Optimistic Access Control Model for Collaborative Editing Systems

Presented by Asma Cherif

Cassis Team, Loria

November 29, 2010

<ロ> (日) (日) (日) (日) (日)

Outline

Introduction Our Coordination Model Consistency and Security Issues Implementation and Evaluation A Garbage Collector for Collaborative Editors Conclusion & Future Work

Introduction

Our Coordination Model

Consistency and Security Issues

Implementation and Evaluation

A Garbage Collector for Collaborative Editors

Conclusion & Future Work

Introduction

Distributed Collaborative Editors Motivations

Distributed Collaborative Editors (DCE) provide computer support for modifying simultaneously shared documents:

- articles,
- wiki pages and
- programming source code

by dispersed users.

Example

- Google Docs
- Coword, Copowerpoint
- ▶ etc.

DCE Requirements

Distributed Collaborative Editors Motivations

DCE must consider human factors

- High responsiveness
- High concurrency
- Consistency
- Decentralized coordination
- Scalability

Distributed Collaborative Editors Motivations

Examples of Real-time Collaborative Editors

- centralized (global and unique order of execution):
 - SOCT4 [Vidot00], GOT [Sun98]
 (-) Client-Server architecture
- decentralized (arbitrary order of execution):
 - adOPTed [Ressel:96], SOCT2 [Suleiman98], GOTO [Sun98], and SDT [Li04]

(-) fixed number of users (use of state vectors to detect causality relationship)

OPTIC [Imine08]: dynamic groups

・ロト ・回ト ・ヨト ・ヨト

Objective

Distributed Collaborative Editors Motivations

Build a generic access control layer:

- suites existing editing solutions characterized by
 - replication
 - log usage
 - undo procedure

 responds to DCEs requirements without adding overhead

<ロ> (日) (日) (日) (日) (日)

Distributed Collaborative Editors Motivations

Challenges

- Ensuring security is a challenging problem in these applications:
 - Balancing the computing goals of collaboration and access control to shared information,
 - Users can join and leave at any time so we have to allow for dynamic change of access rights (scalability requirement).
 - High performance: access rights management should be replicated (responsiveness requirement).

・ロン ・回と ・ヨン ・ヨン

Shared Data Object Shared Policy Object Collaboration Protocol

Our Coordination Model

Our coordination model is composed by two shared and replicated objects:

- Data Object
- Policy Object

・ロン ・回と ・ヨン・

æ

Shared Data Object Shared Policy Object Collaboration Protocol

Shared Data Object

The data object is modeled by a list of

characters, paragraphs, pages, pixels, XML nodes, ...

Operations altering data object state: cooperative operations

Example

- Insert
- Delete
- Update

Shared Data Object Shared Policy Object Collaboration Protocol

Shared Policy Object

We specify an authorization policy by three sets:

- set of subjects (*i.e.* users)
- set of objects
- set of access rights

Policy

A policy is a function that maps a set of subjects and a set of objects to a set of signed rights.

Shared Data Object Shared Policy Object Collaboration Protocol

Shared Policy Object

Authorization rule

$$(\{s_1, s_2, \ldots, s_n\}, \{o_1, o_2, \ldots, o_p\}, \{r_1, r_2, \ldots, r_q\}, +/-)$$

Ownership

each user s_i has its own policy P_i to administrate his objects \Rightarrow Administration distribution

Shared Data Object Shared Policy Object Collaboration Protocol

Shared Policy Object

Two kinds of administrative operations

- AddAuth
- DelAuth

Global Policy Object

The set of all owner policies

Shared Data Object Shared Policy Object Collaboration Protocol

Collaboration Protocol

1. Check a local operation against the appropriate owner policy object.

・ロン ・回と ・ヨン・

Shared Data Object Shared Policy Object Collaboration Protocol

Collaboration Protocol

- 1. Check a local operation against the appropriate owner policy object.
- 2. Once granted and executed, local operations are broadcast to other users.

・ロト ・回ト ・ヨト ・ヨト

Shared Data Object Shared Policy Object Collaboration Protocol

Collaboration Protocol

- 1. Check a local operation against the appropriate owner policy object.
- 2. Once granted and executed, local operations are broadcast to other users.
- 3. When received by a user, remote operations are checked against the appropriate owner policy object.

Shared Data Object Shared Policy Object Collaboration Protocol

Collaboration Protocol

- 1. Check a local operation against the appropriate owner policy object.
- 2. Once granted and executed, local operations are broadcast to other users.
- 3. When received by a user, remote operations are checked against the appropriate owner policy object.
- When an administrator modifies its owner policy object, modifications are sent to other users in order to update their local copies.

Shared Data Object Shared Policy Object Collaboration Protocol

Is convergence maintained in presence of administrative operations



æ

イロン イヨン イヨン イヨン

Out-of-order Execution of Cooperative Operations Out-of-order Execution of Coop & Admin Operations

イロト イヨト イヨト イヨト

Consistency and Security Issues

- Out-of-order Execution of Cooperative Operations.
- Out-of-order Execution of Cooperative and Administrative Operations.

Out-of-order Execution of Cooperative Operations Out-of-order Execution of Coop & Admin Operations

イロト イヨト イヨト イヨト

Out-of-order Execution of Cooperative Operations

Cooperative operations may arrive in different order at different sites.

Issue

This can lead to undesirable situations in collaborative editors: data divergence.

Use OT approach to overcome this problem

Out-of-order Execution of Cooperative Operations Out-of-order Execution of Coop & Admin Operations

<ロ> (日) (日) (日) (日) (日)

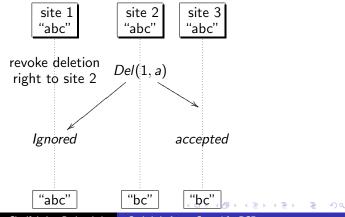
Out-of-order Execution of Cooperative and Administrative Operations

Performing cooperative and administrative operations in different orders at every user site :

- Security holes
- Data divergence

Out-of-order Execution of Cooperative Operations Out-of-order Execution of Coop & Admin Operations

Divergence caused by introducing administrative operations (Scenario1)



Cherif, Imine, Rusinowitch

Optimistic Access Control for DCEs

Out-of-order Execution of Cooperative Operations Out-of-order Execution of Coop & Admin Operations

Divergence caused by introducing administrative operations (Scenario1)

lssue

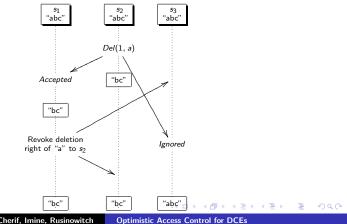
Detect causality between administrative and cooperative operations?

Solution

- Give priority to administrative operations
- Temporary policy violation: optimistic access control
- Recover the last correct state : Undo process

Out-of-order Execution of Cooperative Operations Out-of-order Execution of Coop & Admin Operations

Divergence caused by introducing administrative operations (Scenario 2)



Cherif, Imine, Rusinowitch

Out-of-order Execution of Cooperative Operations Out-of-order Execution of Coop & Admin Operations

・ロト ・回ト ・ヨト

Divergence caused by introducing administrative operations (Scenario 2)

Issue

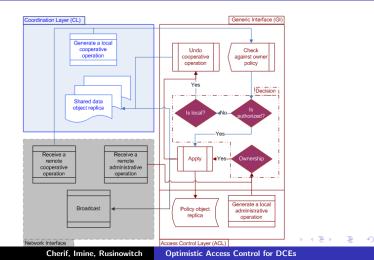
Detect causality between an administrative and cooperative operation?

Solution

 Each modification seen by the object administrator is considered as valid

Out-of-order Execution of Cooperative Operations Out-of-order Execution of Coop & Admin Operations

Access Control Flow Chart



Implementation: p2pAgenda (shared calendar)

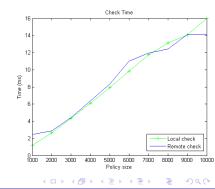
😡 p2pAgenda - alpha			Select Peer(s)	Select peragraph(s)
Menu Edit Administration Help			AL	All Document
₩			black berry	
Connected	agenda			
Try now to create (join a group!	Date Description Place Auth 15/04/09 - 12:30 Professional Meetin Nancy, place Stanislas/cassis	or Select		Select Right(s)
	25/09/10 - 08:30 CoopIs Conference Greece cassis			
19				Insert V Delete
List of peers				Read Update
cassis				
bladi berry				
				Select Permission Type
				Grant
	Insert Delete Update			O Revoke
Edit Policy				
View Rights				
SAVE			Add Del	At position
		0	Authorization List All: DOC:rd::+::0	Administrative Log AddAuth(All::DOC::d::+,0)
		0	AEDOCIMITER	AddAuth(All:DOCIIdII+,0)
			Cancel	Confirm

A desinder tration

Cherif, Imine, Rusinowitch Optimistic Access Control for DCEs

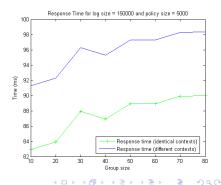
Evaluation: response time

- Worst case: last rule matches the cooperative operation.
- check time is linear
- \blacktriangleright \leq 16 ms (local operations)
- \blacktriangleright \leq 14 ms (remote operations)
- use of Hashset



Evaluation: response time

- Worst case: integrate a remote insert.
- Response time is $\leq 100 ms$ for
- log size = 150000 (100%ins)
- Owner policy size = 5000
- peers number= 80



Objective & Motivations Garbage issues Garbage protocol Implementation & performance study Pros and Cons

イロン イヨン イヨン イヨン

A Garbage Collector for Collaborative Editors

Motivations

- Mobile Devices such as PDAs and cell phones are becoming more and more pervasive (PDAs, iPhones, ...).
- Several works try to integrate desktop applications on these devices.

Challenges

- mobile devices: low space memory, slow wireless connections
- collaborative editing applications: log usage (convergence)
 arge storage space
- ▶ Big logs ⇒ high delays

Principle

Objective & Motivations Garbage issues Garbage protocol Implementation & performance study Pros and Cons

・ロン ・四 と ・ ヨ と ・ ヨ と

- ► Garbage ⇒ global view on the collaborative system (clean while maintaining convergence)
- Existing solution [Sun98]:
 - state vector usage \implies scalability issue
 - identical logs

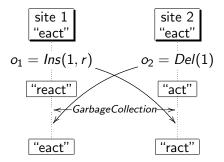
Key idea

Build a global view on the collaboration state in a scalable fashion.

Objective & Motivations Garbage issues Garbage protocol Implementation & performance study Pros and Cons

イロン イヨン イヨン イヨン

Divergence caused by garbage collection applied on different contexts



Objective & Motivations Garbage issues Garbage protocol Implementation & performance study Pros and Cons

イロン イヨン イヨン イヨン

Issue

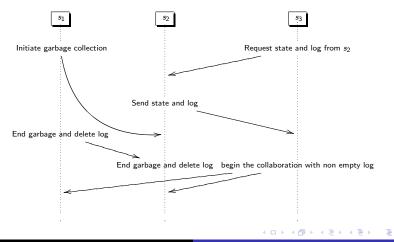
Execution order between garbage operations and cooperative operations

Solution

Wait until agreement about logs (global view)

Objective & Motivations Garbage issues Garbage protocol Implementation & performance study Pros and Cons

Divergence caused by a new user joining the group



Cherif, Imine, Rusinowitch Optimistic Access Control for DCEs

Objective & Motivations Garbage issues Garbage protocol Implementation & performance study Pros and Cons

イロン イヨン イヨン イヨン

lssue

New user is not aware about garbage initiation

Solution

Enforce new or disconnected users to wait until garbage ends

Garbage messages

Objective & Motivations Garbage issues Garbage protocol Implementation & performance study Pros and Cons

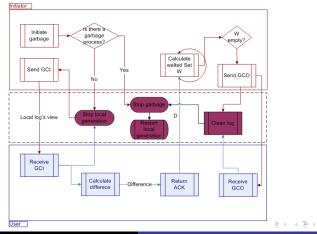
・ロン ・回と ・ヨン ・ヨン

Exchange garbage collection messages

- ▶ Initiation (GCI): initiator identity + local view of log.
- Acquirement (ACK): difference between local and initiator views of log.
- Ordre (GCO): garbage order.

Objective & Motivations Garbage issues Garbage protocol Implementation & performance study Pros and Cons

Garbage protocol



Cherif, Imine, Rusinowitch

Optimistic Access Control for DCEs

Objective & Motivations Garbage issues Garbage protocol Implementation & performance study Pros and Cons

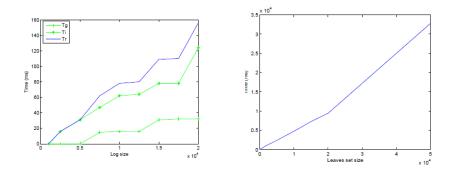
・ロン ・回 と ・ ヨ と ・ ヨ と

3

My first te	st				
Save					
Garbage Collection					
Peer List					
	Disconnect				
•		•			
<u>م</u>		•			

Implementing OPTIC with garbage collector on CDC mobile phones

Objective & Motivations Garbage issues Garbage protocol Implementation & performance study Pros and Cons



Response Time for the CDC environment

Cherif, Imine, Rusinowitch Optimistic Access Control for DCEs

Garbage collection time

Pros & cons

Objective & Motivations Garbage issues Garbage protocol Implementation & performance study Pros and Cons

イロト イヨト イヨト イヨト

- ► (+) scalable
- (+) cleaning equivalent logs
- (+) good performance
- (-) blocking solution: local generation is stopped during garbage collection

Contribution

A first step towards garbage collection in collaborative editors to extend them for mobile devices.

Conclusion

- Contributions:
 - New framework for controlling access in collaborative editing work based on Optimistic Access Control.
 - Description and checking of our authorization policies are very simple (to maintain high responsiveness).
 - Good behavior with large logs.
 - Propose an approach to garbage logs.
 - Performance evaluation with a large scale distributed platform grid5000.
- Perspectives: Deploy the model for social network applications
 - Existing applications are centralized
 - Allow every user to manage access control for his objects

ロン (雪) (目) (目)