A Simulation-based Evaluation of a Hybrid Storage System combining P2P, F2F, and Cloud storage with a Distributed Reputation System

Anders Skoglund

andsk668@student.liu.se

November 04, 2013
Storage methods

- Cloud storage
- P2P storage
- F2F storage
- Hybrid storage + reputation system
Storage methods

Cloud storage

Advantages

+ Scalable
+ High availability
+ Contractual accountability

Disadvantages

– Cost
– Possible single point of failure
Storage methods
P2P storage

Advantages

+ Large number of peers
+ Scalable
+ No single point of failure
+ Cooperative / low cost

Disadvantages

– Semi-anonymous peers
– No accountability
– Peers can’t be trusted
Storage methods
F2F storage

Advantages
+ No single point of failure
+ Cooperative / low cost
+ Social accountability
+ Known and trusted peers

Disadvantages
– Few peers
– Possible availability issues
Storage methods
Hybrid storage + reputation system

Advantages

+ Scalable
+ Trusted friend peers
+ Predictable peer behavior
+ No single point of failure
+ Cooperative / low cost

Disadvantages

- ?
System Overview
Components

- DHT
- Distributed search
- Block distributor
- Reputation system
Algorithm 1 BasicEigenTrust

\begin{algorithm}
\begin{align*}
C &: \text{ Local trust matrix.} \\
\vec{p} &: \text{ Vector of relative trust values for all pre-trusted peers.} \\
\vec{t} &: \text{ Vector of global trust values for all peers.} \\
a &: \text{ Weight given to pre-trusted peers when computing global trust.} \\
\epsilon &: \text{ Threshold used to stop the algorithm once it converges.}
\end{align*}

1: \textbf{function} \textsc{ComputeTrust}(C) \\
2: \quad \vec{t}^0 \leftarrow \vec{p} \\
3: \quad k \leftarrow 0 \\
4: \quad \text{repeat} \\
5: \quad \quad \vec{t}^{k+1} \leftarrow (1 - a)C^T\vec{t}^k + a\vec{p} \\
6: \quad \quad \delta \leftarrow ||\vec{t}^{k+1} - \vec{t}^k|| \\
7: \quad \quad k \leftarrow k + 1 \\
8: \quad \text{until} \ \delta < \epsilon \\
9: \quad \text{return} \ \vec{t}^k \\
10: \textbf{end function}
\end{algorithm}
Algorithm 2 SecureEigenTrust

\[C\] : Local trust matrix.
\[\mathbf{p}\] : Vector of relative trust values for all pre-trusted peers.
\[\mathbf{t}\] : Vector of global trust values for all peers.
\[A_d\] : Peers that have reported local trust values about a daughter peer \(d\).
\[B_d\] : Peers that a daughter peer \(d\) has reported local trust values about.
\[D\] : Daughter peers of the score manager.
\[M_i\] : All score managers for the peer \(i\).
\[a\] : Weight given to pre-trusted peers when computing global trust.
\[\epsilon\] : Threshold used to stop the algorithm once it converges.

1: \textbf{function} ComputeTrust\((C)\)
2: \quad \textbf{for each} \(d \in D\) \textbf{do}
3: \quad \quad \(A_d \leftarrow \text{ServedByDaughter}(d)\)
4: \quad \quad \(B_d \leftarrow \text{HasServedDaughter}(d)\)
5: \quad \quad \(k \leftarrow 0\)
6: \quad \quad \textbf{for each} \(j \in A_d\) \textbf{do}
7: \quad \quad \quad \(c_{jd} \leftarrow \text{QueryLocalTrust}(\text{Hash}(j))\)
8: \quad \quad \textbf{end for}
9: \quad \textbf{repeat}
10: \quad \quad \(t_{d}^{k+1} \leftarrow (1 - a) \sum_{j=1}^{n} c_{jd} t_{j}^{k} + a p_{d}\)
11: \quad \quad \textbf{for each} \(j \in B_d\) \textbf{do}
12: \quad \quad \quad \(M_j \leftarrow \text{Hash}(j)\)
13: \quad \textbf{end for}
14: \quad \quad \text{SendLocalTrust}(c_{dj}, M_j)
15: \quad \quad \text{SendGlobalTrust}(t_{d}^{k+1}, M_j)
16: \quad \textbf{end for}
17: \quad \textbf{for each} \(j \in A_d\) \textbf{do}
18: \quad \quad \(M_j \leftarrow \text{Hash}(j)\)
19: \quad \quad \(c_{jd} \leftarrow \text{ReceiveLocalTrust}(M_j)\)
20: \quad \quad \(t_{j}^{k+1} \leftarrow \text{ReceiveGlobalTrust}(M_j)\)
21: \quad \textbf{end for}
22: \quad \textbf{until} \ |t_{d}^{k+1} - t_{d}^{k}| < \epsilon\)
23: \quad \textbf{end for}
24: \textbf{end function}
Reputation systems
EigenTrust

\[ s_{ij} = \text{sat}(i, j) - \text{unsat}(i, j) \]

\[ c_{ij} = \begin{cases} \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}, & \text{if } \sum_j \max(s_{ij}) \neq 0 \\ p_j, & \text{otherwise} \end{cases} \]

\[ p_i = \begin{cases} \frac{1}{|P|}, & \text{if } i \in P \\ 0, & \text{if } i \notin P \end{cases} \]

\[ t_j = (1 - a) \sum_i c_{ij} t_i + a p_j \]

+ Simple
+ Well analyzed
+ Scalable
- Very simple trust model
- Relative trust values
Modified EigenTrust

\[ S_{ij} = \begin{cases} 
\frac{\text{sat}(i, j)}{\text{sat}(i, j) + \text{unsat}(i, j)}, & \text{if } \text{sat}(i, j) + \text{unsat}(i, j) \neq 0 \\
0, & \text{otherwise} 
\end{cases} \]

\[ C_{ij} = S_{ij} \]

\[ t_j = \begin{cases} 
\sum_i c_{ij} w_{ij}, & \text{if } j \notin P \\
1, & \text{if } j \in P 
\end{cases} \]

- Computes more useful trust values than EigenTrust
- Can use the same distributed algorithms as EigenTrust

\[ w_{ij} = \frac{t_i}{\sum_{k \in A_j} t_k} \]
Reputation systems

Global trust

Global trust (P2P)
- Single trust value per peer
- Scalable
- Many malicious peers
Community trust

- Separate trust values are computed by each peer
- Not scalable
- More trusted peers
- Fewer peers and resources
- Can get stuck in local maximum

Community trust (F2F)
A Simulation-based Evaluation of a Hybrid Storage System
Honest peer

- Always fulfill transactions
- Give honest ratings

Malicious peer

- Always fulfill transactions with malicious peers
- Otherwise return “corrupt” data with probability $p_m$
- Always gives positive ratings to malicious peers
- Otherwise give false (negative) ratings
Distribution policies

- **Global**: Only use global (P2P) peers
- **Community**: Only use community (F2F) peers
- **Mixed**: Use both global (P2P) and community (F2F) peers

File types

- **0.0**: Trust $\geq 0.0$
- **0.3**: Trust $\geq 0.3$
- **0.6**: Trust $\geq 0.6$
- **0.9**: Trust $\geq 0.9$
Results
Malicious rate × Average global trust

- Ideal would be: 
  \[ T_{honest} \approx 1 \]
  and
  \[ T_{malicious} \approx \text{malicious rate} \]
- Global trust is close to the ideal value for both honest and malicious peers
Results

Malicious rate $\times$ Block recovery rate

![Graph showing the relationship between Malicious rate and Block recovery rate. The graph includes lines for different policies: No-trust, Mixed, Global, and Community. Each policy has a legend indicating the file types (0.0, 0.3, 0.6, 0.9). The x-axis represents Malicious rate, ranging from 0.0 to 1.0, and the y-axis represents Block recovery rate, ranging from 0.0 to 1.0. The community size is 20, and resources are 100.]
Results

Malicious rate × Block recovery rate (Global peers)

![Graph showing the relationship between malicious rate and block recovery rate. The graph includes different file types, represented by various markers and colors. The x-axis represents the malicious rate, ranging from 0.0 to 1.0, while the y-axis represents the block recovery rate, also ranging from 0.0 to 1.0. The graph includes data points for different file types, marked at 0.0, 0.3, 0.6, and 0.9, with corresponding lines showing the trend.]

Honest: 20 Malicious: 80 Resources: 100 Community size: 20

$t \geq 0.0$
$t \geq 0.3$
$t \geq 0.6$
$t \geq 0.9$
Results
Malicious rate × Block recovery rate (Global peers)

A Simulation-based Evaluation of a Hybrid Storage System
Results
Malicious rate × Block recovery rate (Global peers)

\[ \frac{20+5}{105} + \frac{80}{105} \times 0.3 \]

Block recovery rate (global)
Honest: 20 Malicious: 80 Resources: 100 Community size: 20

File types
- 0.0
- 0.3
- 0.6
- 0.9

A Simulation-based Evaluation of a Hybrid Storage System
Results

Malicious rate × Block recovery rate (Global peers)

\[ \frac{20 + 5}{105} + \frac{80}{105} \times 0.6 \]

Block recovery rate (global)
Honest: 20 Malicious: 80 Resources: 100 Community size: 20

File types

- 0.0
- 0.3
- 0.6
- 0.9
Results
Malicious rate × Block recovery rate (Global peers)

\[
\frac{20+5}{105} + \frac{80}{105} \times 0.9
\]

\[
t \geq 0.9
\]

A Simulation-based Evaluation of a Hybrid Storage System

A. Skoglund
Results
Malicious rate × Block recovery rate (Trust ≥ 0.0)

Mixed > Community > Global

A Simulation-based Evaluation of a Hybrid Storage System
Results
Malicious rate $\times$ Block recovery rate ($\text{Trust} \geq 0.3$)

Mixed > Community > Global

Block recovery rate
Honest: 20 Malicious: 80 Resources: 100 Community size: 20

Malicious rate
Block recovery rate
Policy No−trust Mixed Global ... 0.3 0.6 0.9

File types $0.0$ $0.3$ $0.6$ $0.9$
Results

Malicious rate × Block recovery rate (Trust ≥ 0.9)

Mixed > Global > Community

A Simulation-based Evaluation of a Hybrid Storage System
Global trust (P2P) performs better when you need a large number of peers. It is possible to compensate for low trust, and it will perform well as long as the user has chosen appropriate trust requirements.

Community trust (F2F) performs better when there are enough peers and resources available in the community graph. It requires much less care when choosing trust requirements, but can easily fail if there are not enough peers available.

A combination of global (P2P) and community (F2F) trust performs at least as well as the best of the two, and often better.
A hybrid system could work, but some improvements must first be made.

Both the P2P and F2F part of the system performs better than expected.

Combining P2P and F2F gives the best performance.

But using a reputation system may be the most important part.
### Possible improvements

**Create a better model of peer availability and storage costs**

Without a better model of availability or storage cost, and a distribution algorithm that can use it, there is no point in simulating cloud storage.

**Find a better trust model**

EigenTrust has limitations that could make it a bad choice for this task.

**Generalize**

This kind of system should be able to handle any task that can be described as a transaction. It should be possible to use this to create a platform for trading/sharing other types of resources and use it for, among other things, distributed computing.
Questions?
Additional details

DHT

A Simulation-based Evaluation of a Hybrid Storage System
A Simulation-based Evaluation of a Hybrid Storage System

Additional details

Distributed search

Query
X:0.0-0.25   Y:0.3-0.7

Result
000100
000101
000110
000111
010000
010001
010010
010011
Additional details
Distributed search

Search space

Chord ring

A Simulation-based Evaluation of a Hybrid Storage System
A Simulation-based Evaluation of a Hybrid Storage System