Self-Recharging Virtual Currency*

David Irwin
Duke University
irwin@cs.duke.edu

Jeff Chase
Duke University
chase@cs.duke.edu

Laura Grit†
Duke University
grit@cs.duke.edu

Aydan Yumerefendi
Duke University
aydan@cs.duke.edu

ABSTRACT

Market-based control is attractive for networked computing utilities in which consumers compete for shared resources (computers, storage, network bandwidth). This paper proposes a new *self-recharging* virtual currency model as a common medium of exchange in a computational market. The key idea is to recycle currency through the economy automatically while bounding the *rate* of spending by consumers. Currency budgets may be distributed among consumers according to any global policy; consumers spend their budgets to schedule their resource usage through time, but cannot hoard their currency or starve.

We outline the design and rationale for self-recharging currency in Cereus, a system for market-based *community* resource sharing, in which participants are authenticated and sanctions are sufficient to discourage fraudulent behavior. Currency transactions in Cereus are *accountable*: offline third-party audits can detect and prove cheating, so participants may transfer and recharge currency autonomously without involvement of the trusted banking service.

Categories and Subject Descriptors

C.2.4 [**Computer-Communications Networks**]: Distributed Systems; D.4.6 [**Operating Systems**]: Security and Protection; H.4.3 [**Information Systems Applications**]: Communications Applications

General Terms

Economics, Management, Performance, Security

Keywords

Virtual Currency, Market

*This research is supported by IBM as an extension to the Cluster-on-Demand project (NSF ANI-0330658).

†Laura Grit is a National Physical Science Consortium Fellow.

SIGCOMM'05 Workshops, August 22–26, 2005, Philadelphia, PA, USA.

1. INTRODUCTION

Technologies for deployable networked computing utilities have matured. Examples of shared "cyberinfrastructure" include content services networks, computational grids, application hosting services, and network testbeds such as PlanetLab. The growing reach and scale of these systems exposes the need for more advanced solutions to manage shared resources. For example, the tragedy of the PlanetLab commons is apparent to any PlanetLab guest or observer, and there is now a broad consensus that networked utilities need resource control that is strong, flexible, and fair.

Market-based resource control seems a logical next step. The Internet has already grown convincing early examples of real, practical networked systems structured as federations of self-interested actors who respond to incentives engineered to induce a desired emergent global behavior (most recently BitTorrent [8]). At the same time, grid deployments are approaching the level of scale and participation at which some form of market-based control is useful or even essential, both to regulate resource allocation and to generate incentives to contribute resources to make them self-sustaining.

This paper proposes a new virtual currency model and outlines its use in Cereus[1], a system for service-oriented utility computing. The Cereus system is based on SHARP [10], in which resource provider sites lease shared hardware resources to a community of consumers (hosted services), with resource exchanges brokered by third parties. Previous SHARP-based systems [6, 10] have been limited to barter exchanges, which rely on consumers to identify mutual coincidences of needs or enter into transitive barter arrangements. Cereus introduces a common currency as a step toward an efficient market economy in which resource control is open, flexible, robust, and decentralized. We are implementing a prototype in which hosted services bid currency to lease resources and bind them to dynamic clusters of Xen [3] virtual machines.

Cereus uses a virtual currency—called *credits*—rather than a fungible cash currency (e.g., dollars). Our premise is that external control over resource allocation policy is crucial in utilities that serve the needs of a research community, such as PlanetLab. In Cereus, consumers are funded with credit budgets according to a pluggable policy; as with other virtual currencies, credits may be distributed and del-

[1]Cereus is a desert cactus that blooms once a year very late at night. The name might also mean "cyberinfrastructure exchange and resource economy for utility services". At least two related projects exist with similar pronunciation but different spelling.

egated in a decentralized or hierarchical fashion to meet policy goals [20, 21]. In contrast, cash markets allow real-world wealth to dictate resource access policy, and currency allotted by a global policy can be diverted to other purposes. We envision that Cereus credits may be purchased or redeemed for cash (or other goods) as a basis for a cash utility market or to reward contributors.

Cereus currency is *self-recharging*: the purchasing power of spent credits is restored to the consumer's budget after some delay; in effect, the credits *recharge* automatically, ensuring a stable budget but bounding the opportunity for hoarding. The purpose of self-recharging currency is to eliminate reliance on fragile mechanisms to recycle currency through the economy. In particular, it is not necessary for each consumer to contribute in proportion to its usage, although a global policy could punish free riders by draining their credit budgets if desired. Self-recharging currency is not a new idea: Lottery Scheduling [21] is one simple and popular example of a system with self-recharging currency, and the Cereus credit currency model was inspired by Sutherland's 1968 proposal for a futures market in computer time [17]. One goal of Cereus is to generalize self-recharging currency to continuous, rolling, brokered, multi-unit futures auctions.

The Cereus design uses the SHARP abstraction of *accountable claims* as the basis for the currency system itself. SHARP claims are digitally signed, timestamped assertions of resource ownership and control. SHARP supports secure delegation of claims, and enables untrusted auditors to detect and prove responsibility for misbehavior by inspecting the sets of claim records issued by any actor over a period of time. While a trusted banking service authenticates participants and issues currency budgets, holders may recharge their currency locally, and there is no need for the bank to arbitrate currency transfers. The bank or its delegates may audit claim trails to detect and punish cheating, selecting the audit degree to balance audit overhead and the probability of detection. An audit-based approach is suitable for a community environment in which actors are not anonymous, faithfulness dominates privacy, and resource providers and brokers are selfish but not malicious.

2. OVERVIEW AND RATIONALE

Cereus is an instance of the large class of market-based resource control systems in which self-interested actors use their currency to obtain resources at market prices, subject to their budget constraints (e.g., [2, 4, 12, 14, 17, 20, 21]). *Currency* is denominated in some common unit that constitutes a standard measure of worth ascribed to goods. Each unit of currency is possessed exclusively by some actor, who may transfer it to another actor in exchange for goods.

Money is any currency that functions as a store of value. The holder of money has full control over when and how it is spent. Money economies depend on a balance of transactions to recycle currency through the economy. Once money is spent, the spender permanently relinquishes ownership of it. Thus actors may run out of money, leaving resources idle even while demand for them exists (*starvation*). As they spend their money, they must obtain more, either by offering something else of value for sale, or by means of an income stream from an external source (e.g., the state). Actors may save or *hoard* income over time to increase their spending power in the future, and may use that accumulation to corner the market, manipulate the price of resources in the system, or starve other users. Some systems (e.g., [7, 14]) limit hoarding by bounding savings or discourage it by imposing a demurrage (a tax on savings). Money economies are prone to cycles of inflation/deflation caused by fluctuations in the circulating money supply.

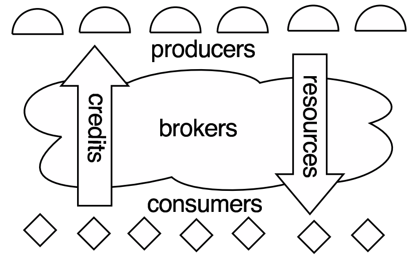

Figure 1: Flow of credits and resources in a Cereus community. Leasing rights for resources flow from resource provider sites down through a broker network to consuming service managers. Brokers sell resource rights using an auction protocol: payments in credits flow up through the broker network.

In contrast, self-recharging currency refreshes each consumer's currency budget automatically as it is spent. In Cereus, credits recharge after a fixed *recharge time* from the time they are spent. The automatic recycling of credits avoids the "feast and famine" effects of money economies (hoarding and starvation). A consumer with a budget of c credits can spend them according to its preferences, substituting resources or scheduling its usage through time according to market conditions, but it can never hold contracts or pending bids whose aggregate face value exceeds c. A consumer can never accumulate more than c credits, and it can spend exactly c credits in any interval equal to the recharge time.

Lottery Scheduling [16, 21] uses a form of self-recharging currency that recharges on every time quantum. A lottery scheduler fills demand in proportion to instantaneous currency holdings; its single-resource single-unit auction protocol is a simple random lottery, making it efficient enough for fine-grained scheduling (e.g., at CPU quanta). However, Lottery Scheduling is not a market: consumers receive no price signals and cannot defer their purchasing power into the future, e.g., they have no incentive to defer a resource request in response to spot shortages. In contrast, credits in Cereus are redeemable for any resource available for sale according to prevailing prices within some time window. With a short recharge time the system is similar to Lottery Scheduling in that it approximates proportional share scheduling and provides no incentive for saving; longer recharge times provide more incentive to schedule resource usage in time, but begin to resemble a money economy with increasing risk of hoarding and starvation.

2.1 The PDP-1 Market

Self-recharging currency was first introduced by Sutherland in 1968 [17] under the name "yen" to implement a self-policing futures market for a PDP-1 computer. Although the paper is widely cited, we are not aware of any subsequent published work that generalizes this form of virtual currency. We now describe credits in the PDP-1 market, then outline how we apply the concept to a continuous market for multiple resources.

Consumers in the PDP-1 market bid yen for time slots on the computer during the next day. In essence, the system runs an open ascending English auction for each time slot.

- Bidders commit their currency at the time it is placed for bid; the currency is held in escrow and is not available for concurrent bids.

- If a bid is preempted or canceled, the bid currency is immediately available to the bidder. A high bid preempts a low one since only a single unit of each good is for sale; preemption is immediately apparent to the loser, which can reclaim its yen.

- Each bidding period determines the allocation for the next day. The bidding period ends before the beginning of the day; once the bidding period ends no user is allowed to change bids.

- The market *recharges* the winning bid's currency after the resource is consumed, i.e., after the purchased slot expires. The yen then become available for auctions for the following day.

2.2 Generalizing the PDP-1 Market

Cereus extends self-recharging currency to a networked market with multiple rolling auctions. As in the PDP-1 market, the bidder commits credits to each bid, and these credits are transferred at the time of the bid.

The crux of the problem is: *when to recharge credits spent for winning bids?* The PDP-1 market recharges credits when the purchased contract expires, which occurs 24-48 hours after the auction, depending on the time of day for the purchased slot. It is easy to see that the PDP-1 credit recharge policy is insufficient for a market with continuous rolling auctions. If credits spent for a contract recharge as soon as the contract expires, then a dominant strategy is to bid for instant gratification so that spent credits recharge sooner. The value of scheduling usage through time would be diminished, and the market would revert to a proportional-share economy such as Spawn [20] or Lottery Scheduling [21]. In general, a variable recharge time disadvantages buyers of resource contracts that expire further in the future. The effect is negated in the PDP-1 market because the buyer always receives the recharged credits in time to bid for the next auction.

To maintain a consistent recharge time across multiple concurrent auctions, Cereus currently enforces the following *credit recharge rule*: spent credits recharge back to the buyer after a fixed interval—the *recharge time*—from the point at which the buyer commits the credits to a bid, as depicted in Figure 2. The recharge time is a global property of the currency system. The credit recharge rule has three desirable properties:

- It encourages early bidding, yielding more accurate price feedback to other bidders, depending on the auction protocol.

- It discourages canceled bids, since shifting credits to another bid delays recharge of the credits.

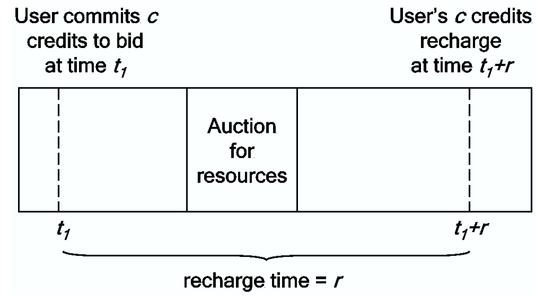

Figure 2: Cereus currently enforces the *credit recharge rule*: currency returns to the buyer after a configurable recharge time r from when the currency is committed to a bid in an auction.

- It encourages early bidders to bid higher, to avoid incurring the opportunity cost on any credits returned by the auction protocol for losing bids.

These properties address known limitations of many common auction protocols. For example, open ascending auctions with fixed call times encourage predatory late bidding just before the auction closes, a property that the PDP-1 market has in common with Ebay. Also, open ascending auctions tend to minimize profit to the seller unless potential buyers are motivated by uncertainty to expose their willingness to bid high: this problem is a key motivation for sealed-bid auctions.

One implication of the credit recharge rule is that it is possible for a buyer to receive credits back before a contract expires, and thus it is possible for a buyer to hold pending bids and contracts whose face value exceeds its credit budget c. This occurs rarely if the recharge time is long relative to the lease terms, as in the PDP-1 market. Whether or not it occurs, it is easy to see that the key property of the PDP-1 market continue to hold: a consumer can commit exactly its budget of c credits in any interval whose duration is the recharge time. A decision to spend or save credits now does not affect the consumer's purchasing power in future intervals.

The credit recharge rule has a second implication for brokered auctions: credits passed to a broker expire to preserve the currency balance when they revert to the consumer's budget after the recharge time. As a result, brokers have an incentive to spend credits quickly. To simplify management of the currency, our prototype imposes a *binding bids* rule for credits: *credits committed to a bid become unavailable until they recharge.* Bidders cannot cancel or reduce their bids once they are committed. In this way, it is never necessary for a broker to return escrowed credits (e.g., to a losing or canceled bid) after the broker has spent them.

The Cereus currency model is well-matched to *price-anticipating* auction protocols as in Tycoon [9, 12], in which brokers accept all bids but the service rendered varies according to contention. Our prototype uses a hybrid auction in which the broker posts a *call price* based on recent history, and returns resources in proportion to the bid during periods of unexpectedly high demand. The protocol is simple for bidders and incentive-compatible if the call price is accurate, but it requires policing of brokers to hold them accountable for their conduct of auctions.

2.3 Self-Recharging Credits vs. Money

Self-recharging credits are analogous to a money economy in which the state provides each actor with a steady flow of income over time, while imposing a 100% income tax to prevent hoarding. Actors may purchase goods by contracting to divert some portion of their income for the duration of the recharge time in exchange for the good: in effect, the state gives a tax credit for mortgaged income. The incentive to conserve income by delaying purchases is adjustable based on the recharge time. With a recharge time of zero there is no incentive to conserve, and the system reverts to proportional share allocation (as in Lottery Scheduling). As the recharge time grows, additional mechanisms must be introduced to recycle funds, as in a money economy.

In a perfect market, each consumer is assured access to a share of resource value proportional to its share of the wealth. With self-recharging currency, this assurance applies to any interval whose duration is the recharge time. However, the purchasing power within any interval depends on the movement of prices through time. Cereus emphasizes the importance of adjustable incentives to schedule resource usage through time, in contrast to many previous systems (including Lottery Scheduling) which balance only instantaneous supply and demand.

3. CEREUS CURRENCY DESIGN

Self-interested actors may lie, cheat, or steal to maximize their utility, which creates challenges for dependable currency management. For example, actors may attempt to spend currency they do not possess, or spend credits before they recharge. One solution is to coordinate all currency transactions through a trusted banking service. Credits offer a simple and enforceable decentralized alternative to cash currency, assuming reasonably synchronized clocks, which are prerequisite to any lease scheduling scheme. Credits have an inherent verifiable time element: recharges must be sufficiently delayed, and transfers have an expiration time. Our approach leverages this advantage and simplifies currency management by enabling local currency actions without synchronous policing. Instead, currency actions are accountable and are subject to certified audit after the fact.

To this end we extend the accountability properties of SHARP and make credit transfers accountable: if an actor misrepresents its credits holdings and exceeds its budget, an auditor may detect any misbehavior and construct undeniable cryptographic proof that is verifiable by any third party. This approach requires that participants are securely bound to sanctionable identities, and authenticate to a banking service before receiving currency. Our premise is that accountability is a sufficient disincentive for abuse by any participant in a community. Actions of an accountable actor are provable and non-repudiable, and may be legally binding. A third party such as a legal authority or reputation service [1, 13] may adjudicate and hold a misbehaving actor accountable for misbehavior by imposing some sanction.

The credit recharge rule and the binding bids rule make it possible to represent credits as chains of SHARP claims, similarly to resource lease contracts or resource "tickets", in a simple and elegant way. Accountable credit management provides a strong foundation on which we can layer more complex policies and protocols to define the rules in a market system and provide the needed incentives to induce certain desired behavior.

The SHARP claim construct allows actors to formulate and exchange unforgeable assertions about control over resources. A claim record is an XML object asserting that a specified principal (the *holder*) controls some resources (a *resource set*) over some time interval (its *term*). To protect the integrity of claims, each claim record is signed with the private key of the principal conferring ownership of the resource (the *issuer*). The issuer names the holder by a public key; by signing the transfer, the issuer certifies that it authorizes any entity possessing the corresponding private key to control the resources named in the claim.

A claim holder may *delegate* a portion of its resources to another principal by generating a new claim record, signing it with its private key, appending it to the parent claim, and issuing it to the receiver. A sequence of such delegated claims constitutes cryptographically self-certifying proof that certain validity properties form a chain of transfers.

Credit currency may be represented as a claim for c units of resources of a special type: credits. Such a claim—together with its supporting sequence of parent claims rooted in a claim issued by the community bank—is called a *credit note*. (A credit note is a special case of a SHARP *ticket*.) Trusted banks issue credit notes to consumers for some term, according to a global policy. The term should be much longer than the recharge time. The consumer bids with its credits by delegating a new credit note for a subset of its credits, setting the term to the recharge time, and transferring the new credit note to a broker. The start time of the delegated credit note is a timestamp for the bid. The broker may inspect the credit note to determine that it is valid. When the consumer's credit note expires, it can no longer generate valid bids until the bank issues it a new credit note, possibly adjusting the consumer's credit allotment c according to some global policy. The term of notes issued by the bank is chosen to balance stability and overhead with agility of the global budget policy.

When a buyer bids with a valid credit note, it delegates control of the bid credits to the broker. The broker may spend the credits at any time by delegating a new credit note to another supplier. The credit note's term represents the expiration time for the note. As described in Section 2.2, credit expiration prevents brokers from accumulating credits, and preserves the balance of credits in the system without allowing brokers to go into deficit or requiring them to hold working capital (after initial bootstrapping of a broker network).

3.1 Auditing

Following the scheme we described, bid credits pass up a chain of intermediary brokers and accumulate at the site providers, where the purchased resources reside and where the leased resource claims originate. Similarly, the resource tickets pass down the chain of brokers to arrive at service managers, who redeem them back to the sites for leases on the purchased resources as described in [10]. Thus all tickets for resources at a given site, and all notes for credits spent to obtain resources, ultimately accumulate at the sites (Figure 3).

It can be seen that the credit notes include all information needed to audit all credits transfers and to detect and prove misbehavior. For example, if any end consumer overspends its credit holdings, or attempts to recharge its credits before

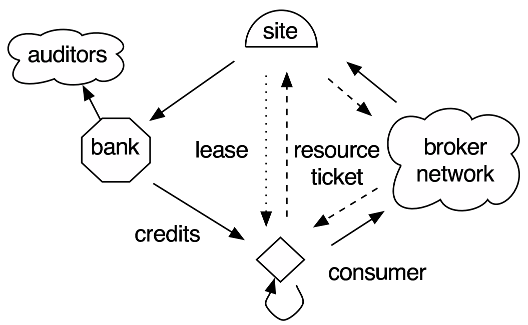

Figure 3: Auditing cycle. A trusted bank service issues initial credit budgets to consumers. Consumers self-recharge their credits and use them to obtain resource tickets through a network of brokers; credit transfers do not involve the bank. Spent credit notes eventually propagate to the originating resource sites, who redeem them to the bank as proof of value offered to the community. The bank or its delegates audit the credit notes to hold participants accountable for their transfers.

their time, this misbehavior is evident in the set of credit notes that it issues, using the standard SHARP algorithm to validate trees of delegated claims. This mechanism is described in detail in [10]: an auditor can build the tree of delegations rooted in any claim record and validate that the aggregate number of units in the children of each claim that overlap at any time do not overcommit the resource units in the parent claim. In addition, it is necessary to check that all credit notes issued by an end consumer have term durations that are equal to the recharge time. SHARP assumes that clocks are synchronized within some tolerance that is small relative to the granularity of lease terms and the recharge time: participants have an incentive to reject credits from peers whose clocks are fast, since they expire sooner.

Note that auditors need not be trusted, assuming precautions are taken to protect against collusion involving auditors (e.g., random selection). Proofs of misbehavior are undeniable and are verifiable by a third party: and cannot be fabricated. To improve performance, the audits may involve probabilistic sampling and they may run in parallel on multiple auditors. Of course, the audits are not privacy-preserving: all transfers must be exposed to the auditors, although it is possible in principle to encrypt predecessor notes with the banker's public key in order to conceal sensitive pricing information from other participants along the supply chain.

It remains only to consider the incentives for provider sites to expose credit notes to an auditor. The expired credit notes held by suppliers are proof of the value they have provided to the network. We propose that they may be returned voluntarily to the bank for a reward such as cash or a fresh supply of credits for consumers associated with the site.

4. RELATED WORK

The systems community has flirted with market systems and currency management regularly over the years [5, 14, 15, 20, 23]. Space precludes a comprehensive survey of the literature.

Proposals such as Karma [19], SWIFT [18], and CompuP2P [11] use currency to address the free-rider problem in a P2P setting where anonymous users enter and leave the system at a high rate. The design of currency management in P2P systems reflects the model that trust is only by reputation. In contrast, participants in a Cereus community are securely bound to identities that can be sanctioned if misbehavior is detected.

Industry initiatives in utility computing (e.g., [22]) are often based on cash and rely on recycling of currency within the larger common market. However, problems of secure bidding, payment, escrow, and contract accountability remain. Virtual currency enables reliance on asynchronous accountability mechanisms in a way that is not possible with cash, which is fungible and may be diverted if fraud occurs. Offline audits can identify theft but the loss is real and potentially permanent.

There have also been recent proposals for community virtual currency in systems similar to Cereus. Management of community virtual currency often assumes a single trust domain that alleviates the need to verify transactions [14]. Networked communities cannot make this assumption. Cereus is similar to other network resource exchanges in its use of secure exchange protocols with a trusted bank that validates currency transactions [7, 12].

Cereus is perhaps most closely related to Tycoon [9, 12], a recent market-based system for networked clusters. The Cereus self-recharging currency model is complementary to Tycoon, and Cereus currently uses a variant of a price-anticipating auction protocol proposed in [9], in which all bidders for a resource receive a share that varies with contention. It is shown in [9] that the protocol always yields a Nash equilibrium when auctions are strongly competitive, competition is stable, and bidders adapt to recent history according to stable utility functions. Like Tycoon, bidding in Cereus can be autonomous, asynchronous, and non-interactive when the bidding agent knows the buyer's utility functions. One difference from Tycoon is that Cereus is based on a leasing abstraction, in which the result of each auction assures the buyer a predictable resource share for the duration of the term. In contrast, resource assignments in Tycoon vary continuously with contention, emphasizing agility over stability and assurance. The Cereus/SHARP leasing abstraction extends easily to a wide range of resource types and offers a configurable balance of agility and stability negotiated between the buyer and broker.

5. CONCLUSION

This paper proposes a *self-recharging* virtual currency model as a common medium of exchange in Cereus, a computational market for *community* resource sharing. A community admits members based on their identities; in return for the benefits of membership, members submit to accepted standards of behavior within the community. Community resource sharing in Cereus is *accountable*: actions taken with a member's identity are non-repudiable, and members are subject to auditing and sanctions for misbehavior, including forfeiture of currency or ejection from the community (discommendation).

Cereus defines a virtual currency for two reasons. First, virtual currency is administered by the community for the benefit of the community: currency may not be drained or diverted for other purposes, and currency obtained from external sources has no power to subvert policy choices within the community. Second, virtual currencies allow experimentation with the nature of currency and its implications for market structure and dynamics. Our premise is that alternative currency models can help to harness market principles as a *technology* for flexible, robust distributed resource allocation, enabling market-inspired systems that are simpler, more stable, more predictable, and/or more controllable than real-world economies.

The self-recharging currency model (credits) used in Cereus defines a recharge time that offers a configurable tradeoff between proportional-share allocation and a money economy. The recharge time limits the rate of consumer spending and bounds hoarding and starvation. Representing credits as SHARP tickets provides a secure and accountable method of currency transfer. Asynchronous, off-line third-party audits can detect misbehavior, allowing currency actions to proceed autonomously without synchronous interaction from a trusted banking service.

6. REFERENCES

[1] K. Aberer and Z. Despotovic. Managing Trust in a Peer-2-Peer Information System. In *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM-01)*, pages 310–317, 2001.

[2] M. Balazinska, H. Balakrishnan, and M. Stonebraker. Contract-based load management in federated distributed systems. In *First Symposium on Networked Systems Design and Implementation (NSDI)*, March 2004.

[3] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *Proceedings of 19th ACM Symposium on Operating Systems Principles (SOSP)*, October 2003.

[4] R. Buyya, D. Abramson, and J. Giddy. An economy driven resource management architecture for global computational power grids. In *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, June 2000.

[5] R. Buyya, D. Abramson, J. Giddy, and K. Stockinger. Economic models for resource management and scheduling in grid computing. *Special Issue on Grid Computing Environments, The Journal of Concurrency and Computation: Practice and Experience (CCPE)*, 14(13-15), Nov.-Dec. 2002.

[6] B. Chun, Y. Fu, and A. Vahdat. Bootstrapping a distributed computational economy with peer-to-peer bartering. In *Proceedings of 1st Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[7] B. N. Chun, P. Buonadonna, A. AuYoung, C. Ng, D. C. Parkes, J. Shneidman, A. C. Snoeren, and A. Vahdat. Mirage: A microeconomic resource allocation system for sensornet testbeds. In *Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors*, May 2005.

[8] B. Cohen. Incentives build robustness in BitTorrent. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[9] M. Feldman, K. Lai, and L. Zhang. A Price-Anticipating Resource Allocation Mechanism for Distributed Shared Clusters. In *Proceedings of the ACM Conference on Electronic Commerce*, June 2005.

[10] Y. Fu, J. Chase, B. Chun, S. Schwab, and A. Vahdat. SHARP: An architecture for secure resource peering. In *Proceedings of 19th ACM Symposium on Operating System Principles (SOSP)*, October 2003.

[11] R. Gupta and A. Somani. CompuP2P: An architecture for sharing of compute power in peer-to-peer networks with selfish nodes. In *Proceedings of Second Workshop on Economics of Peer-to-Peer Systems*, June 2005.

[12] K. Lai, L. Rasmusson, E. Adar, S. Sorkin, L. Zhang, and B. A. Huberman. Tycoon: an implemention of a distributed market-based resource allocation system. Technical Report DC/0412038, HP Laboratories, December 2004.

[13] S. Lee, R. Sherwood, and S. Bhattacharjee. Cooperative Peer Groups in NICE. In *Proceedings of INFOCOM*, 2003.

[14] I. Stoica, H. M. Abdel-Wahab, and A. Pothen. A microeconomic scheduler for parallel computers. In *IPPS '95: Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing*, April 1995.

[15] M. Stonebraker, R. Devine, M. Kornacker, W. Litwin, A. Pfeffer, A. Sah, and C. Staelin. An economic paradigm for query processing and data migration in Mariposa. In *3rd International Conference on Parallel and Distributed Information Systems*, September 1994.

[16] D. G. Sullivan and M. I. Seltzer. Isolation with flexibility: A resource management framework for central servers. In *2000 USENIX Technical Conference*, June 2000.

[17] I. E. Sutherland. A futures market in computer time. *Communications of the ACM*, 11(6):449–451, June 1968.

[18] K. Tamilmani, V. Pai, and A. Mohr. SWIFT: A system with incentives for trading. In *Proceedings of Second Workshop on Economics of Peer-to-Peer Systems*, June 2005.

[19] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer. Karma: A secure economic framework for peer-to-peer resource sharing. In *Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[20] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and W. S. Stornetta. Spawn: A distributed computational economy. *Software Engineering*, 18(2):103–117, February 1992.

[21] C. A. Waldspurger and W. E. Weihl. Lottery scheduling: Flexible proportional-share resource management. In *Operating Systems Design and Implementation*, November 1994.

[22] J. Wilkes, P. Goldsack, G. Janakiraman, L. Russell, S. Singhal, and A. Thomas. *e*os - the dawn of the resource economy. In *Proceedings of Eighth Workshop in Hot Topics in Operating Systems*, May 2001.

[23] R. Wolski, J. Brevik, J. S. Plank, and T. Bryan. Grid resource allocation and control using computational economies. In F. Berman, G. Fox, and A. Hey, editors, *Grid Computing: Making The Global Infrastructure a Reality*. John Wiley & Sons, 2003.