



Memory optimization strategies for linear mappings and indexation-based shared documents

M. Mumtaz Ahmad

Loria, INRIA Nancy - Grand Est, UHP Nancy - 1, IAEM Lorraine

November 14, 2011





Outline

1 General introduction

- Speeding up computations
- Optimizing indexed communications

2 Sequential break down of operations

- Problem description
- Contributions

3 Decentralized collaborative editing system

- Problem description
- Contributions

4 Conclusions

- Summary
- Research directions

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions



Outline

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions

1 General introduction

- Speeding up computations
- Optimizing indexed communications

2 Sequential break down of operations

- Problem description
- Contributions

3 Decentralized collaborative editing system

- Problem description
- Contributions

4 Conclusions

- Summary
- Research directions



Speeding up computations

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

Higher performance

- Modern computer technology has increased the demand for higher performance in all areas of computing.

Significant growth in

- 1 Hardware performance
- 2 Architecture evolution

Stress is raised on

- 1 Compiler technology
- 2 Research communities





Speeding up computations

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

Higher performance

- Modern computer technology has increased the demand for higher performance in all areas of computing.

Significant growth in

- 1 Hardware performance
- 2 Architecture evolution

Stress is raised on

- 1 Compiler technology
- 2 Research communities





Speeding up computations

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

Higher performance

- Modern computer technology has increased the demand for higher performance in all areas of computing.

Significant growth in

- 1 Hardware performance
- 2 Architecture evolution

Stress is raised on

- 1 Compiler technology
- 2 Research communities





Micro-processor

- General-purpose PCs
- Servers to desktop, laptop
- Cell-phones such as iPhone



Permanent challenge

- * Increasing the performance

One of the possible approaches

- Sequential break down of operations



General
introduction

Speeding up
computations

Optimizing indexed
communications

Sequential
break down of
operations

Problem description
Contributions

Decentralized
collaborative
editing
system

Problem description
Contributions

Conclusions

Summary
Research directions



Micro-processor

- General-purpose PCs
- Servers to desktop, laptop
- Cell-phones such as iPhone



Permanent challenge

- * Increasing the performance

One of the possible approaches

- Sequential break down of operations



General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Micro-processor

- General-purpose PCs
- Servers to desktop, laptop
- Cell-phones such as iPhone



Permanent challenge

- * Increasing the performance

One of the possible approaches

- Sequential break down of operations



General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Hello, do you remember me.....

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions





Optimizing indexed communication

Computer networks inspired RCE

- Rapid development of computer networks inspired the advancement of RCE (real time collaborative editors).

In collaborative editing

- Participants are not bound to be in the same location
- Editing shared textual document, to draw a shared graph structure, record ideas during brainstorming meetings.



Documents could be

- Articles, wiki pages and programming source code.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Optimizing indexed communication

Computer networks inspired RCE

- Rapid development of computer networks inspired the advancement of RCE (real time collaborative editors).

In collaborative editing

- Participants are not bound to be in the same location
- Editing shared textual document, to draw a shared graph structure, record ideas during brainstorming meetings.



Documents could be

- Articles, wiki pages and programming source code.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Optimizing indexed communication

Computer networks inspired RCE

- Rapid development of computer networks inspired the advancement of RCE (real time collaborative editors).

In collaborative editing

- Participants are not bound to be in the same location
- Editing shared textual document, to draw a shared graph structure, record ideas during brainstorming meetings.



Documents could be

- Articles, wiki pages and programming source code.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Conflicts may occur

- In DCE (decentralized collaborative editing), multiple copies of the same document may create confusions.

- To reconcile conflicts, indexing the contents of shared document is considered as basic requirement.

Precision control indexing method (to design)

- Decentralized collaborative editing system





Conflicts may occur

- In DCE (decentralized collaborative editing), multiple copies of the same document may create confusions.
- To reconcile conflicts, indexing the contents of shared document is considered as basic requirement.

Precision control indexing method (to design)

- Decentralized collaborative editing system





Conflicts may occur

- In DCE (decentralized collaborative editing), multiple copies of the same document may create confusions.
- To reconcile conflicts, indexing the contents of shared document is considered as basic requirement.

Precision control indexing method (to design)

- Decentralized collaborative editing system





Outline

1 General introduction

- Speeding up computations
- Optimizing indexed communications

2 Sequential break down of operations

- Problem description
- Contributions

3 Decentralized collaborative editing system

- Problem description
- Contributions

4 Conclusions

- Summary
- Research directions

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions



Problem description

Restriction

- A 32 bit processor performs operations on 32 bits.
- Any transformation of data structure required decomposition in successive operations on 32 bits.

Old Strategy

- To exchange contents of registers, it makes copies of the initial data.

Errors

- More number of registers are required
- Loss of speed/power consumption
- Signal duplication occurs

General
introduction

Speeding up
computations
Optimizing indexed
communications

Sequential
break down of
operations

Problem description
Contributions

Decentralized
collaborative
editing
system

Problem description
Contributions

Conclusions

Summary
Research directions



Problem description

General introduction

Speeding up computations
Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

Restriction

- A 32 bit processor performs operations on 32 bits.
- Any transformation of data structure required decomposition in successive operations on 32 bits.

Old Strategy

- To exchange contents of registers, it makes copies of the initial data.

Errors

- More number of registers are required
- Loss of speed/power consumption
- Signal duplication occurs



Problem description

Restriction

- A 32 bit processor performs operations on 32 bits.
- Any transformation of data structure required decomposition in successive operations on 32 bits.

Old Strategy

- To exchange contents of registers, it makes copies of the initial data.

Errors

- More number of registers are required
- Loss of speed/power consumption
- Signal duplication occurs

General
introduction

Speeding up
computations
Optimizing indexed
communications

Sequential
break down of
operations

Problem description
Contributions

Decentralized
collaborative
editing
system

Problem description
Contributions

Conclusions

Summary
Research directions



Proposed strategy

in situ design of computation

- It does not require any extra variable other than the variables available as input

Example

- Let $E : \{0, 1\}^2 \rightarrow \{0, 1\}^2$ be the mapping defined as

$$E(x_1, x_2) = (x_2, x_1)$$

- To compute mapping E by a sequence of operations, we have two options:

- A basic program that completes the task.

- An *in situ* program that completes the task.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Proposed strategy

in situ design of computation

- It does not require any extra variable other than the variables available as input

Example

- Let $E : \{0, 1\}^2 \rightarrow \{0, 1\}^2$ be the mapping defined as

$$E(x_1, x_2) = (x_2, x_1)$$

- To compute mapping E by a sequence of operations, we have two options:

- A basic program that completes the task.

- An *in situ* program that completes the task.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Proposed strategy

in situ design of computation

- It does not require any extra variable other than the variables available as input

Example

- Let $E : \{0, 1\}^2 \rightarrow \{0, 1\}^2$ be the mapping defined as

$$E(x_1, x_2) = (x_2, x_1)$$

- To compute mapping E by a sequence of operations, we have two options:

- A basic program that completes the task.

- An *in situ* program that completes the task.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

Comparison

Old strategy

$$x_0 := x_1$$

$$x_1 := x_2$$

$$x_2 := x_0$$

In situ strategy

$$x_1 := f_1(x_1, x_2)$$

$$x_2 := f_2(x_1, x_2)$$

$$x_1 := g_1(x_1, x_2)$$

$$f_1(x_1, x_2) = f_2(x_1, x_2) = g_1(x_1, x_2) = x_1 \oplus x_2$$

Noticed that

The *in situ* strategy does not make use of an extra variable x_0 which involved in basic strategy.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Comparison

Old strategy

$$x_0 := x_1$$

$$x_1 := x_2$$

$$x_2 := x_0$$

In situ strategy

$$x_1 := f_1(x_1, x_2)$$

$$x_2 := f_2(x_1, x_2)$$

$$x_1 := g_1(x_1, x_2)$$

$$f_1(x_1, x_2) = f_2(x_1, x_2) = g_1(x_1, x_2) = x_1 \oplus x_2$$

Noticed that

The *in situ* strategy does not make use of an extra variable x_0 which involved in basic strategy.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

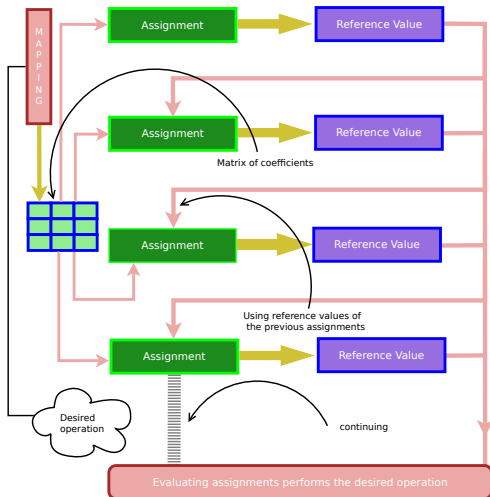
Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

How it works



General introduction

- Speeding up computations
- Optimizing indexed communications

Sequential break down of operations

- Problem description
- Contributions

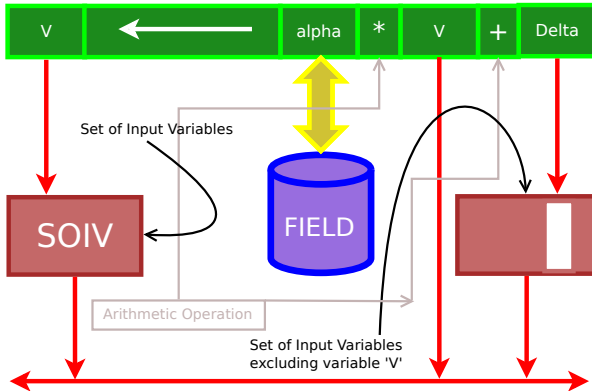
Decentralized collaborative editing system

- Problem description
- Contributions

Conclusions

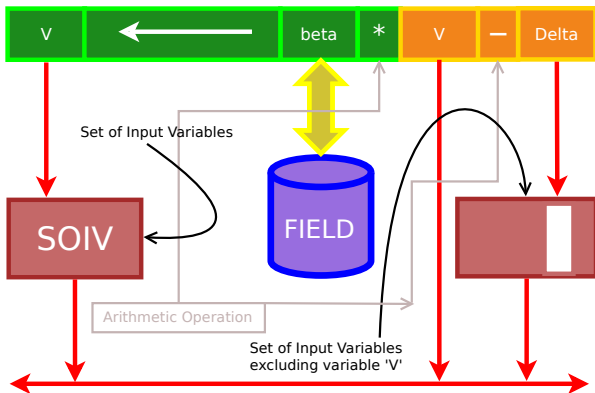
- Summary
- Research directions

Interpreting an assignment



- General introduction
 - Speeding up computations
 - Optimizing indexed communications
- Sequential break down of operations
 - Problem description
 - Contributions
- Decentralized collaborative editing system
 - Problem description
 - Contributions
- Conclusions
 - Summary
 - Research directions

Interpreting reference value



General introduction

- Speeding up computations
- Optimizing indexed communications

Sequential break down of operations

- Problem description
- Contributions

Decentralized collaborative editing system

- Problem description
- Contributions

Conclusions

- Summary
- Research directions



Alternate strategy over fields

- IDC strategy to perform *in situ* computation has been proposed by Burckel et al [Bur07].

Alternate Approach

- A strategy based on Bézout's identity

Theorem

Every linear mapping $E: (x, y) \rightarrow (mx + ny, px + qy)$, where $m, n, p, q \in \mathbb{Z}$, can be computed by a sequence of at most 3 linear assignments with rational coefficients.

▶ Proof

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions



Investigating bound for IDC over integers

Objective

- To find the minimum number of assignments required to compute the mapping sequentially using *in situ* strategy.

Theorem

Let

$$E_n : (x, y) \longrightarrow (F_{n-1}x + F_n y, F_n x + F_{n+1} y)$$

be the mapping over \mathbb{Z}^2 , where F_n is the Fibonacci number.

Then the *mapping E_{4k+2} is computed with $2k+2$ assignments*, where $k = 0, 1, 2, \dots, n$.

▶ Proof

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions



Investigating bound for IDC over integers

Objective

- To find the minimum number of assignments required to compute the mapping sequentially using *in situ* strategy.

Theorem

Let

$$E_n : (x, y) \longrightarrow (F_{n-1}x + F_n y, F_n x + F_{n+1} y)$$

be the mapping over \mathbb{Z}^2 , where F_n is the Fibonacci number.

Then the **mapping E_{4k+2} is computed with $2k+2$ assignments**, where $k = 0, 1, 2, \dots, n$.

▶ Proof

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions

Investigating bound for IDC over integers

Theorem

There exists a linear mapping

$$E : (x, y) \longrightarrow (mx + ny, px + qy)$$

with $m, n, p, q \in \mathbb{Z}$, that cannot be computed by a sequence of at most 6 linear assignments

$$\left. \begin{array}{l} x := ax + by \\ y := cx + dy \\ x := ex + fy \\ y := gx + hy \\ x := ix + jy \\ y := kx + ly \end{array} \right\}$$

where $a, b, c, d, e, f, g, h, i, j, k$, and $l \in \mathbb{Z}$.

General
introduction

Speeding up
computations

Optimizing indexed
communications

Sequential
break down of
operations

Problem description

Contributions

Decentralized
collaborative
editing
system

Problem description

Contributions

Conclusions

Summary

Research directions



Identity (Linking coefficients)

Theorem

Let $E : \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$ be a linear mapping defined as $E(x, y) \rightarrow (mx + ny, px + qy)$ with integral coefficients $m, n, p, q \in \mathbb{Z}$. Let E be computable by a sequence of linear assignments

$$\left. \begin{array}{l} x := a_1x + b_1y \\ y := b'_1x + a'_1y \\ x := a_2x + b_2y \\ y := b'_2x + a'_2y \\ \vdots := \quad \quad \quad \vdots \\ x := a_kx + b_ky \\ y := b'_kx + a'_ky \end{array} \right\}$$

where $a_i, a'_i, b_i, b'_i \in \mathbb{Z}$ are integral coefficients for each $i = 1, \dots, k$. Then the identity

$$mq - pn = a_1a'_1 \dots a_ka'_k \quad \text{holds.} \quad (1)$$

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions



Outline

General introduction

- Speeding up computations
- Optimizing indexed communications

Sequential break down of operations

- Problem description
- Contributions

Decentralized collaborative editing system

- Problem description
- Contributions

Conclusions

- Summary
- Research directions

- 1 General introduction**
 - Speeding up computations
 - Optimizing indexed communications
- 2 Sequential break down of operations**
 - Problem description
 - Contributions
- 3 Decentralized collaborative editing system**
 - Problem description
 - Contributions
- 4 Conclusions**
 - Summary
 - Research directions



Problem description

- To manage conflicting modifications in collaborative editing, indexing method is the fundamental requirement.
- Most recent methods are introduced by

Weiss et al [WUM09]

- Stéphane Weiss, Pascal Urso, and Pascal Molli. Logoot-undo: Distributed collaborative editing system on p2p networks.
IEEE Transactions on Parallel and Distributed Systems, 21:1162–1174, 2010.

Preguiça et al [PMSL09]

- Nuno Preguiça and Joan Manuel Marquês and Marc Shapiro and Mihai Leția. A commutative replicated data type for cooperative editing.
Distributed Computing Systems, International Conference on, 0: 395-403, 2009.

General
introduction

Speeding up
computations

Optimizing indexed
communications

Sequential
break down of
operations

Problem description
Contributions

Decentralized
collaborative
editing
system

Problem description
Contributions

Conclusions

Summary
Research directions



Problem description

- To manage conflicting modifications in collaborative editing, indexing method is the fundamental requirement.
- Most recent methods are introduced by

Weiss et al [WUM09]

- Stéphane Weiss, Pascal Urso, and Pascal Molli. Logoot-undo: Distributed collaborative editing system on p2p networks.
IEEE Transactions on Parallel and Distributed Systems, 21:1162–1174, 2010.

Preguiça et al [PMSL09]

- Nuno Preguiça and Joan Manuel Marquès and Marc Shapiro and Mihai Leția. A commutative replicated data type for cooperative editing.
Distributed Computing Systems, International Conference on, 0: 395-403, 2009.

General
introduction

Speeding up
computations

Optimizing indexed
communications

Sequential
break down of
operations

Problem description
Contributions

Decentralized
collaborative
editing
system

Problem description
Contributions

Conclusions

Summary

Research directions



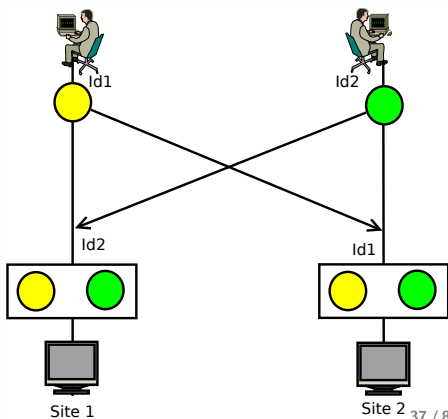
Following situation is not supported

Errors are detected

- These approaches don't support some of the situations.

Identifiers exchange scenario

- 1 Two characters 'a' and 'b' inserted.
- 2 Updates are executed.
- 3 New character lies b/w 'a' and 'b' ?



General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

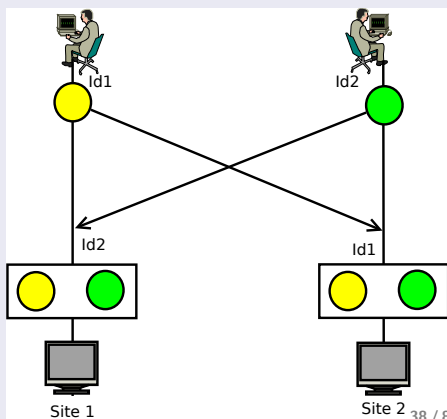
Following situation is not supported

Errors are detected

- These approaches don't support some of the situations.

Identifiers exchange scenario

- 1 Two characters 'a' and 'b' inserted.
- 2 Updates are executed.
- 3 New character lies b/w 'a' and 'b' ?





Approach by Weiss et al

Order Alteration

- General form of identifiers corresponding to two insertions are

$$\langle i, s_1, c_1 \rangle \text{ and } \langle i, s_2, c_2 \rangle$$

$$s_1 < s_2 \text{ or } s_1 = s_2 \text{ and } c_1 < c_2$$

- Let user U_1 intends to insert new character.
- $N = 1$. “*boundary = 1*” then for two identifiers

$$\textcircled{1} p = \langle 0, s, c \rangle$$

with $p < q$ (say)

$$\textcircled{2} q = \langle 0, s, c \rangle,$$



Approach by Weiss et al

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions

Order Alteration

- General form of identifiers corresponding to two insertions are

$$\langle i, s_1, c_1 \rangle \text{ and } \langle i, s_2, c_2 \rangle$$

$$s_1 < s_2 \text{ or } s_1 = s_2 \text{ and } c_1 < c_2$$

- Let user U_1 intends to insert new character.
- $N = 1$. “*boundary = 1*” then for two identifiers

$$\textcircled{1} \quad p = \langle 0, s, c \rangle$$

with $p < q$ (say)

$$\textcircled{2} \quad q = \langle 0, s, c \rangle,$$

Order Alteration

- New identifier computed by the algorithm

$$id = \langle 1, s, c \rangle$$

- Computed identifier should follow

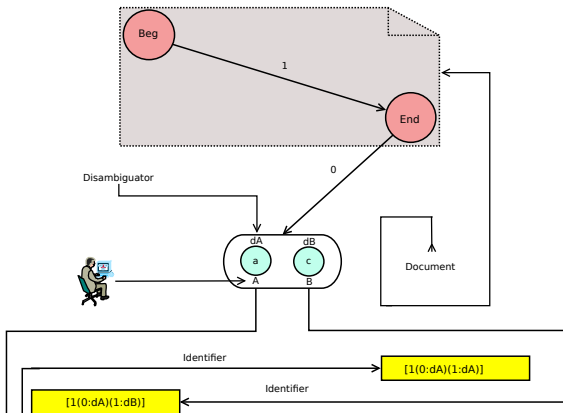
$$p = \langle 0, s, c \rangle < \langle 1, s, c \rangle < \langle 0, s, c \rangle = q$$

$$\because 1 \notin [0, 0]$$

Therefore, new id does not lie b/w identifiers p and q .

Approach by Prejuça et al

Order Alteration



- Two initial identifiers are $(0, d_0)$ and $(1, d_1)$

- $PosID_a = [1(0 : dA)]$ and $PosID_c = [1(0 : dB)]$



Approach by Prejuíça et al

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

- Let user B inserts a new character " a_1 " between " a " and " c " then algorithm returns new identifier as $PosID_{a_1} = [1(0 : dA)(1 : dB)]$

- If $dA = (counter, siteID) = (1, 1)$
and $dB = (counter, siteID) = (1, 2)$

- Then $PosID_a = [1(0 : (1, 1))]$ and $PosID_c = [1(0 : (1, 2))]$
- $PosID_{a_1} = [1(0 : (1, 1))(1 : (1, 2))]$
- Following the definition (Nuno Preguiça et al.)
 $PosID_a < PosID_{a_1}$ and $PosID_c < PosID_{a_1}$

It gives

- $PosID_{a_1} \notin [PosID_a, PosID_c]$.



Approach by Prejuça et al

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

- Let user B inserts a new character " a_1 " between " a " and " c " then algorithm returns new identifier as $PosID_{a_1} = [1(0 : dA)(1 : dB)]$

- If $dA = (counter, siteID) = (1, 1)$
and $dB = (counter, siteID) = (1, 2)$

- Then $PosID_a = [1(0 : (1, 1))]$ and $PosID_c = [1(0 : (1, 2))]$
- $PosID_{a_1} = [1(0 : (1, 1))(1 : (1, 2))]$
- Following the definition (Nuno Preguiça et al.)
 $PosID_a < PosID_{a_1}$ and $PosID_c < PosID_{a_1}$

It gives

- $PosID_{a_1} \notin [PosID_a, PosID_c]$.



Approach by Prejuíça et al

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

- Let user B inserts a new character " a_1 " between " a " and " c " then algorithm returns new identifier as $PosID_{a_1} = [1(0 : dA)(1 : dB)]$

- If $dA = (counter, siteID) = (1, 1)$
and $dB = (counter, siteID) = (1, 2)$

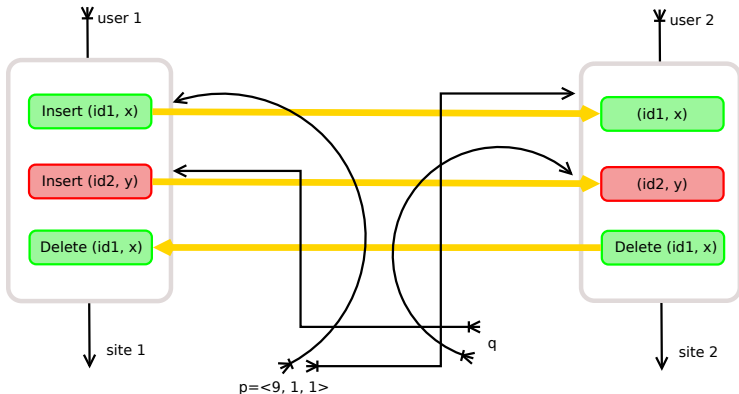
- Then $PosID_a = [1(0 : (1, 1))]$ and $PosID_c = [1(0 : (1, 2))]$
- $PosID_{a_1} = [1(0 : (1, 1))(1 : (1, 2))]$
- Following the definition (Nuno Preguiça et al.)
 $PosID_a < PosID_{a_1}$ and $PosID_c < PosID_{a_1}$

It gives

- $PosID_{a_1} \notin [PosID_a, PosID_c]$.

Redundancy of identifiers

- Let the shared document is with initial and final points $lb = \langle 0, NA, NA \rangle$ and $le = \langle 10, NA, NA \rangle$.
- User U_1 (with $s = 1$ and initial clock $c = 0$) begins



General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Redundancy of identifiers

- It is possible that, $\text{boundary} \geq \text{interval}/N$
- In this case, $\text{step} = \min(\text{interval}/N, \text{boundary}) = \text{interval}/N$

It permits that for $N = 1$, $\text{step} = 9$

- Algorithm returns an identifier $p = \langle 9, 1, 1 \rangle$ such that
- $\{lb, p, le\} = \{\langle 0, NA, NA \rangle, \langle 9, 1, 1 \rangle, \langle 10, NA, NA \rangle\}$

- Let *user 1* inserts another character "y" after "x" and an identifier "q" b/w $p = \langle 9, 1, 1 \rangle, le = \langle 10, NA, NA \rangle$

- Let *user 2* removes character "x", then it is not sure to perform the desired operation.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Redundancy of identifiers

- It is possible that, $\text{boundary} \geq \text{interval}/N$
- In this case, $\text{step} = \min(\text{interval}/N, \text{boundary}) = \text{interval}/N$

It permits that for $N = 1$, $\text{step} = 9$

- Algorithm returns an identifier $p = \langle 9, 1, 1 \rangle$ such that
- $\{lb, p, le\} = \{\langle 0, NA, NA \rangle, \langle 9, 1, 1 \rangle, \langle 10, NA, NA \rangle\}$

- Let *user 1* inserts another character "y" after "x" and an identifier "q" b/w $p = \langle 9, 1, 1 \rangle, le = \langle 10, NA, NA \rangle$

- Let *user 2* removes character "x", then it is not sure to perform the desired operation.

General introduction

Speeding up computations
Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Redundancy of identifiers

- It is possible that, $\text{boundary} \geq \text{interval}/N$
- In this case, $\text{step} = \min(\text{interval}/N, \text{boundary}) = \text{interval}/N$

It permits that for $N = 1$, $\text{step} = 9$

- Algorithm returns an identifier $p = \langle 9, 1, 1 \rangle$ such that
- $\{lb, p, le\} = \{\langle 0, NA, NA \rangle, \langle 9, 1, 1 \rangle, \langle 10, NA, NA \rangle\}$

- Let *user 1* inserts another character "y" after "x" and an identifier "q" b/w $p = \langle 9, 1, 1 \rangle, le = \langle 10, NA, NA \rangle$

- Let *user 2* removes character "x", then it is not sure to perform the desired operation.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Redundancy of identifiers

- It is possible that, $\text{boundary} \geq \text{interval}/N$
- In this case, $\text{step} = \min(\text{interval}/N, \text{boundary}) = \text{interval}/N$

It permits that for $N = 1$, $\text{step} = 9$

- Algorithm returns an identifier $p = \langle 9, 1, 1 \rangle$ such that
- $\{lb, p, le\} = \{\langle 0, NA, NA \rangle, \langle 9, 1, 1 \rangle, \langle 10, NA, NA \rangle\}$

- Let *user 1* inserts another character "y" after "x" and an identifier "q" b/w $p = \langle 9, 1, 1 \rangle$, $le = \langle 10, NA, NA \rangle$

- Let *user 2* removes character "x", then **it is not sure to perform the desired operation.**

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary

Research directions

Errors.....

Infinite Iteration

1... For two identifiers with same values of i , Algorithm does not generate any new identifier due to the reason that $interval = -1$.

2... Proposed formula does not work in certain cases

Example

$p = \langle k_1, 4, 7 \rangle \langle k_2, 9, 5 \rangle$ and $q = \langle k_3, 5, 3 \rangle \langle k_4, 3, 6 \rangle \langle k_5, 3, 9 \rangle$
with assumption $BASE = 100$, $boundary = 10$ and

$$k_1 = k_2 = k_3 = k_4 = k_5 = k$$

To insert N lines between two lines identified by p and q ,

$$\text{prefix}(q, 1) - \text{prefix}(p, 1) - 1 = \begin{cases} -1 & \text{if } k_3 = k_1 \\ 0 & \text{if } k_3 = k_1 + 1 \end{cases}$$

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

Errors.....

Infinite Iteration

1... For two identifiers with same values of i , Algorithm does not generate any new identifier due to the reason that $interval = -1$.

2... Proposed formula does not work in certain cases

Example

$p = \langle k_1, 4, 7 \rangle \langle k_2, 9, 5 \rangle$ and $q = \langle k_3, 5, 3 \rangle \langle k_4, 3, 6 \rangle \langle k_5, 3, 9 \rangle$
with assumption $BASE = 100$, $boundary = 10$ and

$$k_1 = k_2 = k_3 = k_4 = k_5 = k$$

To insert N lines between two lines identified by p and q ,

$$\text{prefix}(q, 1) - \text{prefix}(p, 1) - 1 = \begin{cases} -1 & \text{if } k_3 = k_1 \\ 0 & \text{if } k_3 = k_1 + 1 \end{cases}$$

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

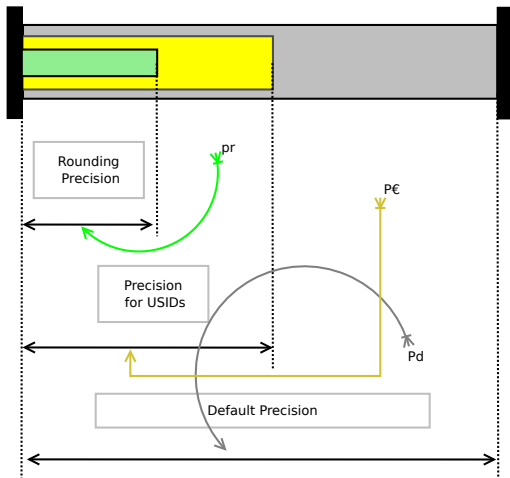
Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

Precision control technique



- **Basic principle $p_r < p_\epsilon < p_d$**

General introduction

Speeding up computations
Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



User/site identifiers (*USIDs*)

Pattern

- $USID \rightarrow 0.00 \dots 0_{p_r+1} d_1 d_2 \dots d_{p_r}$
- where $p_r \geq 1$, $p_\epsilon \geq p_r + \mu$, $\mu \geq 2$, $\mu \in \mathbb{N}$

Example

0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
-------	-------	-------	-------	-------	-------	-------	-------	-------

- Keeping this pattern, we are able to generate 9×10^w different epsilons with $w = p_r - 1$. For $p_r \geq 1$
- The **minimum** value of epsilon is $1 \times 10^{-(p_r+2)}$
- The **maximum** value is $0.00 \dots 0_{p_r+1} 99 \dots 9_{p_r}$
- Property $|USID|_{p_r} = 0$

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



User/site identifiers (*USIDs*)

Pattern

- $USID \rightarrow 0.00 \dots 0_{p_r+1} d_1 d_2 \dots d_{p_r}$
- where $p_r \geq 1$, $p_\epsilon \geq p_r + \mu$, $\mu \geq 2$, $\mu \in \mathbb{N}$

Example

0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
-------	-------	-------	-------	-------	-------	-------	-------	-------

- Keeping this pattern, we are able to generate 9×10^w different epsilons with $w = p_r - 1$. For $p_r \geq 1$
- The **minimum** value of epsilon is $1 \times 10^{-(p_r+2)}$
- The **maximum** value is $0.00 \dots 0_{p_r+1} 99 \dots 9_{p_r}$
- Property $|USID|_{p_r} = 0$

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



User/site identifiers (*USIDs*)

Pattern

- $USID \rightarrow 0.00 \dots 0_{p_r+1} d_1 d_2 \dots d_{p_r}$
- where $p_r \geq 1$, $p_\epsilon \geq p_r + \mu$, $\mu \geq 2$, $\mu \in \mathbb{N}$

Example

0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
-------	-------	-------	-------	-------	-------	-------	-------	-------

- Keeping this pattern, we are able to generate 9×10^w different epsilons with $w = p_r - 1$. For $p_r \geq 1$
- The **minimum** value of epsilon is $1 \times 10^{-(p_r+2)}$
- The **maximum** value is $0.00 \dots 0_{p_r+1} 99 \dots 9_{p_r}$
- Property $|USID|_{p_r} = 0$

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Generating character/line identifiers *LCIDs*

Require

- 1 Points computed for one user must be different from points computed for others
- 2 Set of points computed for each user must be an ordered set
- 3 It should be possible to compute cardinality for the set of points.

- We modified the classical midpoint formula $(a + b)/2$, for an interval $I = [a, b]$ with $0 \leq a < b$ such that $\forall x, y \in I$

$$f(x, y) = x + \frac{y - x}{2} - \epsilon \quad \text{with } x < y \quad (2)$$



Generating character/line identifiers *LCIDs*

Require

- 1 Points computed for one user must be different from points computed for others
- 2 Set of points computed for each user must be an ordered set
- 3 It should be possible to compute cardinality for the set of points.

- We modified the **classical midpoint formula** $(a + b)/2$, for an interval $I = [a, b]$ with $0 \leq a < b$ such that $\forall x, y \in I$

$$f(x, y) = x + \frac{y - x}{2} - \epsilon \quad \text{with } x < y \quad (2)$$



Algorithm outlines

- Assume that an interval $I = [a, b]$, $0 \leq a < b$, presents **shared document** or a and b are two *LCIDs* corresponding to two characters/lines.
- To compute new *LCIDs*, we proceed as follows:

Outlines

- 1 Round a and b over the precision p_r
- 2 Evaluate $\nabla = \frac{|b|_{p_r} - |a|_{p_r}}{2}$ and round it over p_r
- 3 If $\nabla > USID_{max}$ then go to the next step
- 4 $p := \left| |a|_{p_r} + \nabla \right|_{p_r} - USID$
- 5 Verify that $p \not\leq a$ and $p \not\leq b$.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions



Algorithm outlines

- Assume that an interval $I = [a, b]$, $0 \leq a < b$, presents **shared document** or a and b are two *LCIDs* corresponding to two characters/lines.
- To compute new *LCIDs*, we proceed as follows:

Outlines

- 1 Round a and b over the precision p_r
- 2 Evaluate $\nabla = \frac{|b|_{p_r} - |a|_{p_r}}{2}$ and round it over p_r
- 3 If $\nabla > USID_{max}$ then go to the next step
- 4 $p := \left| |a|_{p_r} + \nabla \right|_{p_r} - USID$
- 5 Verify that $p \not\leq a$ and $p \not\leq b$.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary

Research directions



Uniqueness of *LCIDs*

Theorem

*Let two participants be assigned with two different USIDs (ϵ_i , ϵ_j such that $\epsilon_i < \epsilon_j$, for $0 < i < j$). Then for two operations performed by two participants respectively, two different *LCIDs* will be generated (over the interval $I = [a, b]$ with $0 \leq a < b$).*

Example

<i>User</i>	<i>Unique Identifiers</i>									
U_1	0.099	0.199	0.299	0.399	0.499	0.599	0.699	0.799	0.899	0.999
U_2	0.098	0.198	0.298	0.398	0.498	0.598	0.698	0.798	0.898	0.998
U_3	0.097	0.197	0.297	0.397	0.497	0.597	0.697	0.797	0.897	0.997
U_4	0.096	0.196	0.296	0.396	0.496	0.596	0.696	0.796	0.896	0.996
U_5	0.095	0.195	0.295	0.395	0.495	0.595	0.695	0.795	0.895	0.995
U_6	0.094	0.194	0.294	0.394	0.494	0.594	0.694	0.794	0.894	0.994

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

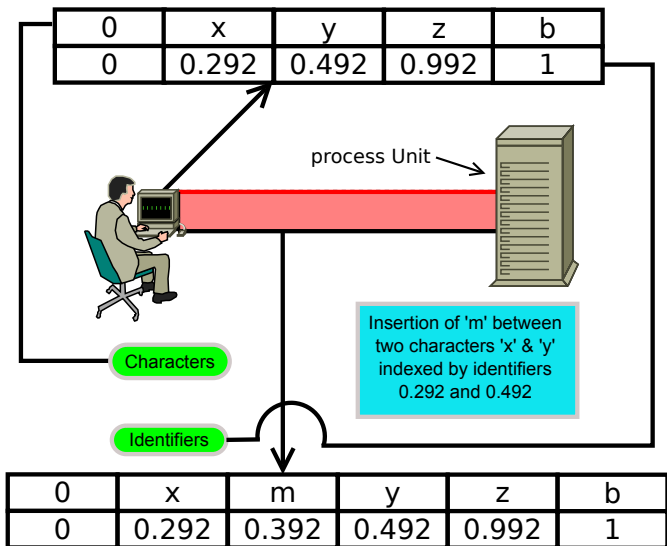
Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

Inserting single character



General introduction

Speeding up computations
Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

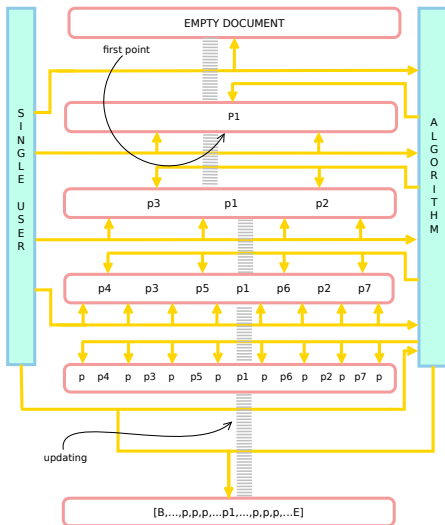
Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

Explanation



General introduction

- Speeding up computations
- Optimizing indexed communications

Sequential break down of operations

- Problem description
- Contributions

Decentralized collaborative editing system

- Problem description
- Contributions

Conclusions

- Summary
- Research directions

Cardinality

- Let $I = [a, b]$, $0 \leq a < b$ and $\gamma = |b - a|$ with $\gamma > 0$.
- Then for the precision p_r , local cardinality that a user can attain is computed as

Formula

-

$$C_I = \begin{cases} \gamma \times 10^{p_r} & \text{if } b > a \geq 0, \text{ for } a, b \in \mathbb{N} \\ (\gamma_1 \times 10^{p_r}) - 1 & \text{if } b > a > 0, \text{ for } a, b \in \mathbb{R} \\ & \text{and } \gamma_1 = |\gamma|_{p_r} \end{cases}$$

such that

$$p_r < p_\epsilon < p_d$$

General
introduction

Speeding up
computations

Optimizing indexed
communications

Sequential
break down of
operations

Problem description
Contributions

Decentralized
collaborative
editing
system

Problem description
Contributions

Conclusions

Summary
Research directions

Cardinality

- Let $I = [a, b]$, $0 \leq a < b$ and $\gamma = |b - a|$ with $\gamma > 0$.
- Then for the precision p_r , local cardinality that a user can attain is computed as

Formula

-

$$C_I = \begin{cases} \gamma \times 10^{p_r} & \text{if } b > a \geq 0, \text{ for } a, b \in \mathbb{N} \\ (\gamma_1 \times 10^{p_r}) - 1 & \text{if } b > a > 0, \text{ for } a, b \in \mathbb{R} \\ & \text{and } \gamma_1 = |\gamma|_{p_r} \end{cases}$$

such that

$$p_r < p_\epsilon < p_d$$

General
introduction

Speeding up
computations

Optimizing indexed
communications

Sequential
break down of
operations

Problem description
Contributions

Decentralized
collaborative
editing
system

Problem description
Contributions

Conclusions

Summary
Research directions



General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions

Example

- Suppose that $I_1 = [0, 1]$, $p_r = 3$, $USID_{min} = 0.00001$, and $USID_{max} = 0.0000999$.
- The local cardinality of a set of *LCIDs*, that single user with $USID = 0.0000439$ can attain, is computed by $C_l = 1 * 10^3 = 1000$.

Example

- If the same user wants to compute local cardinality of *LCIDs* between two *LCIDs* (already computed by him).
- For instance, let $I_2 = [0.7609561, 0.8139561]$ then $\gamma_1 = 0.053$ and $C_l = (0.053 * 10^3) - 1 = 52$.



General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions

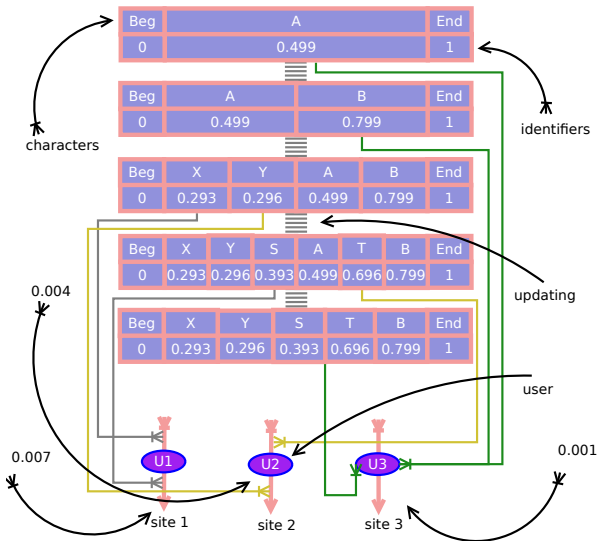
Example

- Suppose that $I_1 = [0, 1]$, $p_r = 3$, $USID_{min} = 0.00001$, and $USID_{max} = 0.0000999$.
- The local cardinality of a set of *LCIDs*, that single user with $USID = 0.0000439$ can attain, is computed by $C_l = 1 * 10^3 = 1000$.

Example

- If the same user wants to compute local cardinality of *LCIDs* between two *LCIDs* (already computed by him).
- For instance, let $I_2 = [0.7609561, 0.8139561]$ then $\gamma_1 = 0.053$ and $C_l = (0.053 * 10^3) - 1 = 52$.

Assuring order preservation



General introduction

- Speeding up computations
- Optimizing indexed communications

Sequential break down of operations

- Problem description
- Contributions

Decentralized collaborative editing system

- Problem description
- Contributions

Conclusions

- Summary
- Research directions



Some properties

Bijection property

- Let S' be the set defined over the interval $I = [a, b], 0 \leq a < b$, as

$$S' = \left\{ \frac{n'}{10^{p_r}} \mid n' \in \mathbb{N}, a < n' \leq \beta, \beta = \gamma * 10^{p_r} \right\}$$

where γ is the length of the interval I with $|\gamma|_{p_r} > 0$.

- For $1 \leq k \leq n, n \in \mathbb{N}$, let ϵ_k be one of the *USIDs* then the function $f_k : S' \rightarrow S''$ defined by $f_k(x') = x' - \epsilon_k, \forall x' \in S'$, is a bijection.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

Some properties

Bijection property

- Let S' be the set defined over the interval $I = [a, b]$, $0 \leq a < b$, as

$$S' = \left\{ \frac{n'}{10^{p_r}} \mid n' \in \mathbb{N}, a < n' \leq \beta, \beta = \gamma * 10^{p_r} \right\}$$

where γ is the length of the interval I with $|\gamma|_{p_r} > 0$.

- For $1 \leq k \leq n$, $n \in \mathbb{N}$, let ϵ_k be one of the *USIDs* then the function $f_k : S' \rightarrow S''$ defined by $f_k(x') = x' - \epsilon_k, \forall x' \in S'$, is a bijection.



Some properties

General introduction

- Speeding up computations
- Optimizing indexed communications

Sequential break down of operations

- Problem description
- Contributions

Decentralized collaborative editing system

- Problem description
- Contributions

Conclusions

- Summary
- Research directions

One of the closure properties

- Let S_i , S_j and S_k be the set of *LCIDs* generated for users U_i , U_j and U_k with *USIDs*, ϵ_i , ϵ_j and ϵ_k over the interval $I = [a, b]$, $0 \leq a < b$ respectively.

- Then for any two points $x \in S_i$, $y \in S_j$, $x < y$ and for $\epsilon_j < \epsilon_k$.

$$\left| x \right|_{p_r} + \left| \frac{|y|_{p_r} - |x|_{p_r}}{2} \right|_{p_r} - \epsilon_k \in S_k$$

Some properties

One of the closure properties

- Let S_i , S_j and S_k be the set of *LCIDs* generated for users U_i , U_j and U_k with *USIDs*, ϵ_i , ϵ_j and ϵ_k over the interval $I = [a, b]$, $0 \leq a < b$ respectively.

- Then for any two points $x \in S_i$, $y \in S_j$, $x < y$ and for $\epsilon_j < \epsilon_k$.

$$\left| |x|_{p_r} + \left| \frac{|y|_{p_r} - |x|_{p_r}}{2} \right|_{p_r} \right|_{p_r} - \epsilon_k \in S_k$$

Analysis of the function

- 1 Under rounding conditions.
 - 2 Under floating point arithmetic.
 - 3 Without rounding conditions,
- Proposed function works nicely.

l	p_e	ϵ_j	ϵ	$Cd(f_c)$	$Cd(f_1)$	$Cd(f_1) - Cd(f_c)$
$[0, 0.01]$	3	0.003	0.003	3	33	30
$[0, 0.01]$	4	0.0003	0.0003	33	243	210
$[0, 0.01]$	5	0.00003	0.00003	257	2216	1959
$[0, 0.01]$	6	0.000003	0.000003	2049	19380	17331
$[0, 0.0001]$	7	0.0000003	0.0000003	257	2216	1959
$[0, 0.0001]$	8	0.00000003	0.00000003	2049	19580	17371
$[0, 0.0001]$	9	0.000000003	0.000000003	32769	165039	132270



Exchange of points

Delayed exchange of points

- Users modify/edit shared document **one by one** and sends updates to all other users.
- It requires to verify the validity of the algorithm for the points **including the points that user has received.**

- Under this environment, we studied two cases.
 - 1 Random participation
 - 2 Ordered participation

Effect on global cardinality

- This environment may prevent global cardinality in approaching to its upper bound.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Exchange of points

Delayed exchange of points

- Users modify/edit shared document **one by one** and sends updates to all other users.
- It requires to verify the validity of the algorithm for the points **including the points that user has received.**

- Under this environment, we studied two cases.
 - 1 Random participation
 - 2 Ordered participation

Effect on global cardinality

- This environment may prevent global cardinality in approaching to its upper bound.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Exchange of points

Delayed exchange of points

- Users modify/edit shared document **one by one** and sends updates to all other users.
- It requires to verify the validity of the algorithm for the points **including the points that user has received**.

- Under this environment, we studied two cases.
 - ① Random participation
 - ② Ordered participation

Effect on global cardinality

- This environment may prevent global cardinality in approaching to its upper bound.

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Exchange of points

General introduction

Speeding up computations
Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

Real time exchange of points

- Contrary to the delayed exchange environment, we propose this environment.
- Under this environment, **the identifiers are not required to compute on the basis of remote points** and all participants are supposed to be active.

Two situations

- Under this environment, we studied two cases.
 - 1 Random participation
 - 2 Ordered participation



Exchange of points

General introduction

Speeding up computations
Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions

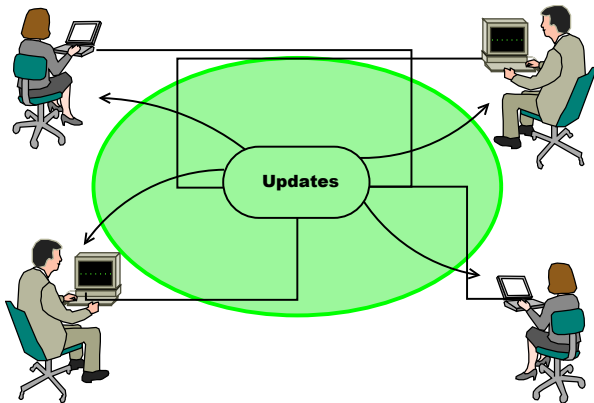
Real time exchange of points

- Contrary to the delayed exchange environment, we propose this environment.
- Under this environment, **the identifiers are not required to compute on the basis of remote points** and all participants are supposed to be active.

Two situations

- Under this environment, we studied two cases.
 - 1 Random participation
 - 2 Ordered participation

Model (Maximizing global cardinality)



Effect on global cardinality

- This environment makes it possible that **global cardinality could reach its upper bound.**

General introduction

Speeding up computations
Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Outline

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions

- 1 **General introduction**
 - Speeding up computations
 - Optimizing indexed communications
- 2 **Sequential break down of operations**
 - Problem description
 - Contributions
- 3 **Decentralized collaborative editing system**
 - Problem description
 - Contributions
- 4 **Conclusions**
 - Summary
 - Research directions



Further work

We contributed in advancing the capabilities of

- 1 Sequential computation and
- 2 Distributed systems

But still a lot of work is to be done both in

- 1 Sequential break-down of operations
- 2 Decentralized collaborative editing system

Dispersion problem

- The removal of point(s) creates the possibility of insertion of point(s).

Each removal does not assure

- The insertion of new point(s).

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Further work

We contributed in advancing the capabilities of

- 1 Sequential computation and
- 2 Distributed systems

But still a lot of work is to be done both in

- 1 Sequential break-down of operations
- 2 Decentralized collaborative editing system

Dispersion problem

- The removal of point(s) creates the possibility of insertion of point(s).

Each removal does not assure

- The insertion of new point(s).

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Further work

We contributed in advancing the capabilities of

- 1 Sequential computation and
- 2 Distributed systems

But still a lot of work is to be done both in

- 1 Sequential break-down of operations
- 2 Decentralized collaborative editing system

Dispersion problem

- The removal of point(s) creates the possibility of insertion of point(s).

Each removal does not assure

- The insertion of new point(s).

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



Further work

We contributed in advancing the capabilities of

- 1 Sequential computation and
- 2 Distributed systems

But still a lot of work is to be done both in

- 1 Sequential break-down of operations
- 2 Decentralized collaborative editing system

Dispersion problem

- The removal of point(s) creates the possibility of insertion of point(s).

Each removal does not assure

- The insertion of new point(s).

General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description
Contributions

Decentralized collaborative editing system

Problem description
Contributions

Conclusions

Summary
Research directions



General introduction

- Speeding up computations
- Optimizing indexed communications

Sequential break down of operations

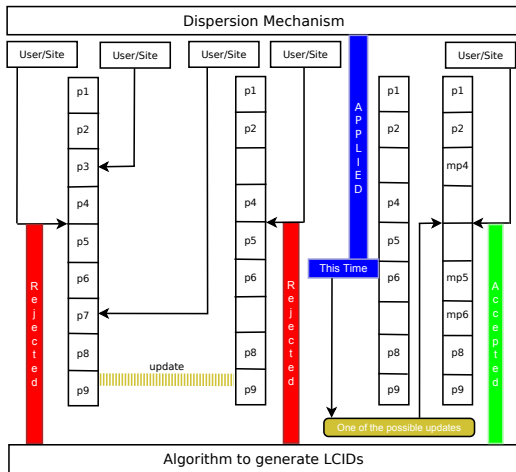
- Problem description
- Contributions

Decentralized collaborative editing system

- Problem description
- Contributions

Conclusions

- Summary
- Research directions





Modification/extension for *IDC*

Limitations

- Combinatorial results and counter examples show that the *in situ* strategies have certain limitations.

Modifications

- Modification of the idea is required so that the investigated draw backs could be removed.

Extensions

- Extension of the idea is required to deal with general cases and the real implementation is required.

General
introduction

Speeding up
computations

Optimizing indexed
communications

Sequential
break down of
operations

Problem description
Contributions

Decentralized
collaborative
editing
system

Problem description
Contributions

Conclusions

Summary

Research directions



General introduction

Speeding up computations

Optimizing indexed communications

Sequential break down of operations

Problem description

Contributions

Decentralized collaborative editing system

Problem description

Contributions

Conclusions

Summary

Research directions

Questions?

References



Serge Burckel.

The parallel-sequential duality : Matrices and graphs.
CoRR, abs/0709.4397, 2007.



Nuno Preguiça, Joan Manuel Marquès, Marc Shapiro, and Mihai Leția.

A commutative replicated data type for cooperative editing.

Distributed Computing Systems, International Conference on, 0:395–403, 2009.



Stéphane Weiss, Pascal Urso, and Pascal Molli.

Logoot: A scalable optimistic replication algorithm for collaborative editing on p2p networks.

In *ICDCS*, pages 404–412. IEEE Computer Society, 2009.