Keynote: Large-scale trustworthy distributed collaborative systems - challenges and prospective solutions
Claudia-Lavinia Ignat

To cite this version:
Claudia-Lavinia Ignat. Keynote: Large-scale trustworthy distributed collaborative systems - challenges and prospective solutions. INFORSID (INFormatique des ORganisations et Systèmes d’Information et de Décision) 2024, May 2024, Nancy, France. hal-04595294

HAL Id: hal-04595294
https://inria.hal.science/hal-04595294
Submitted on 31 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License
Large-scale trustworthy distributed collaborative systems: challenges and prospective solutions

Claudia-Lavinia Ignat
COAST - Centre Inria de l’Université de Lorraine

Inforsid 2024
May 30, 2024
claudia.ignat@inria.fr
Douglas Engelbart: Augmenting Human Intellect

The Mother of all Demos, December 9, 1968

NLS: Online System
https://archive.org/details/dougengelbartarchives
Groupware, early 1990s

«Computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment.» [EGR91]

- Lotus Notes (HCL Notes today), one of the first commercial groupware allowing remote group collaboration
Groupware Time Space Matrix [J88]

- **Face to face interactions**
  - decision rooms, single display groupware, **shared table**, wall displays, roomware, …

- **Continuous task**
  - large public display, team rooms, shift work groupware, project management, …

- **Remote interactions**
  - video conferencing, instance messaging, **chats**/MUDs/virtual worlds, shared screens, **multi-user editors**, …

- **Communication + coordination**
  - email, bulletin boards, blogs, asynchronous conferencing, group calendars, workflow, **version control**, **wikis**, …

**Time/Space Groupware Matrix**

- **same time**
  - synchronous

- **different time**
  - asynchronous

- **same place**
  - co-located

- **different place**
  - remote
Collaborative Systems: from users to community of users

“Isn’t it chaotic to all edit in the same document, even the same paragraph, at the same time?”

“Why would a group ever want to edit in the same line of text at the same time?” [EGR91]
From users to community of users: new practices
From users to community of users: new practices

Example: Intergovernmental Panel Climate Change (IPCC) assessment report editing

- Expert contributions: hundreds of scientists from different countries
- Review stages by experts and governments
- Approval sessions: government delegates and lead authors
- Editing teams: review editors
Limitations of Central Authority Systems
Vision: Peer-to-Peer Collaborative Systems
Collaboration Modes – Concurrent Changes
Collaboration Modes – Offline Work
Collaboration Modes – Ad-hoc Collaboration
Research issues

1. How to maintain consistency of different copies in the face of concurrent modifications?

2. How to secure collaboration data?

3. How to make a collaboration trustworthy?
Research issues

1. How to **maintain consistency of different copies** in the face of concurrent modifications?

2. How to **secure collaboration data**?

3. How to make a collaboration **trustworthy**?
CAP Theorem [B00]

- Properties of shared-data systems
  - Data consistency
  - System availability
  - Tolerance to network partition
- Only two out of three properties can be achieved at a given time

- Relax consistency for availability in the presence of network partitions
Operational transformation (OT) \[EG89\]

- Transforms non commuting operations to make them commute
- Genericity
- Time complexity
  Average: $O(H \cdot c)$
  Worst case: $O(H^2)$
  \[H: \#\text{ops} \quad c: \text{avg. \#conc. ops}\]
- Difficult to write correct transformation functions
- State vectors used for detecting concurrency ⇒ scalability limitations
- Not very suitable for large scale peer-to-peer collaboration

\[
T(\text{ins}(p_1,c_1), \text{ins}(p_2,c_2)) \leftarrow \\
\begin{cases} 
\text{ins}(p_1,c_1) & \text{if } (p_1 < p_2) \\
\text{ins}(p_1+1,c_1) & \text{else}
\end{cases}
\]
Conflict-free Replicated Data Types (CRDT) [SPBZ11]

- Design operations to be commutative by construction
- Abstract data types
  - Designed to be replicated at multiple sites
  - Any replica can be modified without coordination
  - State convergence is guaranteed
- State-based and operation-based approaches
Conflict-free Replicated Data Types (CRDT)
State-based Replication

- Algorithm
  - Periodically, replica at $p_i$ sends its current state to $p_j$
  - Replica $p_j$ merges received state into its local state by executing $m$
  - After receiving all updates (irrespective of order), each replica will have same state
Conflict-free Replicated Data Types (CRDT) 
State-based Replication

• Merge operator:
  • **Commutative**: \( x \odot y = y \odot x \)
  • **Associative**: \( (x \odot y) \odot z = x \odot (y \odot z) \)
  • **Idempotent**: \( x \odot x = x \)

• A semi-lattice is a Partial order \( \leq \) set \( S \) with a least upper bound (LUB), denoted \( \sqcup \)
  • \( m = x \sqcup y \) is a LUB of \( \{ x, y \} \) under \( \leq \) if and only if
    \[ \forall m', x \leq m' \land y \leq m' \Rightarrow x \leq m \land y \leq m \land m \leq m' \]
  • It follows that \( \sqcup \) is commutative, associative and idempotent
Conflict-free Replicated Data Types (CRDT) Operation-based Replication

- An update split into \((t, u)\): \(t\) is a side-effect-free prepare-update method and \(u\) is an effect-update method

- Algorithm
  - Updates delivered to all replicas
  - Causally-ordered broadcast, every message delivered to every node exactly once w.r.t. happen-before order

- Commutativity holds for concurrent updates
Conflict-free Replicated Data Types (CRDT) Operation vs. State-based Replication

• Both approaches are equivalent
  • A state-based object can emulate an operation-based object, and vice-versa

• Operation-based:
  • More efficient since you only ship small updates
  • But require exactly once causally-ordered broadcast

• State-based:
  • Only require reliable broadcast
  • Communication overhead of shipping the whole state

• Delta State-based [ASB18]:
  • Small messages
  • Dissemination over unreliable communication channels
Consistency Maintenance
Conflict-free Replicated Data Types (CRDT)

- Register
  - Last-Writer Wins
- Multi-Value
- Set
  - Grow-Only
  - 2-Phase
  - Observed-Remove
  - Observed-Update-Remove

- Map
- Counter
- Graph
  - Directed
  - Monotonic DAG
  - Edit graph
- Sequence
Conflict-free Replicated Data Types (CRDT) (Text) Sequence [PMSL09][WUM09]

• Document = linear sequence of elements
  • Each element has a unique identifier
  • Identifier constant for the lifetime of the document
  • Dense total order of identifiers consistent with element order:
    • \( \forall id_x, id_y: id_x < id_y \Rightarrow \exists id_z: id_x < id_z < id_y \)

• Different approaches for generating identifiers:
  • TreeDoc, Logoot, LogootSplit, …
Conflict-free Replicated Data Types (CRDT)

Logoot [*WUM09*]

- Logoot identifiers: \(<p_1, s_1, h_1>, <p_2, s_2, h_2>, \ldots, <p_k, s_k, h_k>\)
  
  - \(p_i\) integer
  - \(s_i\) site identifier
  - \(h_i\) logical clock at site \(s_i\)

<table>
<thead>
<tr>
<th>(&lt;1,2,1&gt;)</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;1,2,2&gt;)</td>
<td>o</td>
</tr>
<tr>
<td>(&lt;2,1,2&gt;)</td>
<td>n</td>
</tr>
<tr>
<td>(&lt;3,1,3&gt;)</td>
<td>c</td>
</tr>
<tr>
<td>(&lt;3,1,3&gt;&lt;8,4,5&gt;)</td>
<td>u</td>
</tr>
<tr>
<td>(&lt;3,2,5&gt;)</td>
<td>r</td>
</tr>
<tr>
<td>(&lt;4,1,7&gt;)</td>
<td>e</td>
</tr>
<tr>
<td>(&lt;4,1,7&gt;&lt;9,2,6&gt;)</td>
<td>n</td>
</tr>
<tr>
<td>(&lt;7,2,8&gt;)</td>
<td>c</td>
</tr>
<tr>
<td>(&lt;9,1,7&gt;)</td>
<td>y</td>
</tr>
<tr>
<td>(&lt;10,2,8&gt;)</td>
<td></td>
</tr>
<tr>
<td>(&lt;12,3,1&gt;)</td>
<td>c</td>
</tr>
<tr>
<td>(&lt;12,3,1&gt;&lt;6,5,1&gt;)</td>
<td>o</td>
</tr>
<tr>
<td>(&lt;12,3,1&gt;&lt;7,8,2&gt;)</td>
<td>n</td>
</tr>
<tr>
<td>(&lt;12,3,1&gt;&lt;7,8,2&gt;&lt;12,3,5&gt;)</td>
<td>t</td>
</tr>
<tr>
<td>(&lt;12,3,1&gt;&lt;7,8,2&gt;&lt;13,3,6&gt;)</td>
<td>r</td>
</tr>
<tr>
<td>(&lt;12,3,1&gt;&lt;7,8,2&gt;&lt;14,3,7&gt;)</td>
<td>l</td>
</tr>
</tbody>
</table>

- **Time complexity**
  - Average: \(O(k \log(n))\)
  - Worst case: \(O(H \cdot \log(H))\)
  
  - \(H\): #ops
  - \(n\): doc. size (non deleted chars.)
  - \(k\): avg. size of Logoot identifier

- **No need for concurrency detection**
- **Identifiers storage cost**
- **New design for each data type**
- **Suitable for large-scale collaboration**
Conflict-free Replicated Data Types (CRDT)

LogootSplit [AMOI13]

**Aggregation of elements** to limit metadata

LogootSplit identifiers

<table>
<thead>
<tr>
<th>p₁</th>
<th>…</th>
<th>pₙ</th>
<th>site_id</th>
<th>clock</th>
<th>begin</th>
<th>end</th>
</tr>
</thead>
</table>

**Base**

**Interval**

| 1,1,[0,16] | concurrency contrl |
| 1,1,[0,5] | concur |
| 1,1,5,2,1,[0,0] | r |
| 1,1, [6,16] | ency contrl |

Insert **r** between “concur” and “ency contrl”

| 1,1,[0.5] | concur |
| 1,1,5,2,1,[0,0] | r |
| 1,1, [6,15] | ency contrl |
| 1,1,15,3,1,[0,0] | o |
| 1,1,[16,16] | l |

Insert **o** between “ency contr” and “l”
Conflict-free Replicated Data Types (CRDT) LogootSplit - Impact

- Implemented in MUTE P2P collaborative editor [NEOIC17] [github.com/coast-team/mute, OpenPaaS::NG, ADT MUTE]
- Better performances than GoogleDocs [DI16a]

- Related approach on wikis transferred to XWiki [IAO21]
User Study: The effect of delay on users [IOFSC15]

- Delays in seeing modifications of other users
  - Network delay
  - Time complexity of consistency maintenance algorithms
  - Types of architecture

- How does delay influence group performance?
Experiment design

• 20 groups of 4 students
  • Perform several collaborative editing tasks
    • A proofreading task
    • A sorting task
    • A note taking task
  • Use the provided collaborative editor (Etherpad) + chat
  • Each group experienced a certain delay (0, 4, 6, 8, 10 s)
• Registration of user keyboard inputs
• Video recording of user activities on desktop
Note-taking [IOFSC15]

The participants ranged in age from 21 – 27. All participants used French in their daily activities. An electronic announcement solicited participation. One of the researchers organized interested participants into sets of 4 and scheduled the session. All participants received a 10 Euro gift certificate for their participation.

2.2 Apparatus

The experiment was conducted using four GNU/Linux desktop computers in a classroom setting. Participants were separated by partitions and could not directly observe other team members while they worked, although typing activity was audible. The server running the Etherpad application was hosted on an Amazon Elastic Compute Cloud (EC2) instance located in the US East (Northern Virginia) Region. Each desktop ran the Mozilla Firefox web browser executing the Etherpad web client application. Etherpad hosted the task stimuli and a Chat dialogue facility (see Figure 2). User operations appeared color-coded in both the text and chat. Etherpad relies on a client-server architecture where each client/user edits a copy of the shared document. When a user performed a modification it was immediately displayed on the local copy of the document and then sent to the server. The server merged the change received from the user with other user changes and then transmitted the updates to the other users. When a user edited a sequence of characters, the first change on the character was immediately sent to the server, while the other changes were sent at once only upon reception of an acknowledgement from the server. With each change sent to the server, it created a new version of the document. Gstreamer software enabled the video recording of user activity. We also instrumented Etherpad to register all user keyboard inputs on the client side and to introduce delays on the server-side. The editor window displayed 50 lines of text. Users editing above the field of view of a collaborator could cause the lines within the collaborator's view to “jump” inexplicably. Such a property is consistent with...
Delay reduces Group Performance

- Delay increases error rate and redundancy
Delay reduces Group Performance

- Delay decreases proportion of keywords
Design implications

• Reduce the delay by the choice of the architecture and synchronisation algorithms

• Make users aware of existing delays such that they can compensate for the delay by coordination strategies

• Analyse real collaboration traces to understand collaboration patterns and behavior [NI18]
Perspectives – Replication for complex data

• Composition of replication mechanisms for simple data
  • Maintaining global invariants (e.g. integrity constraints) over CRDTs for replicated relational databases [IEB24]
  • Replication mechanisms for P2P cloud storage [AII23]
  • Automated approach to combine CRDTs meeting expressed constraints
• Validate with users the defined merge semantics
Research issues

1. How to maintain consistency of different copies in the face of concurrent modifications?

2. How to secure collaboration data?

3. How to make a collaboration trustworthy?
Security in peer-to-peer collaboration

- Client-side E2E encryption
- Group key management
- Access control without central authority
Encryption for mutable data by means of CRDTs

- CRDTs suitable for client-side E2E encryption
Group key management for large and dynamic groups

- TreeKEM - Message Layer Security (MLS) RFC 9420
  - Operations: Add, Remove and Update

  - Delivery Service: a central server that resolves conflicts among operations

- Goal: Distribute the Delivery Service [PIFTI23]
  - Use of reliable broadcast protocols for operation delivery
  - Use consensus between group members for operation commits

- Modified/updated keys
- Keys on co-path, used to encrypt new keys to subgroups

Updating member B

Adding member E

A B C D
Access control for peer-to-peer systems without central authority

Goals
- Replicated data
- Replicated access rights
- Multiple administrators

Approach
- Compose CRDTs for data with CRDTs for access rights [RIP23]
- Dependencies
  - Between access rights operations
  - Between document and access rights

Example:
- Doc: Hello
  - Site1: Read, Write
  - Site2: Admin, Read, Write

Function: write ('World', 6)
- Doc: Hello World
  - Site1: Read, Write
  - Site2: Admin, Read, Write

Function: removeRight(W, Site1)
- Site1: Read
  - Site2: Admin, Read, Write

Function: undo write ('World', 6)
- Doc: Hello
  - Site1: Read, Write
  - Site2: Admin, Read, Write

Function: write ('World', 6)
- Doc: Hello World
  - Site1: Read, Write
  - Site2: Admin, Read, Write

Function: undo write ('World', 6)
- Doc: Hello
  - Site1: Read, Write
  - Site2: Admin, Read, Write
P2P Cloud Storage – Défi Alvearium with hive

Replication and security mechanisms application

- A distributed peer-to-peer cloud exploiting unused capacity of computers
  - +20% of computing resources remain unused
  - Allows users to share these computing resources
  - Builds a virtual cloud to offer compute and storage services

- Sustainable
- Secure & private
- Data sovereignty
- Affordable
P2P Cloud Storage – collaboration with hive

Research topics

- How to place data in the peer-to-peer storage in order to ensure high availability?
- How to manage data that can be modified by different peers?
- How to deal with Byzantine nodes? How to handle Sybil attacks?
- How to define a peer-to-peer security mechanism (manage access rights, client-side E2E encryption, group key management)
Research issues

1. How to maintain consistency of different copies in the face of concurrent modifications?

2. How to secure collaboration data?

3. How to make a collaboration trustworthy?
Trust in collaborative tools \cite{L21}

- Crisis management – issues on information sharing
- Technical issues: tools’ complexity, network delay, no interoperability
- Communicational issues
Peer-to-peer user trust

• Prediction of future collaborative user behavior based on the quality of user contributions
Trust computation - contribution

- Evaluate collaborators trust based on their past behavior
  - Contract-based collaborative editing
  - Trust game
Trust computation in collaborative editing

- Respect/Violation of contracts [TI11, TIBM11, TIM12, ID21]

![Diagram showing trust relationships and contract enforcement](image)
Peer-to-peer trust metric in trust game \[\text{[BDM95]}\]

- **Design a trust metric** dealing with fluctuating user behaviour \[\text{[DI16a]}\]
  - For a round:  
    \[ \text{current}_\text{trust}_i = \log(\text{send}_\text{proportion}_i \times (e - 1) + 1) \]
    \[ \text{send}_\text{proportion}_i = \frac{\text{sending}_\text{amount}_i}{\max_\text{sending}_\text{amount}} \]
  - For accumulated rounds:
    \[ \text{aggregate}_\text{trust}_i = \alpha_t \times \text{current}_\text{trust}_i + (1 - \alpha_t) \times \text{aggregate}_\text{trust}_{i-1} \]
Validation of the trust metric – experimental design [IDS19]

- **Experiment with users** on 4 trust games: Simple Game, Identity Game, Score game and Combine game

- Showing either trust score or identity improves the measure
- Showing both of them does not change the measure
- Trust-based systems can be a replacement of identity-based systems
- Collaboration with Department of Psychology Wright State University
PEPR eENSEMBLE (https://pepr-ensemble.fr/)
PILOT – Practices and infrastructure for long term collaboration

• Socio-technical models for long-term collaboration
• Multi-organizational, sovereign, secure, interoperable collaboration platforms promoting trust and digital well-being

• Coordinators
  • Claudia-Lavinia Ignat, Inria
  • François Charoy, Université de Lorraine
  • Myriam Lewkowicz, Université de Technologie de Troyes
PILOT – Project axes

• Axis 1: Understanding current and future forms of long-term collaboration
• Axis 2: Open technical frameworks and protocols for long-term collaboration
• Axis 3: Conceptual Frameworks for long-term collaboration
• Axis 4: Actionable guidelines and demonstrators for long term collaboration
PILOT architecture

1. Identify new forms of collaboration / new social needs
   - Collect, build and organize a repository of up-to-date analyses, use cases and reports about future forms of long-term collaboration, their challenges and their limits.

2. Investigate new technological capacities
   - Foundations for the future infrastructure for long-term collaboration by providing new technology components that can be at the heart of safe, sustainable and secure collaborative services.

3. Define conceptual and analysis tools to ensure socio-technological approaches
   - New conceptual constructs for analysis, modelling, and design for articulating social needs and technology capabilities in a socio-technical perspective,

4. Provide actionable visions, guidelines and patterns
   - Prototypes and demonstrators, guidelines, visions, patterns
PILOT – Resources

• Budget: 5 M€
• Period: 7 years (Sept. 2023 – Aug. 2030)
• Non permanent resources
  • 18 PhD theses
  • 4 postdocs of 2 years
  • 2 engineers of 4 years
  • 12 internships of 6 months
• 14 partner institutions: Inria, CNRS, Univ. Paris-Saclay, Univ. Grenoble Alpes, Sorbonne Univ., IMT, UTT, INSA Lyon, UCB, Nantes Université, ENSAM, Univ. de Lille, Univ. de Toulouse
Large-scale trustworthy distributed collaborative systems

- New uses and new practices due to large scale adoption
- New challenges
  - Consistency of replicated complex data
  - Security mechanisms without a central authority
  - Trust in collaborative platforms and trust among collaborators
Thank you

COAST Team
http://team.inria.fr/coast/
References

- **[AII23]** Quantifying the Performance of Conflict-free Replicated Data Types in InterPlanetary File System. Q. Acher, C.-L. Ignat, S. Ibrahim. DICG@Middleware 2023


- **[ASB18]** Delta state replicated data types. P. S. Almeida, A. Shoker, C. Baquero. Journal of Parallel and Distributed Computing, 2018


- **[DI16b]** Computational Trust Model for Repeated Trust Games. Q. V. Dang, C.-L. Ignat. Trustcom 2016

- **[EG89]** Concurrency Control in GroupWare Systems, C.A. Ellis and S.J. Gibbs. ACM SIGMOD Record 18(2), 1989.

References

- [ID21] Users trust assessment based on their past behavior in large scale collaboration. C.-L. Ignat, Q.-V. Dang. ICCP 2021
References

- **[NI18]** An Analysis of Merge Conflicts and Resolutions in Git-Based Open Source Projects. H. L. Nguyen, C.-L. Ignat. Journal of CSCW, 2018


- **[PMSL09]** A commutative replicated data type for cooperative editing, N. Preguica, M. Joan, M. Shapiro, and M. Letia. ICDCS 2009.


