Long-term Digital Archiving Based on Selection of Repositories Over P2P Networks

(IEEE P2P 09)

Luis C. E. Bona
Tiago Vignatti, Luis C. E. Bona, André Vignatti, Marcos Sunye

C3SL - Center for Scientific Computing and Free Software
Department of Informatics
Federal University of Parana Curitiba - Parana - Brazil

Seattle – September 10th 2009
Outline

1. Introduction
2. Model
3. A Peer-to-Peer Digital Archiving System
4. Conclusion
Outline

1. Introduction
2. Model
3. A Peer-to-Peer Digital Archiving System
4. Conclusion
The Importance of Digital Data

- Society’s heritage has been presented on many different materials, including stone, vellum, bamboo, silk, paper and etc. But today a considerable amount of relevant information exists in digital formal.
- The interesting point is that the digital media is more vulnerable to deterioration than some old fashion media.
- Who Never Lost Digital Data Cast The First Stone!
Some Examples

- One notable example is the Domesday Book, a record of English settlements compiled by William the Conqueror in 1085. The document survives in a secure, environmentally controlled facility, but a digitized version produced in 1986 lasted only 20 years.

- September 11th, 2001, a total of 60 nonprofit organizations had offices in WTC. Many of them lost records, and historical data.

- Likewise, traditional photos have been preserving family memories for decades, whereas our DVDs can deteriorate in few years.
The goal of Digital preservation is ensure long-term, error-free retrieval and interpretation of digital information.

Long-term may extend indefinitely.

Main issue: (a) Bit stream preservation; (b) Format obsolescence.

In our context Digital Archiving tries to guarantee Bit Stream Preservation.

One of our main concerns is cost.

An alternative is to replicate the information in multiple storage repositories, consisting of conventional and low cost computers.
Peer-to-Peer (P2P) networks appear as a promising approach to organize these multiple storage repositories. But most of P2P networks do not ensure long-term data preservation. Our work is to design a totally distributed P2P archiving system. We defined a model for reliable replication of immutable data. We also showed how our model can be implemented over DHT P2P networks.
Outline

1. Introduction
2. Model
3. A Peer-to-Peer Digital Archiving System
4. Conclusion
Items (Digital Information) and Repositories, they are all different

A **reliability metric** is associated with each repository
- This metric denotes the probability that the data will not be lost or damaged in a given period of time.

Each item needs to be stored with a **desired reliability** that reflects its importance

To ensure the desired reliability of an item, several replicas are created in different repositories.
- The number of replicas required in order to preserve a specific item is given by item desired reliability and by reliability of each repository in which replicas are stored.
Selecting Repositories

Multiple choices are possible:

1. \[1 - ((1-0.99) \times (1-0.95) \times (1-0.85)) = 99.993 \%\]

2. \[1 - ((1-0.95) \times (1-0.95) \times (1-0.85) \times (1-0.75)) = 99.991 \%\]
The model is composed by:

- A set $N$ of $|N| = n$ items. Each item has a size $s_i$ and an assigned probability $0 < r_i < 1$, called the desired reliability of an item.

- A set $M = \{\ell_1, \ldots, \ell_m\}$ of $|M| = m$ repositories, where each repository $\ell_j$ has associated a storage capacity $c_j$ and a probability $0 < p_j < 1$, called the reliability of the repository.

- The reliability of the subset $S \subseteq M$ is defined as $1 - \prod_{\ell_j \in S} (1 - p_j)$, which denotes the probability of at least one repository in $S$ does not lose data in a given time interval.
We define repositories selection problem as follows:

- Choose a subset $S_i$ of repositories where each repository in $S_i$ receives a copy of the item (replica) to satisfy the desired reliability $r_i$. In other words, we select $S_i \subseteq M$ such that:

$$1 - \prod_{\ell_j \in S_i} (1 - p_j) \geq r_i. \quad (1)$$

- In addition, each repository in $S_i$ must have enough free space to receive the copy
Strategies for Replica Creation

- The selection is over $M^o \subseteq M$, a selection over $M$ would be unfeasible in real world implementations
  - $M^o$ is selected at random from $M$; in practice, the way that $M^o$ is selected may depend on the system features
- We tested 3 strategies: Randomized, Greedy and Ideal Subset
- In all strategies, when a replica is assigned to a full repository, we ignore it and randomly select another repository of the considered set.
Randomized

begin
\[ S = \emptyset \]
while reliability of \( S \) is less than \( r_i \) do
\[ \text{choose } \ell_j \in M^o \text{ uniformly at random} \]
\[ S = S \cup \{\ell_j\} \]
end

return \( S \)

**Algorithm 1**: Randomized
Greedy

begin
    sort $M^o$ in non-increasing order according to the values $a_j$
    $S = \emptyset$
    $t = 1$
    while $\sum_{\ell_j \in S} a_j < b_i$ do
        $S = S \cup \{\ell_t \in M^o\}$
        $t = t + 1$
    return $S$
end

Algorithm 2: Greedy over Subset
To create the replicas, we select the subset $S \subseteq M^o$ that provides the reliability that is closest to the desired reliability of the item. It is like avoiding waste resource.

It's very similar to *Knapsack problem*.

There is a *dynamic programming* algorithm which solves this problem in pseudo-polynomial time, but in practice is satisfactory for the vast majority of instances.
Evaluation

- Number of replicas created
- Makespan
- Number of Items Inserted
Number of Items Inserted

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>40</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomized</td>
<td>18569</td>
<td>30228</td>
<td>32464</td>
<td>34138</td>
</tr>
<tr>
<td>Greedy over Subset</td>
<td>17544</td>
<td>28182</td>
<td>30891</td>
<td>33899</td>
</tr>
<tr>
<td>Ideal Subset</td>
<td>20535</td>
<td>31094</td>
<td>35399</td>
<td>39107</td>
</tr>
</tbody>
</table>

Tabela: Insertion of the items in function of the size of the subset.

- Items have size of 35 MB and desired reliability of 99%.
- Repositories storage capacity varies between 100 MB and 100.000 MB
- Repositories average reliability is 67% (standard deviation of 17%)
Outline

1. Introduction
2. Model
3. A Peer-to-Peer Digital Archiving System
4. Conclusion
The replication model proposed is designed in a generic way and can be implemented on any distributed mechanism for organizing the storage repositories.

In particular, structured P2P networks using DHT appears as natural candidates as they are highly scalable for data distribution and retrieval.

However, it is not trivial to select a specific subset of nodes using the routing method from DHTs.

An extra difficult because we want a pure P2P system (no Super Peers).
Let $h_1, h_2, \ldots, h_r$ be the $r$ hash functions. The hash functions have global visibility, i.e., they are the same for all nodes.

Given the key $k$ of a digital object, we apply $k$ to the hash functions, i.e., $h_1(k), h_2(k), \ldots, h_r(k)$.

Thus, for each object to be inserted, we get $r$ nodes where we can place replicas (i.e., the set $M^o$).

This approach allow us to know where replicas should be without the need of storing information about their location.
Examples of $M^o$:

- A white circle represents a set $M^o$.
- Black nodes represent repositories selected in each set $M^o$.
System Operations

- Fundamental operations are insert and retrieve.
- Auditing is a work in progress. It could improve system efficiency.
input: key, value, reliability $r_i$
begin
  $M^o = \emptyset$
  for $i = 1$ to $r$
do
    $M^o = M^o \cup \{\ell(h_i(key))\}$
  $S = \text{insertion}\_\text{strategy}(M^o, r_i)$
  foreach $s \in S$ do
    $j \leftarrow \text{hash function number of } s$
    put($h_j(key)$, value)
end

Algorithm 3: insert (key, value, reliability)
Retrieve

Algorithm 4: retrieve (key)

input: key
begin
  for i = 1 to r do
    value ← get(h_i(key))
    if value is not null then
      return value
  return −1 /* not found */
end
Experimental Results

- The P2P archiving system was implemented and evaluated through experiments carried out in a real world environment.
- Overlayweaver was the P2P’s implementation used. Overlay Weaver has itself a mechanism of replication that has been turned off for the evaluation of our experiments.
- It was just a proof of concept.
Digital Preservation is a real problem today and need to be tackled.

We can conclude two main characteristics of the P2P system and the replication model proposed:

- **Independence of object preservation**: different information requires different storage time. Collections of photos, journals and articles may need few years of storage; other information such as digital objects in museums and libraries requires hundreds of years.

- **Optimization of the storage resources**: storage repositories may suffer many types of damages on their contents, so each repository has a different reliability. Allowing each storage repository with a parameter capable of measuring the independent probability of failure is the closest way to model real networks.

Backup your data today :)
Future Work

- Improve the model, e.g., avoid fault correlation.
- Auditing support. If we check replicas in short times we can fix it and improve reliability. But auditing requires read all data an expensive task.
- How estimate repositories reliability? Reputation?
- Implement a ready to use system based on these ideas.
Questions, Answers, Ideas, Suggestions?