Allocating Resources in Storage Cooperatives with Pseudo-Currencies *

David A. Turner † Daniel Havey ‡ John Ewart§

January 25, 2003

Abstract

Storing multiple copies of data on widely distributed nodes is a strategy to increase availability and survivability of data. To reduce the cost of distributing redundant data, researchers have been investigating the use of storage cooperatives in which members share bandwidth and storage resources. We present the foundations of a simple, extensible economy in which network entities share storage and bandwidth resources by buying and selling data storage contracts with currencies created by members of the cooperative. Methods are developed to protect users from untrustworthy members, so that trading relationships can develop between anonymous users. Our work emphasizes the development of a simple web service interface that encourages multiple implementations within multiple application contexts.

1 Introduction

Several systems have been proposed to enable independent network entities to form peer-to-peer networks to share bandwidth and storage resources. In these cooperatives, peers contribute storage and bandwidth resources in exchange for the storage and bandwidth resources of other peers. In [1], [2] and [3], members form a storage cooperative by joining an underlying peer-to-peer network through which operations and data are routed. These systems assume that members faithfully adhere to a single global control algorithm. In contrast, we specify a minimal set of messages that members use to communicate, and encourage individual nodes to provide their own decision-making algorithms, including the decision of which other nodes will store replicas of its data.

It is relevant to note that peer-to-peer file sharing networks such as Kazaa, Morpheus, etc. were designed as systems that were globally controlled by a single algorithm. However, alternative implementations are now being incorporated into these communities. For example, there is now a "Kazaa crack" that enables users to circumvent rules established by the initial implementation.

There is a substantial body of work on the application of economic models to resource allocation problems in computer systems, which can is described in [6]. There have also been a number of companies that have tried to utilize pseudo currencies to the problem of resource allocation in resource sharing cooperatives. For example, the now defunct company called Mojonation tried to build a

*The support of the National Science Foundation under award 9810708 is gratefully acknowledged.

†Cal. State Univ. San Bernardino, dturner@csusb.edu

‡Cal. State Univ. San Bernardino, dhavey@yahoo.com

§Cal. State Univ. Stanislaus, john@unixninjas.org

Napster-like file sharing system in which resource allocation is controlled with a pseudo-currency. Our proposal differs from all these systems in that we define a simple mechanism by which members can optionally manage their own currencies to be used generally in the cooperative. We also define simple yet sufficient mechanisms to ensure reliability and privacy in the midst of untrustworthy or hostile members.

In contrast to systems with a global control algorithm, the authors of [4] propose a system of autonomous agents in which members trade equal amounts of storage space. Thus, local control involves the decision of whether or not to store data with a given node. The work in [5] provides the possibility for more elaborate local control by defining a bidding mechanism through which members exchange potentially unequal storage resources. However, barter is not easily adapted to the problem of exchanging different types of services, and thus does not provide an evolutionary path for a cooperative to enhance its effectiveness through additional services. For example, a lookup service is needed for members to locate the interfaces of other members. The possibility of collecting service fees motivates members to provide and advertise such a lookup service.

The goal of this work is to lay the foundation for an evolving storage cooperative comprised of potentially hostile autonomous agents. We define an initial set of member-provided services that are essential for such a cooperative, and we describe the security mechanisms that protect members from hostile acts. In Section2, we provide an overview of the system. In Section3 that follows this, we describe the minimal set of necessary web services. We conclude in Section4 with a summary of future work.

2 Overview

Members buy and sell retrieval options, which grant to the buyer the right to retrieve byte ranges of a stored object from the seller until an expiration deadline. Two prices are involved: a purchase price to pay for the option, and an exercise price to pay in the event the buyer wishes to retrieve data before the expiration deadline of the option. This two stage pricing scheme permits sellers to charge separately for the consumption of storage and bandwidth resources. The purchase price of the option reflects the cost of storing the data until expiration plus the cost of transporting the data to the seller. The exercise price reflects the cost of transporting the data to the buyer if she were to request it. In the case that the buyer is paying to extend the deadline of an existing option, no data transmission is needed, and thus the purchase price to extend the option reflects only the consumption of storage needed until the new expiration deadline.

Inability of a buyer to obtain her data by exercising a retrieval option can be due to many factors, including network failure, failure of the seller's hardware, intentional sabotage by the seller, temporary unavailability of the seller's interface, etc. Thus, storage services obtained from the cooperative are inherently unreliable. However, members can improve service reliability in two fundamental ways: data replication and quality assessment. With data replication, members store multiple copies of data with several different members. With quality assessment, members continuously and frequently apply quality test procedures to assess the reliability of service provided by other members. Another member establishes trust when he consistently passes quality assessment testing. Members with high reliability ratings will naturally charge more for their services, and un-trusted members will be

able to obtain higher prices as they slowly establish trust by passing reliability tests over time.

Members have the choice of creating their own money or using money created by another member. A member holds currency units denominated by another member by being the owner of an account managed by the other member. The currency provider exposes an interface through which an account owner transfers currency units from her account into the accounts of other members. Currency providers control their own money supply.

Members of the Internet Storage Cooperative communicate by exchanging XML documents over a secure channel. The secure channel is established with procedures that make use of members' public/private key pairs. The public key also functions as the member's identity, although users will define locally unique aliases for members with whom they interact.

Members do not need public key certificates, because trust is established through ongoing quality assurance procedures. There is no need for an authority to vouch for the identity of another member, because a member's identity is equivalent to her public key. Contracts do not need to be signed by keys bound to individuals, because coop contracts are not meant to be legally binding. The cooperative does not use the fear of litigation as an incentive for members to fulfill their promises. Instead, members try to attain high reliability ratings in order to receive higher prices for services they provide. In this manner, they generate more cash to purchase quality services for themselves.

3 Web services

Each member exposes a web service interface through which messages are transported. We define an initial set of messages that represent core functionality required for a minimal system, which include three messages related to currencies and three related to options. Enhanced functionality, such as a lookup service, comprises the extended interface. New messages can easily be added to allow new services or trading mechanisms.

The messages related to currency include: transferFunds, notifyTransfer, and inquireFunds. Member A transfers to B money denominated by C by sending a TransferFunds message to the web service of C. When this occurs, C notifies B of the transfer by sending a notifyTransfer message to B's web service. A member A can find out the amount of funds she holds in the currency of B by sending an inquireFunds message into the web service of B.

The messages related to options include: proposeOption, exerciseOption, and inquireOptions. When A wants to purchase a retrieval option from B, she sends a proposeOption message into the web service of B. If B accepts the proposal, A sends the data to B, and then transfers the agreed price to B. Member A also uses the proposeOption message to purchase an extension to an existing retrieval option, or to negotiate a refund in exchange for canceling an existing option. If A wants to retrieve data from B, she sends an exerciseOption message to B's web service. Member B delivers the requested bytes, and A transfers the exercise price to B. The exerciseOption is also used for quality assurance testing. To test that B actually as access to the data, A exercises her option to retrieve a small number of bytes at a random offset in the data. To test that B would provide a whole object on request, A would exercise her retrieval option for the entire object. The message inquireOptions is used for reconstructing lost data. Member A would send an inquiresOptions message to B to obtain a list of option con-

tracts that A has for data stored by B. To guard against a ransom threat, A would need to simulate system reconstruction by invoking the inquiresOptions operation followed by a complete retrieval of all her data.

In addition to the core interface, the cooperative needs a lookup service to map member identifiers (public keys) to the network location of the member's interface. Members behind static IP addresses can operate lookup services. The two messages needed for the lookup service include registerPresence and lookupMember.

That all services in the cooperative may command a fee ensures that these services are made available. Therefore, it is expected that members operating currencies will charge a transaction fee when the transferFunds operation is invoked. Also, members providing the lookup service will similarly charge a lookup fee.

4 Future directions

The Internet Storage Cooperative [7] aims to provide reference implementations of the secure communication channel, core services and the extended lookup service. This work is nearing completion. Experimentation is needed to determine methods by which users can specify decision-making policies that the underlying agent can execute.

The most obvious application of the system is for data backup that can survive natural and manmade disasters. However, there are also other possible applications, including the elimination of temporary service outages for mobile devices that rely on a single remote storage system. Research is needed to uncover and apply the technology to these other application domains.

If a member is participating in the cooperative to backup data, then she will need a survivable version of her retrieval options and data decryption keys. This data will therefore need to be stored with other members in the same manner as other data. This is easily done with the core web services defined, however, the user will need to make a record of her public and private keys, and the ids or web addresses of the members storing the option records and data decryption keys, so that she may provide these to recovery software in the event that she experiences local data loss. More research is needed to determine reasonable methods by which users can securely record such essential information needed to recover from catastrophic loss.

References

[1] F. Dabek, M.F. Kaashoek, D. Karger, R. Morris and I. Stoica. Wide-area cooperative storage with CFS. In proc. ACM SOSP'01, Banff, Canada, Oct 2001.

[2] P. Druschel and A. Rowstron. PAST: A large-scale, persistent peer-to-peer storage utility. In Proc. HotOS VIII, Schloss Elmau, Germany, May 2001.

[3] J. Kubiatowicz, et al. OceanStore: An Architecture for Global-Scale Persistent Storage. In Proc. Ninth International Conference on Architectural Support for Programming Lanugages and Operating Systems (ASPLOS 2001), Nov 2000.

[4] S. Elnikety, M. Lillibridge, M. Burrows. Peer-to-peer Cooperative Backup System. WiPs paper in USENIX FAST 2002, January 2002, Monterey, CA.

[5] Brian F. Cooper and Hector Garcia-Molina. Bidding for storage space in a peer-to-peer data preservation system. International Conference on Distributed Computing Systems 2002.

[6] D. Ferguson, C. Nikolaou, J. Sairamesh, Y. Yemini. Economic Models for Allocating Resources in Computer Systems. In Market-Based Control: A Paradigm for Distributed Resource Allocation, Scott Clearwater (ed.), World Scientific Press, 1996.

[7] Internet Storage Cooperative. http://iscoop.org