Replica Management Should Be A Game[1]

Dennis Geels and John Kubiatowicz
University of California
Berkeley, CA 94720 USA
{geels,kubitron}@cs.berkeley.edu

Abstract

We believe that large-scale replica management solutions should be based on an economic model. In this paper, we discuss the benefits provided by an economic approach and outline important directions for future research.

1. Introduction

As demand for information increases, centralized servers become a bottleneck. Content providers cope by distributing *replicas* of their files to servers scattered throughout the network. The replicas then respond to local client requests, reducing the load on the central server. *Replica Management* refers to the problem of deciding how many replicas of each file to distribute, and where to place them.

In a perfect system, replicas are placed near the clients that access them. Shrinking network distance decreases access latency and sensitivity to congestion and outages.

Also, exactly enough replicas should exist to handle the cumulative demand for each file. With too few replicas, servers become overloaded, and clients see reduced performance. Conversely, extra replicas waste bandwidth and storage that could be reassigned to other files, as well as the money spent to rent, power, and cool the host machine.

Replica management alternatives Several approaches to replica management have been developed. One solution, perhaps best embodied by Content Distribution Networks (CDNs)[1, 4, 20], involves deploying new machines throughout the network. These machines only host replicas of their company's content.

Peer-to-Peer storage systems, including FreeNet[6], Gnutella[2], and many research prototypes (e.g. Bayou[9] and CFS[8]), consist of independently owned and operated machines. Each machine controls its own set of replicas, but freely stores and serves content produced elsewhere. Limited resources are usually handled with a simple cache algorithm such as LRU.

A third approach, sharing many P2P characteristics, applies concepts from Economics to the replica management problem[3, 12, 21, 23]. Here, machines earn (real or virtual) money by hosting replicas and use that money to purchase access to replicas hosted by other machines.

Replica management economies In these economic systems, individual machines are *autonomous*–free to choose which replicas they host. They may make such decisions using simple on-demand algorithms or complicated predictive methods. In fact, each could use a different algorithm. Together, payment-based cooperation and ambivalence towards local server algorithms are the defining characteristics of a *Replica Management Economy*, or *RME*.

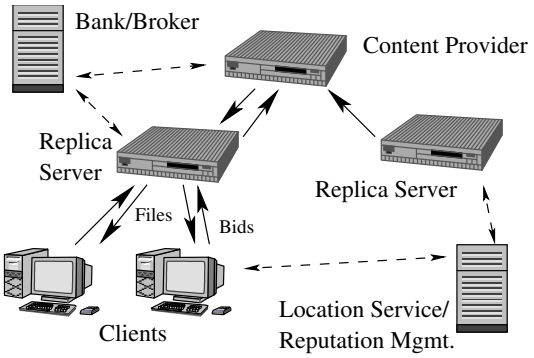

Figure 1. *A simple RME.* Clients bid for access to local replicas. Third party machines provide peripheral services, including currency exchange and reputation management.

In the following section we argue that the RME is a very flexible and robust solution to large and complex replica management problems. We then present a few examples of successful experiments with RMEs. Finally, we discuss current limitations and directions for future work.

[1]This research supported by NSF career award #ANI-9985250 and NFS ITR award #CCR-0085899. Dennis Geels is supported by the Fannie and John Hertz Foundation

2. Why an economic model?

Automatic resource management When two clients compete for access to a server with limited resources, some decision must be made as to which requests are *more important*. An economic approach defines the *importance* of a request as the amount the requester is willing to pay. A client provides useful feedback about its priorities by offering to pay servers more for certain replicas. One could argue that money is really nothing more than society's best attempt at producing a ranking of the relative importance of most everything.

The economic model also helps a replica system cope with fluctuating demand. As hot spots appear, such as when important news breaks or a popular web site links to a normally-low-traffic page, the high demand increases the cost that servers can charge for access to replicas of the hot content. This increase encourages other servers to host a replica, distributing the load and sharing the profit. Similarly, an economy can adapt to the addition or deletion of machines without intervention from human administrators.

Also, an economy provides an easy way to decide *when* to add new servers to a system. System administrators, like capitalist entrepreneurs, can monitor price fluctuations for areas with consistently high prices, which suggests that client demand exceeds replica supply.

Scalability Replica Management Economies also share the scalability benefits of cooperative P2P alternatives. Their use of local, greedy control algorithms avoids the computation and bandwidth bottlenecks that may appear if storage allocation, network monitoring, and failure detection are performed by a central authority.

Guarantees through mechanism design One subfield of Game Theory, called Mechanism Design, studies techniques for setting system rules (algorithms, prices, etc.) in order to induce outcomes with certain desired properties. These properties may include cooperation, a balanced budget, and various definitions of "fairness".

As a simple example, we could define an economy in which clients and servers interact using a Second-Price Auction. Each client submits a bid for replica access; the server then awards access to the highest bidder but charges the amount bid by the runner-up. It can be shown[25] that this method guarantees that "rational" clients will bid honestly. Many generalizations of this simple second-price auction have been proposed which may prove useful in replica management economies.

Benefits in a federated environment A network of machines is said to be *federated* if the machines operate in separate administrative domains. They may cooperate to attain a common good, but each is autonomous and primarily concerned with its own success and profitability.

RMEs fit naturally in this type of environment, which motivates most of Microeconomics and Game Theory. RMEs explicitly deal with real trust and administrative boundaries, as well as real money. They assume that machines may often reject requests, will not always volunteer truthful information, and demand payment proportionate to the work they expend. These concepts usually must be grafted onto other systems before they can be deployed in a federated environment.

Benefits in a trusted infrastructure On the opposite end of the spectrum, one could imagine an environment containing a single administrative domain. All machines cooperate fully, accepting external storage and retrieval requests for the common good.

Despite their apparent differences, both content distribution networks and pure, cooperative P2P systems assume this environment. The former tend to employ a more global allocation algorithm and possibly restrict the set of machines that initiate the storage requests, but both approaches rely on the same inter-machine cooperation.

In contrast, machines in a replica management economy accept external requests only when paid enough to make the action worthwhile. There is no need in this environment for machines to maintain individual profitability; however, this restriction on cooperation can improve system robustness.

Unbounded cooperation, although conceptually simple and morally pleasing, allows a single machine to reduce the availability of many others. Poorly configured or broken machines may accidentally flood the system with unnecessary storage requests. Compromised machines may launch Denial of Service attacks. Or, perhaps more likely, greedy users will consume more resources than they should[14].

In an RME, faulty or malicious machines must pay for service, and their funds are finite. Overloaded machines can raise their prices until demand drops or the failed machines run out of money. Thus, unlike more trusting models, an RME bounds the impact of failure or active attack.

One could impose a similar bound on any replica management system; however, fixed bounds can be overly restrictive. They limit the flexibility of machines that are functioning perfectly yet require a great deal of resources. In an RME, the limit is soft; a machine can always acquire access to a replica if is willing and able to pay enough. In Game Theory, this property is called *consumer sovereignty*.

Benefits in the internet The internet is arguably the most important environment to consider when designing a large-scale replica system. Like many networks, it is neither fully cooperative nor fully federated; it contains many competitive domains, each containing machines that cooperate more or less completely.

One could treat domains as opaque units and only impose a replica management economy among them. This ap-

proach would allow competitors to share resources safely.

One could also expose the machines in each domain and extend the economy to handle intra-domain interactions as well. As shown above, the economic model provides interesting benefits even within trusted domains.

Machines could still be programmed to favor others from their own domain. The RME does not prevent such *coalitional* activity; however, increasing the dependencies between machines decreases the robustness benefits of an RME. As in the real world, tying a greater portion of one's income or output to a favored trading partner or single resource is often risky. The lessons from Economics must be considered when programming members of an RME.

3. Previous results

A small number of previous projects have explored an economic approach to replica management. We summarize their results here, in order to frame the following discussion of future work.

Kurose and Simha [17] used an analytical model to examine the convergence rates of a decentralized allocation algorithm. They assumed a mostly-cooperative environment wherein machines redistributed files to needier machines.

Ferguson ([11] and later in [12]) implemented and measured the performance of a competitive bid-auction mechanism. He found that simple bidding mechanisms produced good allocations for a various access patterns.

Later, the Mariposa project [23] published a design for the most complete replica management economy to date. They implemented an auction mechanism that distributed database tables and queries among autonomous replicas. They found that their economic system balanced query load across replicas better than a static query optimizer.

Clearwater's book [7] and a paper from Tucker and Berman [24] are useful sources for other, less related work. The latter includes a discussion of reasons for which the economic paradigm has not yet been widely adopted.

Recent work applying Game Theory to computer systems seems to focus on networking problems, such as sharing the cost of multicast [10, 15] and handling congestion in the internet [16]. These papers show intriguing adaption of economic theory but are not directly applicable to replica management.

4. Directions for future research

In this section we address the major obstacles to widespread adoption of the RME in popular replica management systems.

Player design Individual machines must be programmed to prioritize their requests and set prices. In general, a machine requires a *utility function*, which rates the relative worth of sets of files. When deciding, for instance, whether to discard a replica to free resources, the machine simply compares the expected *utility* of each alternative.

The utility function could consider the storage and network resources consumed by the content it stores, the money it expects to receive in exchange for those resources, and the amount of money it plans to spend to acquire content in the future.

A simple utility function may assign a fixed worth to each request. We are currently investigating utility functions that favor clusters of files that are expected to be accessed in the near future.

The amount of computation and prediction required for a good utility function is an open problem. Evidence suggests that simple methods will do reasonably well [12].

Performance Greedy, decentralized algorithms rarely achieve the maximum level of performance achievable by centralized, analytical methods; however, one can sometimes bound the difference. The characteristics of economies and players that enable such bounds should be explored more fully.

Also, we should consider the total effect of an economic approach. One might prefer an RME, which requires extra machines and network resources, over an analytical model that requires more human intervention, is less flexible, or imposes heavy control overhead.

Complexity Some may argue that an RME is harder to understand, and hence to control and repair, than a more centralized approach. For large networks, however, the complexity of any system soon exceeds the limits of direct analysis. Instead, we rely on abstractions, summaries, and models of the system's behavior. The field of Economics has developed many useful models that we may apply to the behavior of an RME.

Not group strategyproof The term *strategyproof* from Game Theory refers to games whose rules discourage rational players from lying. A game is *group strategyproof* if a group of players cannot benefit even if the entire group cheats together.

This restriction is often desirable, but very hard to guarantee. Real economies are not group strategyproof; they often develop overseer organizations and anti-trust legislation to prevent certain behavior by coalitions of greedy players. A replica management economy may require similar solutions. Reputation management systems may also help limit the spread of destructive coalitions.

Perhaps future results in Mechanism Design will better characterize rules that induce group strategyproofness.

Starvation Purely economic systems present the danger that poorer clients might be unable to afford reasonable ser-

vice. This problem, like the previous one, may be dealt with either through Mechanism Design or external restrictions.

For example, one could dedicate a small set of servers to serve one's clients for free. The other servers, which operate in the RME, would provide higher QoS guarantees for those able to pay.

Need electronic currency An RME requires a secure, efficient payment mechanism if its digital money is tied to "real" money. A large system must handle millions of transactions per second, and latency and availability requirements rule out centralized systems.

No existing system meets all of our requirements, but several warrant further research. Digital cash systems [5, 13] provide secure, anonymous, offline payments, but require significant computational overhead. Probabilistic methods [18, 22] amortize communication with a central bank across many transactions. Millicent [19] used symmetric-key cryptography to optimize the transaction phase, which required relaxing security goals.

5. Conclusion

We have argued for the Replica Management Economy as a robust, flexible solution for large-scale replica management. RMEs allow machines a level of autonomy that should be expected in a heterogeneous environment like the internet. They rely on an economic model of interaction that allows, yet flexibly bounds, cooperation across domains.

Much work remains in the design of RME protocols and local player algorithms. We are currently building an experimental testbed in which to explore this design space, within the framework of a large-scale storage system. We hope to develop a system that matches current methods in performance and surpasses them in robustness and flexibility.

References

[1] Akamai technologies, inc. http://www.akamai.com/.

[2] Gnutella. http://www.gnutellanews.com/information/.

[3] Mojonation. http://www.mojonation.net/.

[4] S. Acharya and S. B. Zdonik. An efficient scheme for dynamic data replication. Technical Report CS-93-43, Department of Computer Science, Brown University, 1993.

[5] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. In *Communications of the ACM*, 1985.

[6] I. Clark, O. Sandberg, B. Wiley, and T. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Proc. of the Workshop on Design Issues in Anonymity and Unobservability*, pages 311–320, Berkeley, CA, July 2000.

[7] S. H. Clearwater, editor. *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific Press, 1996.

[8] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with CFS. In *Proc. of ACM SOSP*, October 2001.

[9] A. Demers, K. Petersen, M. Spreitzer, D. Terry, M. Theimer, and B. Welch. The Bayou architecture: Support for data sharing among mobile users. In *Proc. of IEEE Workshop on Mobile Computing Systems & Applications*, pages 2–7, 1994.

[10] J. Feigenbaum, C. H. Papadimitriou, and S. Shenker. Sharing the cost of muliticast transmissions. In *Proc. of ACM STOC*, 2000.

[11] D. Ferguson. *The Application of Microeconomics to the Design of Resource Allocation and Control Algorithms in Distributed Systems*. PhD thesis, Columbia University, 1989.

[12] D. Ferguson, C. Nikolaou, J. Sairamesh, and Y. Yemini. Economic models for allocating resources in computer systems. In S. H. Clearwater, editor, *Market-Based Control: A Paradigm for Distributed Resource Allocation*. 1996.

[13] N. Ferguson. Single term off-line coins. In *EUROCRYPT*, 1993.

[14] G. Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[15] K. Jain and V. Vazirani. Group strategyproofness and no subsidy via lp-duality. 1999.

[16] P. Key and D. McAuley. Differential qos and pricing in networks: where flow-control meets game theory. In *IEE Proceedings Software*, 1999.

[17] J. F. Kurose and R. Simha. A mircoeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on Computers*, 8(5):705–717, May 1989.

[18] R. Lipton and R. Ostrovsky. Micro-payments via efficient coin-flipping. In *Financial Cryptography Conference*, 1998.

[19] M. Manasse. The millicent protocols for electronic commerce. In *USENIX Workshop of Electronic Commerce*, 1995.

[20] M. Rabinovich and A. Aggarwal. Radar: A scalable architecture for a global web hosting service. In *The 8th Int. World Wide Web Conf*, May 1999.

[21] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz. Maintenance free global storage in oceanstore. In *Proc. of IEEE Internet Computing*. IEEE, Sept. 2001.

[22] R. Rivest and A. Shamir. Payword and micromint: Two simple micropayment schemes. In *Security Protocols Workshop*, 1996.

[23] J. Sidell, P. Aoki, S. Barr, A. Sah, C. Staelin, M. Stonebraker, and A. Yu. Data replication in Mariposa. In *Proc. of IEEE ICDE*, pages 485–495, Feb. 1996.

[24] P. Tucker and F. Berman. On market mechanisms as a software technique. Technical Report CMU-CS-87-143, U. C. San Diego, Dec. 1996.

[25] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Finance*, 16:8–37, 1961.