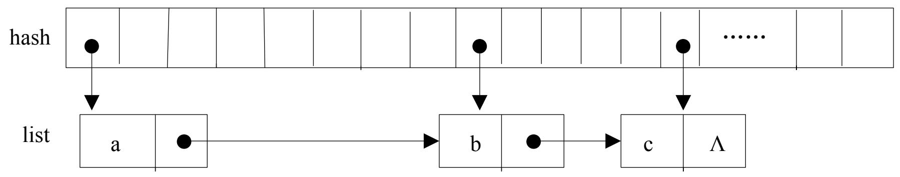

Fig.2. hash table and linked list in RGA

Every operation object has an s4vector including four integers, which is used as a unique index and used to resolve conflits between concurren operation. An s4vector is defined as <int ssn, int sid, int sum, int seq>, ssn is a global session number that increases automatic, sid is the site ID unique to the site, sum is the sum of the state vector, seq is reserved for purging tombstones .

Local operations find their target elements with integer indexes. If the operation is insertion, link the object after the target element in the linked list. If the operation is deletion, make the object tombstone in the linked list Meanwhile, remote operations retrieve their target elements via the hash table with their unique s4vector. A remote insertion needs to compare its s4vector to other concurrent insertion objects' s4vector at the same position, insert its object in a proper position. A remote deletion make the target element tomstone.

C. Collaborative Document Editing Simulation System

In this section, we implement RGA algorithm using C# language in visual studio 2010. Based on RGA algorithm,we design a collaborative document editing simulation system, which can ensure consistent for all client s at last.

1) Implement RGA algorithm

The local algorithm for insert is shown is Fig.3.The local algorithm for delete is shown is Fig.4. The remote algorithm for insert is shown is Fig.5.The remote algorithm for delete is shown is Fig.6.

```
private void LocalInsert(RGAOperation op)
      { RGANode newnd = new RGANode(op.gets4vector(), op.getContent());
        if (op.getIntPos() == 0)
        {  newnd.setNext(head.getNext());
           head.setNext(newnd);
         }
        else
        {
           RGANode target = getVisibleNode(op.getIntPos());
           if (target == null) throw new Exception( "Don't find " + op.getIntPos());
           newnd.setNext(target.getNext());
           target.setNext(newnd);
         }
        hash.Add(op.gets4vector(), newnd);
      }
```

Fig.3. Local algorithm for insert of RGA

```
private void LocalDelete(RGAOperation op)
    {
       RGANode node = getVisibleNode(op.getIntPos());
       if (node == null) throw new Exception("Don't find " + op.getIntPos());
       node.makeTombstone();
     }
```

Fig.4. Local algorithm for delete of RGA

```
private void RemoteInsert(RGAOperation op)
    { RGANode newnd = new RGANode(op.gets4vector(),op.getContent());
      RGANode prev, next;
       RGAS4Vector s4v = op.gets4vector();
       if (op.getPos() == null) prev = head;
       else prev = (RGANode)hash[op.getPos()];
       next = prev.getNext();
       while (next != null)
       { if (s4v.compareTo(next.getKey()) == RGAS4Vector.AFTER)
            break;
          prev = next;
            next = next.getNext(); }
       newnd.setNext(next);
       prev.setNext(newnd);
       hash.Add(op.gets4vector(), newnd);
     }
```

Fig.5. Remote algorithm for insert of RGA

```
private void RemoteDelete(RGAOperation op)
     {
        RGANode node = (RGANode)hash[op.getPos()];
        if (node == null) throw new Exception("Cannot find" + op.getPos());
        node.makeTombstone();
     }
```

Fig.6. Remote algorithm for delete of RGA

2) Instance Analysis

In this section, we take a specific collaborative editing scenario as an example, then give all the steps of opearations execution in all sites and consistent result theoretically. Fig.7. shows a collaborative scenario, suppose three sites from the same initial state "abc". Three sites concurrently generates O_1, O_2, O_3 respectively. s_j^i represents the jth state of the ith site. $\cancel{\beta}$ represents tombstone of the character β .

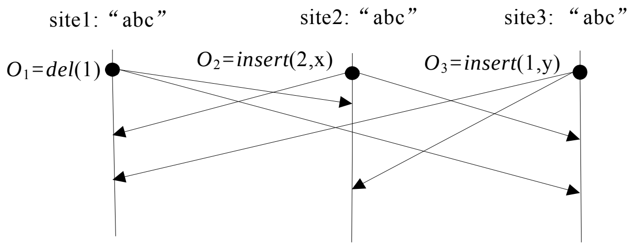

Fig.7. A collaborative scenario

The state vector of all operations is SV_{O1}=(1,0,0), SV_{O2}=(0,1,0),SV_{O3}=(0,0,1). Suppose the session number of the initial state "abc" is 0, the session number of all operations generated in is 1. The s4vectors of "a", "b" and "c" are s4vector(*a*)=(0,1,1,0),s4vector(*b*)=(0,2,1,0),s4vector(*c*)= (0,3,1,0). The s4vectors of all operations are s4vector(O_1)=(1,1,1,1),s4vector(O_2)=(1,2,1,0),s4vector(O_3).

At site1, execute O_1, set "b" as tombstone, s_1^1="aƀc". When receive O_2, execute O_2, s_2^1="aƀxc". When receive O_3, execute O_3, s_3^1="ayƀxc". At site2, execute O_2, s_1^2="abxc". When receive O_1, execute O_1, s_2^2="aƀxc". When receive O_3, execute O_3, s_3^2="ayƀxc". At site3, execute O_3, s_1^3="aybc". When receive O_2, execute O_2, s_2^3="aybxc".When receive O_1, execute O_1, s_3^3="ayƀxc". At last ,all three sites get the same result "ayƀxc" .

3) Collaborative Simulation System

We design a collaborative simulation system based on RGA algorithm. Fig.8. shows the interface of the system.

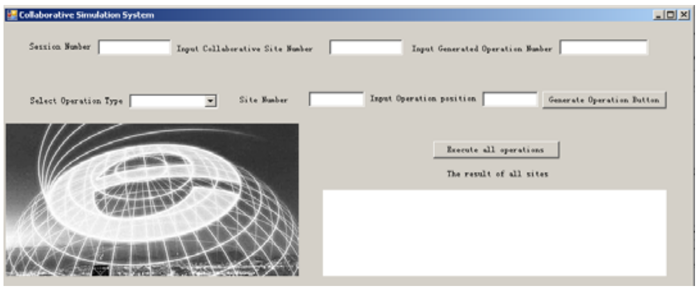
Collaborative Simulation System
Session Number
Input Collaborative Site Number
Input Generated Operation Number
Select Operation Type
Site Number
Input Operation position
Generate Operation Button
Execute all operations
The result of all sites

Fig.8. The collaborative simulation system

In Fig.8, we need to input session number, site numbers and integrated operations. Then we can select operation type: delete or insert. Then we need to input operation position. When we click the button "Generate Operation Button", all operations have been generated, then when we click the button "Execute all operation" , all operations have been executed. The result of all sites will be shown in the interface. In this system, we use the function operation.random() to generate all operations, pos.random() to generate random positions, character.random() to generate random characters.

Conclusion

In this paper, we propose the whole framework of collaborative document management system. A fully-replicated architecture is to achieve high responsiveness but a great challenge for consistency maintenace. Then we take collaborative document editing as an example, select RGA algorithm to show

how to achieve convergence. At last, we design a collaborative simulation system, and RGA algorithm have been applied to the system to achieve the convergence, which is a good guidance for the design of real- time collaborative document management system.

References

[1] Ellis C A, Gibbs S J. “Concurrency control in groupware systems,”Proceedings of ACM SIGMOD international conference on Management of data. Portland, Oregon, USA, 1989, 18(2): 399-407.

[2] Saito Yasushi, Shapiro Marc. “Optimistic replication,” ACM Computing Surveys, 2005, 37(1): 42-81.

[3] Shi Mei-Lin, Xiang Yong, Wu Shang-Guang. “Cooperative science-from synergetics to CSCW,”Journal of Tsinghua University (Sci & Tech), 1997,37: 85–88.

[4] Ressel M, Nitsche-Ruhland D, Gunzenhäuser R. “An integrating, transformation-oriented approach to concurrency control and undo in group editors,”Proceedings of the ACM conference on Computer supported cooperative work. Boston, MA, USA, 1996: 288-297.

[5] Prakash A, Knister M J. “A framework for undoing actions in collaborative systems,” ACM Transactions on Computer-Human Interaction (TOCHI), 1994, 1(4): 295-330.

[6] Sun Cheng-Zheng, Ellis C A. “Operational transformation in real-time group editors: issues, algorithms, and achievements,”Proceedings of the ACM conference on Computer Supported Cooperative Work. Seattle, WA, USA, 1998: 59-68.

[7] Vidot N, Cart M, Ferrié J, Suleiman M. “Copies convergence in a distributed real-time collaborative environment,”Proceedings of the ACM conference on Computer Supported Cooperative Work. Philadelphia, PA, USA, 2000: 171-180

[8] Imine A, Molli P, Oster G, Rusinowitch M. “Proving correctness of transformation functions in real-time groupware,”Proceedings of the 8th European Conference of Computer-supported Cooperative Work.Helsinki, Finland, 2003: 277-293.

[9] Sun C, Xu Y, Agustina A. “Exhaustive search of puzzles in operational Trans-formation,”Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing. Baltimore, MD, USA , 2014: 519-529.

[10] Preguica N, Marques J M, Shapiro M, Letia M. “A commutative replicated data type for cooperative editing,”Proceedings of the 29th IEEE International Conference on Distributed Computing Systems. Montreal, QC,Canada,2009: 395-403.

[11] Shapiro M, Preguica N. “Designing a commutative replicated data Type,” France,Inria, Report: RR-6320, 2007 .

[12] Wu Q, Pu C, Ferreira J E. “ A partial persistent data structure to support consistency in real-time collaborative editing,”Proceedings of 26th IEEE International Conference on Data Engineering (ICDE).Long Beach, CA,USA, 2010: 776-779.

[13] Wu Q, Pu C. “Consistency in real-time collaborative editing systems based on partial persistent sequences,” Georgia:Georgia Institute of Technology, Report: GIT-CERCS-09-07, 2009.

[16] Roh H G, Jeon M, Kim J S, Lee J. “Replicated abstract data types : Building blocks for collaborative applications,” Journal of Parallel and Distributed Computing, 2011, 71(3): 354-368.