Creating Trading Networks of Digital Archives

Brian Cooper and Hector Garcia-Molina
Department of Computer Science
Stanford University
{cooperb,hector}@db.stanford.edu

ABSTRACT

Digital archives can best survive failures if they have made several copies of their collections at remote sites. In this paper, we discuss how autonomous sites can cooperate to provide preservation by trading data. We examine the decisions that an archive must make when forming trading networks, such as the amount of storage space to provide and the best number of partner sites. We also deal with the fact that some sites may be more reliable than others. Experimental results from a data trading simulator illustrate which policies are most reliable. Our techniques focus on preserving the "bits" of digital collections; other services that focus on other archiving concerns (such as preserving meaningful metadata) can be built on top of the system we describe here.

KEYWORDS

preservation, digital archiving, replication, fault tolerance, data trading

1. INTRODUCTION

Digital materials are vulnerable to a number of different kinds of failures, including decay of the digital media, loss due to hackers and viruses, accidental deletions, natural disasters, and bankruptcy of the institution holding the collection. Archives can protect digital materials by making several copies, and then recover from losses using the surviving copies. Copies of materials should be made at different, autonomous archives to protect data from organization-wide failures such as bankruptcy. Moreover, cooperating archives can spread the cost of preservation over several institutions, while ensuring that all archives achieve high reliability. Several projects [4, 12, 24, 10] have proposed making mutiple copies of data collections, and then repeatedly checking those copies for errors, replacing corrupted materials with pristine versions.

A key question for a digital archive participating in a replication scheme is how to select remote sites to hold copies of collections. The archivist must balance the desire for high reliability with factors such as the cost of storage resources and political alliances between institutions. To meet these goals, we propose that archives conduct peer-to-peer (P2P) *data trading*: archives replicate their collections by contacting other sites and proposing trades. For example, if archive A has a collection of images it wishes to preserve, it can request that archive B store a copy of the collection. In return, archive A will agree to store digital materials owned by archive B, such as a set of digital journals. Because archive A may want to make several copies of its image collection, it should form a *trading network* of several remote sites, all of which will cooperate to provide preservation.

In previous work [5], we have studied the basic steps involved in trading and the alternatives for executing these steps. For example, in one step a local site selects a trading partner from among all of the archive sites. This requires the local site to choose some strategy for picking the best partner. In another step, the local site asks the partner to advertise the amount of free space it is willing to trade. Then, the local site can determine if the partner will trade enough space to store the local site's collections. We summarize our conclusions from this previous study for these and other issues in Section 2.2 below.

In this paper, we discuss how a digital archive can use and extend these basic trading building blocks to provide preservation services. Archives must take into consideration real-world issues that impact the decisions they make while trading. For example, an archive may have budgetary constraints that limit the amount of storage it can provide. Storage resources cost more than just the expense of buying disk space. In particular, an archive must also provide working servers, administrators to maintain those machines, network access to the servers, and so on. Here, we study how the amount of storage a site provides impacts its ability to trade and the number of copies it is able to make.

Another issue that archives must confront is that they may choose trading partners for a number of reasons beyond simply achieving the highest reliability. For example, the libraries of a particular state university system may be directed to cooperate by the university's board of regents. We call such a grouping of sites a trading *cluster*. The cluster may be large enough to serve the needs of its member sites, or sites may need to seek binary *inter-cluster* links

*This material is based upon work supported by the National Science Foundation under Award 9811992.

JCDL'01, June 24-28, 2001, Roanoke, Virginia, USA.

with other archives to expand their trading networks. We examine the ideal cluster size as well as the number of inter-cluster links that must be formed to compensate for a too-small trading cluster.

A site also may have to deal with trading partners that are more or less reliable than itself. For example, a very reliable site must decide whether to trade with all archives or only with those that also have high reliability. We examine these issues to determine how sites can make the best decisions in the face of varying site reliabilities.

Other researchers have examined using redundancy to protect against failures in systems such as RAID [21], replicated file systems [8], and so on. Our work is similar to these systems in that we use replication, we balance resource allocation and high reliability, and we attempt to ensure high data availability.

Unlike these previous systems, our data trading scheme is focused on respecting the differences between individual digital archives, even as these archives cooperate to achieve reliability. Thus, a primary concern of ours is site autonomy. Archivists should be able to decide who they trade with, what types of collections they store and how much storage they provide. Such local decisions are not as important in a system such as RAID, in which a central controller makes all of the decisions. Archives also may have differing reliability goals, such that one archive is willing to expend more resources and expects correspondingly higher reliability in return. It may therefore be important to consider different policies for high and low reliability sites, such that both kinds of sites can protect their data. Similarly, different archives may experience different rates of failure, and an archive may wish to take these failure rates into account when replicating collections. An array of similar components (such as RAID) does not face this issue. Finally, an archivist has unique concerns that are not addressed in traditional systems. It is often important to establish the provenance of collections, and this task is difficult if the collections are moved from site to site frequently or without the archivist's control. An archivist may also wish to keep collections contiguous, so that they can be served to users as a complete unit. Our trading mechanism is flexible enough to address all of these concerns, from autonomy to contiguous collections, while still providing a great deal of protection from failures.

In this paper, we examine how a digital archive can preserve its collections by forming and participating in P2P trading networks. In particular, we make several contributions:

- We present a trading mechanism that can be used by an archive to reliably replicate data. This mechanism is tuned to provide the maximum reliability for the archive's collections, and can be extended if necessary in consideration of individual archivists' needs and goals.

- We identify how to configure an archive for trading by examining the amount of storage that the site should provide and the number of copies of collections a site should try to make.

- We examine the impact of trading with remote partners chosen for political reasons, as opposed to trading with all archive sites. We also discuss the optimal trading network size, and examine when an archivist may wish to seek out additional trading partners.

- We discuss how an archive might trade with sites that have different *site reliabilities*, or rates of failure, by adjusting its trading policies to take these reliabilities into account. We also discuss the importance of accurately estimating the reliabilities of other sites.

In order to evaluate each of these issues, we have used a simulator that conducts simulated trading sessions and reports the resulting reliability. Our concern is primarily in selecting remote sites for storing copies of archived collections. Once trades have been made and collections are distributed, archivists can use other existing systems to detect and recover from failures, enforce security, manage metadata, and so on. Other projects have examined these issues in more detail [4, 22, 17, 23, 19]. It is also possible to enhance our basic techniques to deal with digital objects which change over time, or trades with sites that provide a specialized service (such as storage for a fee). In ongoing work, we are extending our model to provide negotiation for access services (such as search) in addition to storage services. We are also extending our model to deal with trades of other commodites, such as money or processing power, in addition to storage space.

This paper is organized as follows. In Section 2 we discuss the basic trading mechanism, as well as extensions to the basic mechanism for trading networks of digital archives. Section 3 presents evaluations of alternative trading policies using simulation results. Section 4 discusses related work, and in Section 5 we present our conclusions.

2. DATA TRADING

Data trading is a mechanism for replicating data to protect it from failures. In this section, we summarize the techniques used in data trading. We also discuss the extensions and enhancements to data trading that are needed to use the mechanism for digital archives. A full discussion of the basic data trading algorithm, as well as analysis of the tradeoffs involved in tuning the algorithm, is presented elsewhere [5].

2.1 Archival services

Our model of a digital archiving service contains the following concepts:

Archive site: an autonomous provider of an archival storage service. A site will cooperate with other autonomous sites that are under the control of different organizations to achieve data replication. The focus of this paper is the decisions made by a particular archive site; we refer to this site as the *local site*.

Digital collection: a set of related digital material that is managed by an archive site. Examples include issues of a digital journal, geographic information service data, or a collection of technical reports. Although collections may consist of components such as individual documents, we consider the collection to be a single unit for the purposes of replication. Here, we assume that all collections are equally important and require the same effort to preserve.

Archival storage: storage systems used to store digital collections. Some of the storage, called the *public* storage, is dedicated to remote sites that have concluded trades with the local site, and is used to store collections owned by the remote sites. An archive site must decide how much public

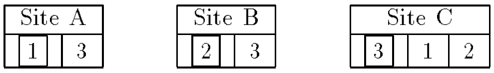

Figure 1: Reliability example.

storage P_{total} to provide. Here, we assume that a site uses a *storage factor* F, such that if the site has N bytes of archived data, it purchases $F \times N$ total storage space. The site uses N bytes of this space to store its own collections, and has $P_{total} = F \times N - N$ extra space to trade away.

Archiving clients: users that deposit collections into the archive, and retrieve archived data. When a client deposits a collection at an archive site, that site is said to "own" the collection, and takes primary responsibility for protecting it.

Trading network: a local site must connect to *remote sites* and propose trades. In the general case, any site can connect to any other site. In a digital archiving domain, it may be more desirable to select a set of "trusted" sites to trade with. This issue is discussed in more detail below.

Automation: The archive should operate as automatically as possible, while allowing librarians or archivists to oversee its operation and adjust its configuration. Thus, an archiving site may automatically replicate copies of a digital collection, but would do so according to the desired goals and constraints important to the administrators.

These concepts are used to study the properties of a trading network primarily concerned with protecting the data itself from loss. While we do not consider other archival concerns (such as preserving access or establishing chain of custody) for simplicity, our model can be extended to deal with such concerns.

Each archive site can fail (lose data), and we model the possibility of failures as a *site reliability*: a number which indicates the probability that the site does not experience data loss. Although site reliabilities may change over time, here we assume for simplicity that reliabilities are constant. Given site reliabilities and a placement of copies of collections at these sites, we can calculate two values:

- *global data reliability*: the probability that no collection owned by any site is lost.
- *local data reliability*: the probability that no collection owned by a particular site is lost.

Thus, global data reliability measures the success of the trading mechanism as a whole, while local data reliability measures the success of decisions made by an individual site participating in data trading. For example, consider Figure 1. This figure shows three sites, each of which owns one collection (shown boxed), while storing copies of collections owned by other sites. Let us assume that the site reliability of each site is 0.9, that is, each site has a ten percent chance of experiencing data loss. In one possible scenario, sites B and C fail but site A does not, while in another scenario, none of the sites fail. We can calculate *global data reliability* by examining all possible scenarios of sites failing or surviving; in this case, there are eight possibilities. For each scenario, we assign the score "0" if at least one collection is lost, or "1" if no data is lost. Thus, in the scenario where sites B and C fail, collection 2 is lost and we assign the score 0. We then weight each score by the probability of the scenario; the situation where B and C fail but A does not will occur with probability $0.1 \times 0.1 \times 0.9 = 0.009$. Finally, we sum the weighted scores to find the expected probability of data loss. The distribution of collections shown in Figure 1 has a global reliability of 0.981, indicating that there is less than a two percent chance of data loss.

We can calculate *local data reliability* in much the same way, except that we only consider the collections owned by a particular site when assigning scores. For example, if we wish to calculate the local reliability of site A, we examine all possible scenarios, and assign a score of "0" if collection 1 is lost, or "1" if collection 1 is preserved. In this way, we can calculate the local data reliability of site A and site B to be 0.99, while the local data reliability of site C is 0.999. Site C enjoys higher data reliability because it has made more copies of its collection.

We can interpret local and global data reliabilities as the probability that data will not be lost within a particular interval, say, one year. Then, we can calculate the expected number of years before data is lost, known as the *mean time to failure* (MTTF). An increase in reliability from 0.99 to 0.999 actually represents an increase in the MTTF from 100 years to 1000 years. Because MTTF better illustrates the results of a particular policy by giving an indication of how long data will be protected, we report our simulation results in Section 3 in terms of MTTF.

In this paper we are primarily concerned about evaluating the choices made by individual sites, and preserving the autonomy of those sites. Therefore, we will examine data trading from the perspective of local data MTTF. In previous work [5] we have assumed that all sites have the same probability of failure, but here we consider the possibility that different sites have different reliabilities.

2.1.1 The trading network

There are two reasons why a local site may choose a particular remote site as a P2P trading partner. First, the remote site may have a reputation for high reliability. Second, there may be political or social factors that bring several autonomous archives together. An archive must make trades that take both reliability and politics into account.

We refer to the set of potential trading partners for a local site as that site's *trading network*. In our previous work we have assumed that a site's trading network includes all other archive sites. However, a local site may participate in one or more *clusters*, or sites that have agreed to form partnerships for political, social or economic reasons. For example, all of the University of California libraries may join together in one cluster. A local site may also have individual *inter-cluster* links for political or reliability reasons. If an archive at say MIT is well known for high reliability, one of the University of California libraries may form a partnership with MIT in addition to the California cluster. Once a site has found trading partners, it can continue to consider politics and reliability when proposing trades. In Section 2.2 we discuss how a site can use site reliabilities to select sites for individual trades.

There are two challenges that face a site when it is constructing a trading network. The first challenge is deciding how many sites should be in the network, and what inter-cluster partnerships to form. The second challenge in constructing a trading network is estimating the site reliabilities of other sites. One possible method is to examine the past behavior of the site. Sites with many failures are likely to have more failures in the future, and are assigned a lower site reliability than sites that rarely fail. Another method is to examine components of the archive's storage mechanism [6]. Sites that use disks that are known to be reliable or security measures that have successfully protected against hackers should be given a higher site reliability. A third possibility is to use the reputation of the site or institution hosting the site. Thus, even the perceived reliability of a site can be influenced by political or social factors.

We evaluate the ideal size for trading clusters, and give guidelines for how many inter-cluster partnerships should be formed in Section 3. We also examine the impact of site reliability estimates in that section.

2.2 Conducting trades

When a client deposits a collection at an archive site, the site should automatically replicate this collection to other sites in the trading network. This is done by contacting these sites and proposing a trade. For example, if site A is given a collection of digital journals, site A will then contact other sites and propose to give away some of its local archival storage to a site willing to store a copy of the journals.

We have developed a series of steps for conducting trades in previous work [5]. These steps are summarized in the DEED_TRADING algorithm shown in Figure 2. This is a distributed algorithm, run by each site individually without requiring central coordination. A deed represents the right of a local site to use space at a remote site. Deeds can be used to store collections, kept for future use, transferred to other sites that need them, or split into smaller deeds. When a local site wants to replicate a collection, it requests from a remote site a deed large enough to store the collection. If the remote site accepts, the local site compensates the remote site with a deed to the local site's space. In the simplest case, the deed that the local site gives to the remote site is equal to the deed that the remote site gives to the local site. There are other possibilities; see below.

Several details of the DEED_TRADING algorithm can be tuned to provide the highest reliability:

<**S**>: The *trading strategy* <S> dictates the order in which other sites in the trading network will be contacted and offered trades. The best strategy is for a site to trade again with the same archives it has traded with before. This is called the *clustering* strategy, because a site tries to cluster its collections in the fewest number of remote sites. If there are several sites that have been traded with before, the local site selects the remote site holding the largest number of the local site's collections. If there is still a tie, or if there are no previous partners, the local site chooses a remote site randomly. For the special case where sites have small storage factors (e.g. $F = 2$), the *best fit* strategy is best. Under best fit, the remote site with the smallest advertised free space is chosen. In [5] we examine several other strategies, such as *worst fit*, where the site with the most advertised free space is preferred. If different sites have different reliabilities, as we assume in this paper, it is possible to adjust the strategy to reflect those reliabilities; see below.

<**A**>: A site must decide how much of its storage space to offer for trades. The best advertising policy <A> is the *data-proportional* policy, where a site advertises some multiple y of the total amount of data N owned by the site. If the amount of remotely owned data stored so far is P_{used}, and the amount of free public space is P_{free}, then the advertised amount is:

$$MIN(N \times y - P_{used}, P_{free})$$

Thus, the amount of advertised space is the total amount of "available" public space minus the amount of public space used so far, except that a site cannot advertise more public space than it has free. Our experiments show that the best setting for y is $y = F - 1$, where F is the site's archival storage factor (see Section 2.1).

<**U**>: If a local site has a deed for a remote site, it can use that deed to make a copy of any collections that fit in the deed but do not already exist at the remote site. A site must decide when to use a deed that it holds to make more copies of collections. The *aggressive* deed use policy, which provides the highest reliability, dictates that a site will use its deeds to replicate as many collections as possible, in order of rareness. Thus, a site holding a deed will use it to replicate its "rarest" collection (the collection with the fewest copies) first. If some of the deed is left over, the site will make a copy of the next rarest collection, and so on. These collections are replicated even if they have already met the replication goal <G>.

<**R**>: If a site is unable to make <G> copies of a collection C_L, it can try to trade again in the future to replicate the collection. The *active retries* policy says that a site will not wait to be contacted by other sites to make copies of C_L, but instead will run DEED_TRADING again after some interval to replicate C_L. A site must choose an appropriate event to trigger the retry; for example, the site may wait one week before trying again.

DEED_TRADING also uses the following policies, which are investigated in this paper:

<**G**>: A site tries to make <G> copies of a collection. Once this target is met, the site does not have to make any more trades. Appropriate values of <G> are discussed in Section 3.

<**D**>: The deed that L gives to R may or may not be the same size as the deed that R gives to L. In our previous work, we have assumed that the two deeds were of equal size. Here, we investigate the possibility that the deed size is influenced by the site's reliability. This issue is discussed below.

2.2.1 *Adapting trading policies for differing site reliabilities*

We can extend the basic trading framework presented in [5] (summarized above) to allow a local site to use the estimated reliabilities of its partners in order to make good trading decisions. There are two aspects of DEED_TRADING that could be modified based on site reliabilities: the trading strategy <S>, and the deed size policy <D>.

One way to change the trading strategy <S> is to look only at site reliabilities when making trades. In the *highest reliability* strategy, a site seeks to trade with partners that have the best reliability. The idea is to make trades that will best protect the local site's collections. In contrast, the *lowest reliability* strategy seeks out sites with the worst reliability. Although each trade may be less beneficial, the low reliability sites may be more desperate to trade than high reliability sites, meaning that the local site can make more copies of its collections. Finally, the *closest reliability* strategy seeks to find the sites with reliability closest to the local site's; the local site must then estimate its own reliability.

I. The local site L repeats the following until it has made <G> copies of collection C_L, or until all sites in the trading network have been contacted and offered trades:

1. Select a proposed deed size $D_L = size(C_L)$.
2. Select a remote site R in the trading network according to the trading strategy <S>.
3. If L has a deed for R then:
   (a) If the deed is large enough to store C_L, then use the deed to make a copy of C_L at R. Return to step I.
   (b) Otherwise, set $D_L = D_L - size$(existing deed).
4. Contact R and ask it to advertise its free space <A>$_R$.
5. If <A>$_R$ $< D_L$ then:
   (a) Contact sites holding deeds for R. Give those sites deeds for local space (at L) in return for the deeds for R. Add these deeds to the existing deed L holds for R. Adjust D_L downward by the total amount of the newly acquired deeds.
   (b) If L cannot obtain enough deeds this way, then it cannot trade with R, and returns to step I.
6. R selects a deed size D_R according to the deed size policy <D>.
7. If L's advertised free space <A>$_L$ $< D_R$, the trade cannot be completed. Return to step I.
8. The trade is executed, with L acquiring a deed of size D_L for R's space, and R acquiring a deed of size D_R for L's space.
9. L uses its deeds for size R to store a copy of C_L.

II. If the goal of <G> copies for C_L is not met, L can try this process again at some point in the future, according to the retry policy <R>.

III. At any time a site may use a deed that it posesses to replicate its collections, according to its deed use policy <U>.

Figure 2: The DEED_TRADING algorithm.

Another way to change the trading strategy is to use site reliabilities in combination with other factors. In the clustering strategy, the local site chooses the remote site holding the most copies of collections owned by the local site. In the *weighted clustering* strategy, the local site weights the number of collections by the reliability of the site. For example, site A (reliability 0.5) might hold three collections while site B (reliability 0.9) might hold two collections. We consider the partnership value of site A to be $0.5 \times 3 = 1.5$, while the partnership value of site B is $0.9 \times 2 = 1.8$; thus, site B is chosen. Other strategies could be weighted in a similar manner. In the case of best fit and worst fit, we can multiply the advertised space by the site's reliability, and use the weighted value in the best fit or worst fit calculations. In this way, we are calculating the "expected" amount of space at the remote site based on the probability that the space will actually be available.

The deed size policy <D> can use reliabilities to encourage a "fair" trade between sites. Under the (previously studied) *same size* policy, the local site and remote site exchange deeds that are the same size. However, if the reliabilities of the two sites differ, then a deed for the more reliable site may be considered "more valuable," and the less reliable site will have to give a larger deed to compensate. We can denote the site reliability of site i as P_i, and the size of the deed that the site gives in trade as D_i. Then, we can calculate the *reliability-weighted* value of the deed as $P_i \times D_i$. The *weighted size* policy dictates that the reliability-weighted values of the exchanged deeds must be equal, e.g. if the local site L trades with the remote site R then $P_L \times D_L = P_R \times D_R$. The local site chooses a deed size based on the collection it wants to replicate, so the size of the deed that the remote site must give in return is $D_R = (P_L \times D_L)/P_R$.

A local site must be able to estimate the site reliability of its trading partners (and possibly itself) in order to make decisions which take reliability into account. We can denote site i's estimate of site j's reliability as $P_{i,j}$. In an ideal situation, each site could calculate reliabilities exactly, such that $P_{i,j} = P_j$. However, it is difficult to predict which sites will fail, and thus reliability estimates may be innaccurate. A local site can use information about a remote site's reputation, any previous failures, and the reliability of the storage components to estimate the reliability. Thus, it is likely that sites which are in fact highly reliable are known to be reliable, while more failure prone sites are known to be less reliable. In other words, $P_{i,j} \approx P_j$.

In Section 3.3 we examine the reliability resulting from trading strategies that account for reliability and the impact of the same size and weighted size policies. We also examine the effects of innaccurately estimating site reliabilities.

3. RESULTS

3.1 The data trading simulator

In order to evaluate the decisions that a local site must make when trading, we have developed a simulation system. This system conducts a series of simulated trades, and the resulting local data reliabilities are then calculated. Table 1 lists the key variables in the simulation and the initial base values we used; these variables are described below.

The simulator generates a *trading scenario*, which contains a set of sites, each of which has a quantity of archival storage space as well as a number of collections "owned" by

Variable	*Description*	*Base values*
S	Number of sites	2 to 15
F	Site storage factor	2 to 7
P_{MIN}, P_{MAX}	Min/max site reliability	$P_{MIN} = 0.5$ or 0.8 $P_{MAX} = 0.99$
P_{est}	P_i estimate interval	0 to 0.4
$CperS_{MIN}$, $CperS_{MAX}$	Min/max collections per site	$CperS_{MIN} = 4$, $CperS_{MAX} = 25$
$Csize_{MIN}$, $Csize_{MAX}$	Min/max collection size	$Csize_{MIN} = 50$ Gb, $Csize_{MAX} = 1000$ Gb
$Ctot$	Total data at a site	$Ctot_{MIN}$ to $Ctot_{MAX}$
$Ctot_{MIN}$, $Ctot_{MAX}$	Min/max value of $Ctot$	$Ctot_{MIN} = 200$ Gb, $Ctot_{MAX} = 10,000$ Gb
<G>	Replication goal	2-15 copies
<S>	Trading strategy	9 strategies tried
<D>	Deed size policy	*same size* and *weighted size*

Table 1: Simulation variables.

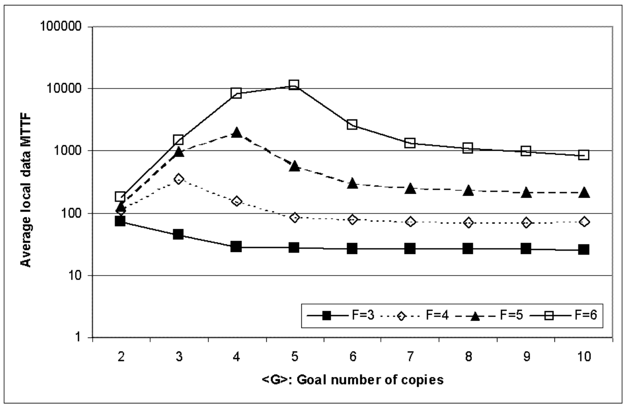

Figure 3: The trading goal and storage capacity.

the site. The number of sites S is specified as an input to the simulation. The number of collections assigned to a site is randomly chosen between $CperS_{MIN}$ and $CperS_{MAX}$, and the collections assigned to a site all have different, randomly chosen sizes between $Csize_{MIN}$ and $Csize_{MAX}$. The sum of the sizes of all of the collections assigned to a site is the *total data size* $Ctot$ of that site, and ranges from $Ctot_{MIN}$ to $Ctot_{MAX}$. The values we chose for these variables represent a highly diverse trading network with small and large collections and sites with small or large amounts of data. Thus, it is not the absolute values but instead the range of values that are important.

The archival storage space assigned to the site is the storage factor F of the site multiplied by the $Ctot$ at the site. In our experiments, the values of F at different sites are highly correlated (even though the total amount of space differs from site to site). By making all sites have the same F, we can clearly identify trends that depend on the ratio of storage space to data. Therefore, we might test the reliability that results from a particular policy when all sites use $F = 2$. In this case, one site might have 400 Gb of data and 800 Gb of space, while another site might have 900 Gb of data and 1800 Gb of space. The scenario also contains a random order in which collections are created and archived. The simulation considers each collection in this order, and the "owning" site replicates the collection. A site is considered "born" when the first of its collections is archived. A site does not have advance knowledge about the creation of other sites or collections. Our results represent 200 different scenarios for each experiment.

We model site failures by specifying a value P_i: the probability that site i will not fail. This value reflects not only the reliability of the hardware that stores data, but also other factors such as bankruptcy, viruses, hackers, users who accidentally delete data, and so on. In our experiments, we consider the situation where all sites are relatively reliable (e.g. $0.8 \leq P_i \leq 0.99$) as well as the case where some sites are quite unreliable (e.g. $0.5 \leq P_i \leq 0.99$). To consider site reliability estimates, we assume that site i's estimate $P_{i,j}$ of site j's reliability is randomly chosen in the range $P_j \pm P_{est}$.

3.2 Local configuration issues

An archive site should have enough space to store the collections deposited by local clients. In order to participate in data trading, a site also needs extra *public* storage space that it trades away. We call the ratio of total space to locally owned collections the *storage factor* F. In this section we examine the best value of F, which indicates the appropriate amount of extra storage a site must provide.

A related issue is the number of copies of collections that a site will attempt to make. If more copies are made, higher reliability results. However, remote sites must have enough storage to hold all of the copies, and the local site must have enough public storage space to trade away to make these copies. In other words, the *goal* <G> number of copies is related to the storage factor F.

To examine the relationship between <G> and F, we tested a situation where 15 archive sites replicate their collections; each site had a reliability of 0.9. We varied F in the range $2 \leq F \leq 6$ and tested goals from 2 to 15 copies. The results are shown in Figure 3. Note that the vertical axis in this figure has a logarithmic scale, and that there are separate data series for $F = 3, 4, 5$ and 6. As expected, providing more storage increases the local reliability. The best reliability (11,000 years MTTF) is obtained when $F = 6$ and sites try to make five copies. (We are mainly concerned with finding the policy that has the highest reliability, regardless of the actual magnitude of the MTTF value.) Trying to make more copies results in decreased reliability because there is not enough space to make more than five copies of every site's collections. If one site tries to make too many copies, this site uses up much of the available space in the trading network, resulting in decreased reliability for other sites.

Sites may wish to purchase less space than six times the amount of data for economic reasons. Our results show that with $F = 5$ and <G>= 4, sites can achieve 2,000 years MTTF, and with $F = 4$ sites can achieve 360 years MTTF if the goal is three copies. Therefore, while buying a lot of space can provide very high reliability, archives can still protect their data for hundreds of years with a more modest investment.

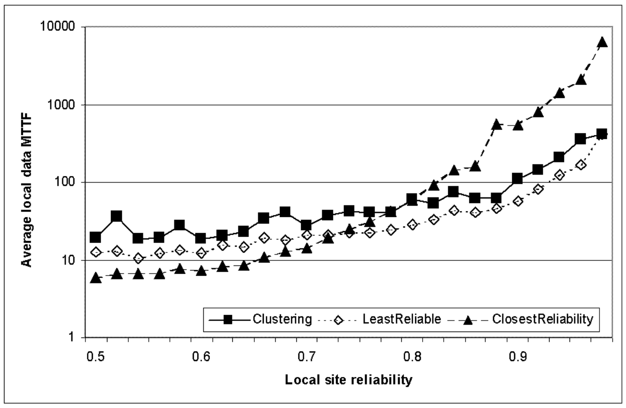

Figure 4: Trading strategies.

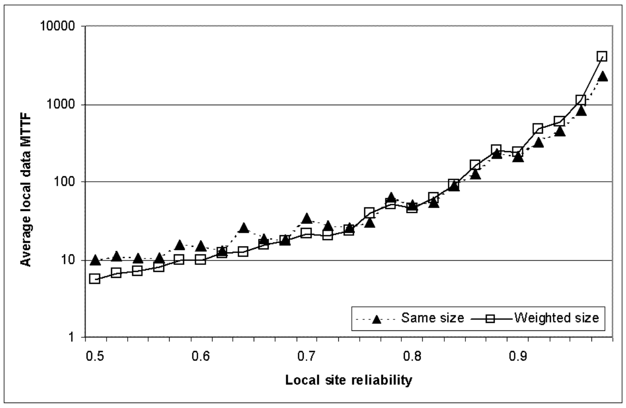

Figure 5: The deed size policy.

3.3 Trading policies that consider reliability

Archive sites can use reliability information about other sites to make trading decisions (Section 2.2). First, we examined trading strategies by running simulations where each site had different reliabilities; site reliabilities were randomly chosen in the range $0.5 \leq P_i \leq 0.99$. In this experiment, there were 15 sites, each with a storage factor of $F = 4$ and a target <G> of three copies. We also assumed (for the moment) that each site was able to predict site reliabilities accurately, so that $P_{i,j} = P_j$. The results are shown in Figure 4. (For clarity, not all strategies are shown; the omitted strategies are bounded by those in the figure.) Recall that the clustering strategy is to trade again with previous trading partners, the closest reliable strategy is to trade with sites of reliability close to that of the local site, and the least reliable strategy is to prefer the least reliable site. The results indicate that the clustering strategy is best for sites with relatively low reliability, but that sites with $P_i \geq 0.8$ are better off using the closest reliability strategy. For example, a site with $P_i = 0.9$ achieves a local data MTTF of 540 years using closest reliability, versus 110 years MTTF resulting from clustering. These results assume that all sites are using the same strategy. We ran another experiment where the high reliability sites ($P_i \geq 0.8$) used one strategy, but the lower reliability sites used another. These results (not shown) confirm that it is always best for the high reliability sites to use the closest reliable strategy, and for the low reliability sites to use clustering. We ran similar experiments with $0.8 \leq P_i \leq 0.99$, and reached the same conclusions, although the range of high reliability sites that should use closest reliability was $P_i \geq 0.9$.

High reliability sites clearly benefit by trading among themselves, so that every trade they initiate places a copy of a collection at a very reliable site. If low reliability sites were to try to trade only among themselves, they would lose reliability by excluding the benefits of trading with high reliability sites. If low reliability sites were to try to trade preferentially with the high reliability sites (as in the highest reliability strategy), they would quickly find the high reliability sites overloaded. Therefore, the best strategy is to make as many trades as possible in a way that is neutral to the remote sites' reliability, and this is what the clustering strategy does. The high reliability sites will not seek out low reliability sites to make trades, but will *accept* trade offers made by those sites.

In order to use strategies that depend on site reliabilities, a site must be able to estimate the reliabilities of itself and its trading partners. We examined the importance of accuracy in these estimates by allowing the probability estimate interval P_{est} to vary. The failure probability P_i of each site is selected at random from the range $0.5 \leq P_i \leq 0.99$, and sites with $P_i \geq 0.8$ used closest reliability while other sites used clustering. Each local site i's estimate of the remote site j's reliability was randomly chosen in the range $P_j \pm P_{est}$. The results (not shown) indicate that the best reliability results in the ideal case: when the estimates are completely accurate. As long as sites are able to make estimates that are within seven percent of the true value, their local data reliability is quite close to the ideal case. However, as the error increases beyond seven percent, the local data reliability drops. For example, when estimates are innaccurate by 30 percent, archives using closest reliability can only achieve a local MTTF of 200 years, versus 500 in the ideal case. If sites can estimate a site reliability close to the true value, they can usually separate high reliability archives from low reliability archives, and select the high reliability sites for trading. If estimates are very innaccurate (e.g. by 25 percent or more) very high reliability sites (e.g. $P_i \geq 0.94$) achieve better reliability using the clustering strategy. However, moderately reliable sites ($0.8 \leq P_i \leq 0.94$) still achieve better MTTF with the closest reliability strategy.

Another policy that can take site reliabilities into account is the deed size policy <D>. We have compared the weighted size policy with the same size policy in an experiment with 15 sites, where $0.5 \leq P_i \leq 0.99$, the storage factor $F = 4$, and the target <G>= 3. The results are shown in Figure 5. (In this experiment, the high reliability sites, $P_i \geq 0.8$, used the closest reliability strategy, and other sites used clustering.) The figure indicates that the weighted size policy, which considers deeds from reliable sites to be more valuable, is good for high reliability sites ($F \geq 0.8$). For example, a site with $P_i = 0.9$ can achieve 240 years MTTF using the weighted size policy, a 14 percent increase over

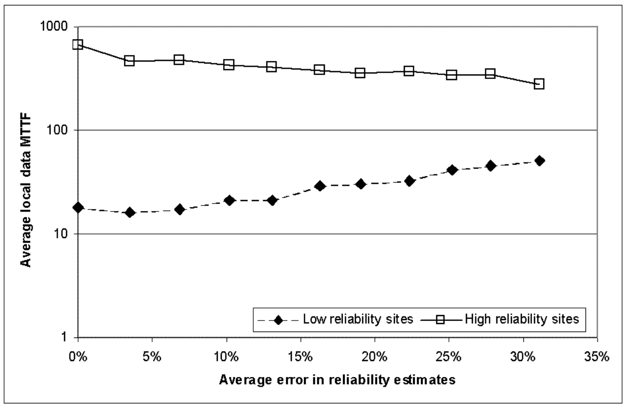

Figure 6: The impact of estimating site reliabilities.

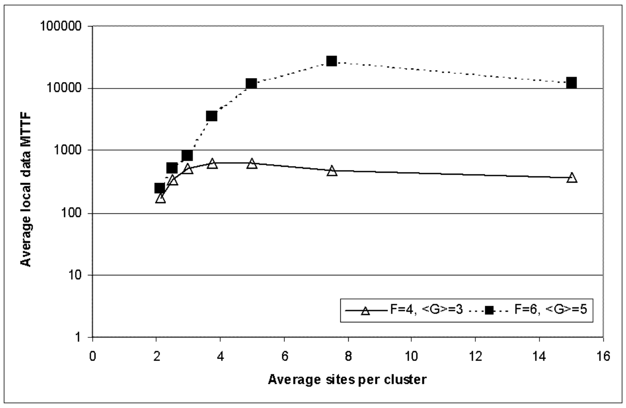

Figure 7: The impact of cluster size.

the same size policy MTTF of 210 years. In contrast, low reliability sites are hurt by the weighted size policy, with as much as a 50 percent decrease in MTTF (from 25 years to 12 years) when $P_i = 0.64$. High reliability sites are the beneficiary of the weighted size policy because they receive more space in trades, and the most reliable sites can demand the most space from other sites. These results indicate that it may be better for low reliability sites to avoid paying the high penalties of the weighted size policy by trading only with other low reliability sites. However, the results (not shown) of another experiment we conducted indicate that it is still better for low reliability sites to try to trade with high reliability archives, even when the weighted size policy is used. If the low reliability sites ignore the high reliability sites by using closest reliability instead of clustering, they experience an average decrease in local data MTTF of 15 percent (from 16 years to 14 years).

Once again, we have examined the effect of estimating reliabilities. Figure 6 shows the impact on local data MTTF versus the accuracy of the estimates. In this experiment, $0.5 \leq P_i \leq 0.99$ and sites estimated reliabilities randomly in the range $P_j \pm P_{est}$ such that a larger P_{est} resulted in a larger average error (shown on the horizontal axis in Figure 6). These results show that high reliability sites suffer when estimates are innacurate, while low reliability sites benefit. This is because a low reliability site can be mistaken for a high reliability site, and thus can get larger deeds from its trading partners. Similarly, high reliability sites can be mistakenly judged to have less reliability, and must accept correspondingly smaller deeds. Nonetheless, most high reliability sites ($0.8 \leq P_i \leq 0.98$) still achieve higher MTTF under the weighted size policy than under the same size policy, even when estimates are as much as 30 percent wrong on average.

In summary, if some archives are more reliable than others:

- Highly reliable sites should trade among themselves. However, if site reliability estimates are off by 25 percent or more, then the clustering strategy is better.

- Less reliable sites should continue to use clustering.

- Highly reliable sites can use the weighted size policy to extract larger deeds from low reliability sites.

- Less reliable sites should try to trade using the same size policy, but should continue to trade with highly reliable sites even if the weighted size policy is used.

3.4 The trading network

In this section, we investigate the ideal trading network size. Specifically, we examine the effects of *clusters*, or groupings of sites that cooperate for political or social reasons. If the cluster is not large enough to serve a site's trading needs, the site will have to seek *inter-cluster* partnerships to expand the trading network. Note that in previous sections, we assumed a local site could potentially trade with any remote site. Even with the clustering strategy, any site was eligible to become a trading partner. In this section we consider the case where clusters are pre-ordained.

In order to determine the ideal cluster size, we ran a simulation in which 15 archive sites were divided into N clusters, where $N = 1, 2...7$. In this experiment, each cluster is *fully isolated*: there are no inter-cluster links. Thus, when $N = 1$ all sites trade with each other, but when $N = 3$ there are three clusters of five sites, and sites trade only within a cluster. We examined the case where $F = 4$ and <G>=3, as well as $F = 6$ and <G>=5. The results are shown in Figure 7. When space is tight ($F = 4$), a cluster of about 5 sites provides the best reliability (with a MTTF of 630 years). In contrast, when there is more space ($F = 6$), then about seven sites is the best cluster size, with a MTTF of 26,000 years. In both cases, larger clusters are actually detrimental, decreasing the local data reliability of the member sites. Large clusters mean that a member site must trade with many other archives, and this can cause some sites to become overloaded; thus their public storage becomes filled up. When this happens, the overloaded sites are less able to make trades, and their reliability suffers. *Therefore it is not necessary or even desirable to form very large clusters in order to achieve reliability.*

If sites participate in trading clusters that are smaller than the ideal size, they can seek inter-cluster partnerships to enhance reliability. We have simulated a situation where 12

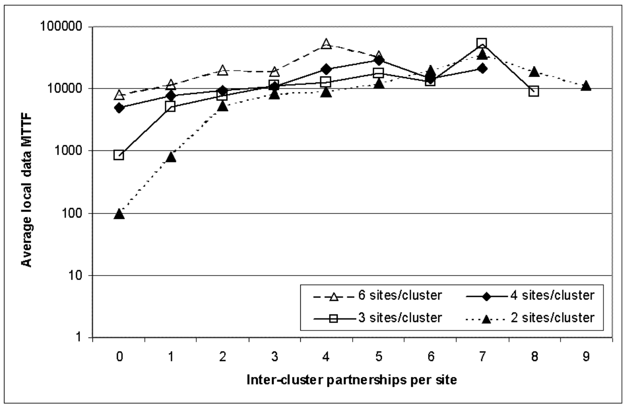

Figure 8: Inter-cluster partnerships, $F = 6$.

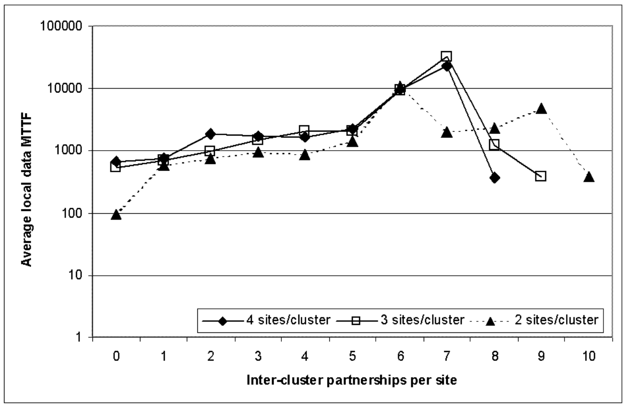

Figure 9: Inter-cluster partnerships, $F = 4$.

sites were divided into small clusters, and each site randomly chose partners outside of its own cluster. Figure 8 shows the results for $F = 6$, where average local data reliability is plotted against the number of inter-cluster partnerships per site. The results show that smaller clusters must seek out many inter-cluster partnerships to achieve the highest reliability. Thus, sites in clusters of three or fewer archives must find roughly seven partners in other clusters, while clusters with four sites should find roughly five additional partners. Even sites in relatively large clusters (e.g. with six sites) can benefit by seeking four inter-cluster partnerships. Seeking too many inter-cluster partners can hurt reliability. A local site may try to find partners outside the cluster, but unless the partners are fully integrated into the cluster, then the local site must field all of the partner's trading requests, and quickly becomes overloaded. Similarly, when $F = 4$, inter-cluster partnerships are beneficial. Our results, shown in Figure 9, indicate that for clusters of less than five sites, six or seven inter-cluster partnerships are needed to achieve the best reliability.

In summary:

- Sites in clusters of about five archives (for $F = 4$) or seven archives (for $F = 6$) achieve the highest reliability.
- Sites in smaller clusters can seek inter-cluster partnerships to improve their reliability.
- If a cluster is too large or if a site has too many inter-cluster partners, reliability can suffer.

4. RELATED WORK

The problems inherent in archiving data are well known in the digital library community [11]. Researchers have confronted issues such as maintaining collection metadata [23, 17], dealing with format obsolescence [25, 19, 14], or enforcing security policies [22]. These efforts complement attempts to simply "preserve the bits" as exemplified by projects like SAV [4], Intermemory [12], LOCKSS [24], or OceanStore [10]. The work we present here can be used to replicate collections in order to best preserve the bits, and can be augmented if necessary (e.g. with a metadata management scheme.)

Many existing data management systems use replication to provide fault tolerance. However, these systems tend to focus on access performance and load balancing [7, 26, 27], whereas we are primarily concerned about reliability. Sites using our clustering strategy attempt to emulate *mirrored disks* [2]. In contrast, database systems tend to prefer a strategy called *chained declustering* [15], which trades some reliability for better load balancing after a failure [18]. Digital archives, which are primarily concerned with preservation, prefer the more reliable mirrored disks; hence, they use the clustering strategy. Moreover, we are concerned with placing archived data that is not likely to change, and therefore are not as concerned as previous researchers with the ability to correctly update distributed replicates [1, 13]. Thus, while a distributed transaction protocol could be added if necessary, efficient or correct updates are less important than preserving the data.

Other systems (such as Coda [16] or Andrew [9]) use replication in the form of *caching*: data is moved to the users to improve availability. Then, if the network partitions, the data is still readable. Our goal is to place data so that it is most reliably stored, perhaps sacrificing short term availability (during network partitions) for long term preservation. Specifically, Andrew and Coda eject data from caches when it is no longer needed. Our scheme assumes that data is never ejected.

The problem of optimally allocating data objects given space constraints is well known in computer science. Distributed bin packing problems [20] and the File Allocation Problem [3] are known to be NP-hard. Trading provides a flexible and efficient way of achieving high reliability, without the difficulties of finding an optimal configuration.

5. CONCLUSIONS

In this paper, we have examined how archives can use and extend peer-to-peer data trading algorithms to serve their data preservation needs. This provides a reliable storage layer that can be enhanced with other services (such as format migration or authenticity verification) to create a complete archiving solution. In order to trade effectively, a site must make certain policy decisions. We have provided guidelines for selecting the amount of storage a local site must provide. We have presented and evaluated trading policies that exploit site reliability estimates, significantly improving reliability. In particular, we have shown that high reliability sites should trade amongst themselves, while low reliability sites should try to trade their collections using the clustering strategy. Finally, we have examined the impact of trading clusters shaped by political and social concerns, and how many extra trading partners a member of such a cluster must find to achieve the highest reliability.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments.

6. REFERENCES

[1] F. B. Bastani and I-Ling Yen. A fault tolerant replicated storage system. In *Proc. ICDE*, May 1987.

[2] Andrea Borr. Transaction monitoring in Encompass [TM]: Reliable distributed transaction processing. In *Proc. 7th VLDB*, September 1981.

[3] W. W. Chu. Multiple file allocation in a multiple computer system. *IEEE Transactions on Computing*, C-18(10):885–889, October 1969.

[4] Brian Cooper, Arturo Crespo, and Hector Garcia-Molina. Implementing a reliable digital object archive. In *Proc. ECDL*, pages 128–143, September 2000. In LNCS vol. 1923.

[5] Brian Cooper and Hector Garcia-Molina. Peer to peer data trading to preserve information. http://dbpubs.stanford.edu/pub/2001-7, 2001. Technical Report.

[6] Arturo Crespo and Hector Garcia-Molina. Modeling archival repositories for digital libraries. In *Proc. ECDL*, pages 190–205, September 2000. In LNCS vol. 1923.

[7] Xiaolin Du and Fred Maryanski. Data allocation in a dynamically reconfigurable environment. In *Proc. ICDE*, February 1988.

[8] Barbara Liskov et al. Replication in the Harp file system. In *Proc. 13th ACM SOSP*, October 1991.

[9] J. H. Morris et al. Andrew: A distributed personal computing environment. *CACM*, 29(3):184–201, March 1986.

[10] John Kubiatowicz et al. OceanStore: An architecture for global-scale persistent storage. In *Proc. ACM ASPLOS*, November 2000.

[11] John Garrett and Donald Waters. Preserving digital information: Report of the Task Force on Archiving of Digital Information, May 1996. Accessible at http://www.rlg.org/ArchTF/.

[12] Andrew Goldberg and Peter Yianilos. Towards an archival intermemory. In *Proc. ADL*, 1998.

[13] Jim Gray, Pat Helland, Patrick O'Neal, and Dennis Shasha. The dangers of replication and a solution. In *Proc. ACM SIGMOD*, June 1996.

[14] Alan Heminger and Steven Robertson. Digital Rosetta Stone: A conceptual model for maintaining long-term access to digital documents. In *Proc. 6th DELOS Workshop on Preservation of Digital Information*, June 1998.

[15] Hui-I Hsiao and Devid DeWitt. Chained declustering: A new availability strategy for multiprocessor database machines. In *Proc. ICDE*, February 1990.

[16] J. J. Kistler and M. Satyanarayanan. Disconnected operation in the coda file system. *ACM TOCS*, 10(1):3–25, February 1992.

[17] Carl Lagoze, Jane Hunter, and Dan Brickley. An event-aware model for metadata interoperability. In *Proc. ECDL*, September 2000. In LNCS vol. 1923.

[18] Edward Lee and Chadramohan Thekkath. Petal: Distributed virtual disks. In *Proc. 7th ACM ASPLOS*, October 1996.

[19] Nuno Maria, Pedro Gaspar, Antonio Ferreira, and Mario Silva. Information preservation in ARIADNE. In *Proc. 6th DELOS Workshop on Preservation of Digital Information*, June 1998.

[20] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. J. Wiley and Sons, Chichester, New York, 1990.

[21] David Patterson, Garth Gibson, and Randy H. Katz. A case for redundant arrays of inexpensive disks (RAID). *SIGMOD Record*, 17(3):109–116, September 1988.

[22] Sandra Payette and Carl Lagoze. Policy-carrying, policy-enforcing digital objects. In *Proc. ECDL*, September 2000. In LNCS vol. 1923.

[23] Arcot Rajasekar, Richard Marciano, and Reagan Moore. Collection-based persistent archives. In *Proceedings of the 16th IEEE Symposium on Mass Storage Systems*, March 1999.

[24] David S. H. Rosenthal and Vicky Reich. Permanent web publishing. In *Proc. USENIX Annual Technical Conference*, June 2000.

[25] Jeff Rothenberg. Ensuring the longevity of digital documents. *Scientific American*, 272(1):24–29, January 1995.

[26] Harjinder Sandhu and Songnian Zhou. Cluster-based file replication in large-scale distributed systems. In *Proc. ACM SIGMETRICS*, June 1992.

[27] Ouri Wolfson, Sushil Jajodia, and Yixiu Huang. An adaptive data replication algorithm. *ACM TODS*, 2(2):255–314, June 1997.