Decentralized Internet Archive using the Cothority framework

Nicolas PLANCHEREL

Prof. Bryan FORD
Eleftherios KOKORIS KOGIAS
Kirill NIKITIN

Decentralized and Distributed Systems (DEDIS)
School of Computer and Communication Sciences (IC)
École Polytechnique Fédérale de Lausanne (EPFL)

EPFL, Master Thesis oral presentation February 2018
Outline

- Motivation
- Description
- Evaluation And Discussion
- Demo
- Conclusion
MOTIVATION
Motivation

Objectives

- Create a censorship resistant internet archive
  - Archiving avoiding tampering or deletion (by one or a small collusion of entity)
  - Store only relevant content
  - Possibility to check integrity once archived
  - Consider that the censor can try to add, modify or delete data
Motivation
Context - Centralized Internet
Motivation

Context - Centralized Internet

- Malicious actor that tamper data
Motivation

Context - Centralized Internet

✗ Malicious actor that tamper data
✗ Weak regarding to long-term availability
Motivation

Context – Centralized Internet - Archive.org

✗ Malicious actor that tamper data
✗ Weak regarding to long-term availability
✔ Archiving ! (on demand)
Motivation

Context – Centralized Internet - Archive.org

✗ Malicious actor that tamper data
✗ Weak regarding to long-term availability

✔ Archiving!
✗ Still vulnerable to malicious archive!
Motivation

Context - Decentralized Internet – ZeroNet

✔ Distributed By Design !
✔ Strong regarding long-term availability
✔ Censorship resistant
✗ No interaction with today’s internet
Motivation

Overview

- Centralized Internet is vulnerable to censorship
  - Malicious actor
  - Deletion and Tampering
- Solutions exist but still have weaknesses
  - Centralized: Archive.org
  - Decentralized: ZeroNet
- So we developed a Decentralized Internet Archive
DESCRIPTION
Description

Objectives

- Create a censorship resistant internet archive
  - Avoid Tampering using decentralized storage system: Skipchain
  - Filter content by reaching a consensus on the content of the webpage
  - Using the CoSi Service of the Cothorithy framework (collective signature)
  - Avoid adding malicious data using a trusted reference to make a consensus on
Operations

- Save
  - Consensus on the content of the webpage
  - Collectively Sign the common subset of the page
  - Store the signed page on the skipchain

- Retrieve
  - Get the correct signed page
  - Verify the signature
Description
High-Level

Saving/Receiving Service

Skipchain Handling Service
Description
Saving (with a tree-based consenus protocol)

1. Request web page

1. Save Request

EPFL, Master Thesis 2018
Decentralized Internet Archive
Description
Saving (with a tree-based consensus protocol)

1. Request web page
2. Leader get page
Description
Saving (with a tree-based consensus protocol)

1. Request web page
2. Leader get page
3. Leader transmit url and tree

1. Save Request
2. Get web page
3. Create Tree
4. Send Tree + url
Description

Saving (with a tree-based consensus protocol)

1. Request web page
2. Leader get page
3. Leader transmit url and tree
4. Every child get own tree and send common parts
Description
Saving (with a tree-based consensus protocol)

1. Request web page
2. Leader get page
3. Leader transmit url and tree
4. Every child get own tree and send common parts
5. Leader create common tree
Description
Saving – Creating the HTML tree

<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <h1>DECENARCH</h1>
  </body>
</html>

- A web page consists of
  - An html code text
  - Additional Data
    - Images
    - CSS file(s)

- Get Html Code
Description
Saving – Creating the HTML tree

<!doctype html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="css/style.css">
    </head>
    <body>
        <h1>DECENARCH</h1>
    </body>
</html>

- Get Html Code
- Infer Html Tree from code
Description
Saving – Creating the HTML tree

```html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <h1>DECENARCH</h1>
  </body>
</html>
```

- Get Html Code
- Infer Html Tree from code
- Hash the data of every node individually
Description

Saving – Signing the HTML tree

- Leader’s MasterTree
Description

Saving – Signing the HTML tree

- Leader’s Master Tree
- Child’s Local Tree
Description

Saving – Signing the HTML tree

- Leader’s MasterTree
- Child’s Local Tree
Description
Saving – Signing the HTML tree

- Leader’s MasterTree
- Child’s Local Tree
Description
Saving – Signing the HTML tree

- Leader’s MasterTree
- Child’s Local Tree
Saving – Signing the HTML tree

- Leader’s MasterTree
- Child’s Local Tree
Description
Saving – Signing the HTML tree

🔹 Leader’s MasterTree

🔹 Child’s Local Tree
Description
Saving – Signing the HTML tree

- Leader’s MasterTree
- Child’s Local Tree
Description
Saving – Signing the HTML tree

- Leader’s MasterTree
- Nodes in BFS order
- Seen array
  \[1_A \ 1_B \ 1_C \ 1_D \ 1_E \ 0_F \ 1_G \ 1_H \ 0_I \ 1_J\]
- Signature
  \[\text{sign}(h_A + h_B + h_C + h_D + h_E + 0 + h_G + h_H + 0 + h_I)\]
Description
Saving - Aggregation

- Leader’s MasterTree

◆ Seen arrays

\[
\begin{align*}
\sum_3 & A \quad 3 \quad B \quad 3 \quad C \quad 2 \quad D \quad 3 \quad E \quad 2 \quad F \quad 2 \quad G \quad 3 \quad H \quad 2 \quad I \quad 2 \quad J \\
1_1 & A \quad 1 \quad B \quad 1 \quad C \quad 1 \quad D \quad 1 \quad E \quad 1 \quad F \quad 1 \quad G \quad 1 \quad H \quad 1 \quad I \quad 1 \quad J \\
1_1 & A \quad 1 \quad B \quad 1 \quad C \quad 1 \quad D \quad 1 \quad F \quad 1 \quad G \quad 1 \quad H \quad 0 \quad I \quad 1 \quad J \\
1_1 & A \quad 1 \quad B \quad 1 \quad C \quad 0 \quad D \quad 1 \quad E \quad 0 \quad F \quad 1 \quad H \quad 1 \quad I \quad 0 \quad J
\end{align*}
\]
Description
Saving - Aggregation

- Leader’s MasterTree

- Seen arrays

\[
\begin{array}{cccccccccc}
A & B & C & D & E & F & G & H & I & J \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
\sum & 3 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 \\
\end{array}
\]

- Keep A, B, C, E, H

Output html code collectively signed
Description
Handling the Skipchain

- We have: A representation of the common subset of the page, collectively signed
- We want: An efficient, tampering resistant storage system
Description
Retrieving the archived web page

1. Send new data.

1. Request URL & time

1. Request URL & time
Description
Retrieving the archived web page

1. Send new data.

2. Look in blockchain to find correct data

1. Request URL & time

2. == URL ? && ≤ time
Description

Retrieving the archived web page

1. Send new data.
2. Look in blockchain to find correct data.
3. Send back the data.

3. Save on fs
EVALUATION AND DISCUSSION
Evaluation and Discussion

- Does it scale in terms of:
  - Bandwidth use?
  - Time complexity?

- The ‘trusted leader’ constraint
Evaluation and Discussion

Evaluation - Theory

\[ N \] number of machines
Evaluation and Discussion

Evaluation - Theory

N number of machines

K html node’s number

W size of the webpage

EPFL, Master Thesis 2018
Decentralized Internet Archive
Evaluation and Discussion

Evaluation - Theory

\[ N \] number of machines

\[ K \] html node's number

\[ W \] size of the webpage

\[ A \] additional resources (image, css)
Evaluation and Discussion

Bandwidth

- Variables:
  - $N$ number of machines.
  - $W$ size of webpage.

- Bandwidth use is linear $O(N \cdot W)$
  - $N + 1$ request to the distant server of size $O(W)$
  - Finite total number of message of size $O(W)$
Evaluation and Discussion
Evaluation - Theory

- Variables definitions:
  - \( N \) number of machines.
  - \( K \) html node’s number.
  - \( A \) time cost of handling additional data (image, css) on one machine.

- Overall save time complexity is polynomial\( O(N \cdot K^2 + (1+A) \cdot N \cdot K + N) \):
  - Tree comparison and aggregation is in \( O(N \cdot K^2) \)
  - Handling the additional data of the web page is in \( O(A \cdot N \cdot K) \)
  - Storing the website is in \( O(N \cdot K) \)
  - Collective signing is in \( O(N) \)
Evaluation and Discussion

Evaluation - Simulations

- Standardized Website
- Html Tree Node increase
Evaluation and Discussion

Evaluation - Simulations

- Real-Life Website
- Html Tree Node increase

- Main time component:
  Handling the additional data.
Evaluation and Discussion
Evaluation - Simulations

- Standardized Website
- Conode nbr increase

- Main time component:
  The consensus

- Seems linear but require a larger simulation
Why the trusted leader?
Evaluation and Discussion

Discussion

- Why the trusted leader?
  - Why the tree structure?
    - Keep a valid html document anytime.
    - Granularity.
Evaluation and Discussion

Discussion

- Why the trusted leader?
  - Why the tree structure?
    - Keep a valid html document anytime.
    - Granularity.
  - Why a reference?
    - Union of Tree is NP.
    - Undeterministic matching, depends on order.
"Anything that can go wrong will go wrong".
- Murphy’s Law
Demo

Ain’t nobody got time for demo
Conclusion

- Decentralized Internet Archive
  - Tree-based consensus with largest common subset
  - Decentralized storage with skipchain
  - Has a polytime complexity in $O(K^2 \cdot N)$

- Improvements?
  - Storage Management
  - Additional Data filtering
  - Finer granularity
  - Confidentiality
Conclusion

- Decentralized Internet Archive
  - Tree-based consensus with largest common subset
  - Decentralized storage with skipchain
  - Has a polytime complexity in $O(K^2 \cdot N)$

- Improvements?
  - Storage Management
  - Additional Data filtering
  - Finer granularity
  - Confidentiality
• [Master Thesis] Plancherel Nicolas 2018, Decentralized Internet Archive, EPFL