Grid Resource Allocation and Control Using Computational Economies[*]

May 4, 2004

Rich Wolski[˜] John Brevik[‡] James S. Plank[†] Todd Bryan[˜]

[˜] Department of Computer Science
University of California, Santa Barbara

[‡] Mathematics and Computer Science Department
Wheaton College

[†] Department of Computer Science
University of Tennessee

Abstract

In this chapter, we describe the use of economic principles as the basis for Grid resource allocation policies and mechanisms. A computational economy in which users "buy" resources from their owners is an attractive method of controlling Grid resource allocation for several reasons. Economies are intuitively easy to understand, they fit the model of flexible resource usage under local control (which is fundamental to Grid computing), and they can be analyzed through a considerable body of extant theory. We discuss many of the fundamental characteristics of computational economies, particularly as they pertain to Grid computing. We also present G-commerce — a framework that we have used to investigate Grid resource economies — as an example of the type of results that are possible. Finally, we discuss several of the issues associated with empirical investigation of Grid economies as a motivation for future work.

1 Introduction

Most, if not all, Computational Grid resource allocation and scheduling research espouses one of two paradigms: centralized omnipotent resource control [16, 20, 7, 8] or localized application control [11, 4, 2, 18]. The first is not a

[*] This work was supported in part by NSF grants EIA-9975020, EIA-9975015, and ACI-9876895.

scalable solution either in terms of execution efficiency (the resource broker or scheduler becomes a bottleneck) or fault resilience (the allocation mechanism is a single point of failure). On the other hand, the second approach can lead to unstable resource assignments as "Grid-aware" applications adapt to compete for resources. The complexity of the allocation problem and the dynamically changing performance characteristics of Grid resources (due to contention and resource failure) are such that automatic scheduling programs are needed to allocate and re-allocate resources. With resource allocation decisions under local control, the potential for instability exists as competing application schedulers constantly adjust to load fluctuations they, themselves, induce.

As Grid systems are deployed, a variety of engineering approaches to Grid resource allocation have been and will continue to be used [30, 19, 10, 16, 24]. However, almost all of them either rely on a centralized information base, or offer little assurance of allocation stability. There are two formal approaches to the Grid resource allocation problem, however, that specifically address questions of efficiency, stability, and scalability. They are *control theory* and *economics.* Control theory is a viable choice since it includes the notion of feedback explicitly, and system response can be represented by stochastic variables. Good stochastic models of Grid performance response, however, remain elusive, making it difficult to define formally tractable control theoretic mechanisms.

Economic systems, on the other hand, are attractive for several reasons. First, there is a considerable body of theory that attempts to explain the "emergent" behavior of the overall system based on the presumed behavior of the constituents. Secondly, the concept of "efficiency" is well-defined, although it is different from the notion of efficiency typically understood in a computer performance evaluation setting. Finally, economic systems and the assumptions upon which they are based seem familiar, making common intuition a more valuable research asset. While it may be difficult to visualize the effect of a change in the covariance matrix (needed by many control theoretic systems), most people understand fairly quickly that an increase in price for a resource will squelch demand for it.

For these reasons, economic systems have garnered quite a bit of attention as an approach to Grid resource allocation under the heading *computational economy* [12, 47, 9, 7]. In this chapter, we will focus on computational economies for the Grid. It is important to note, however, that the term "computational economy" is also used in e-commerce settings to refer to the use of computers to solve difficult financial problems. For example, researchers are studying ways to resolve combinatorial auctions [33, 34, 28, 17] optimally so that they may be used to distribute "real-world" goods in different economic settings. The distinction largely hinges on whether the economy in question is the "real" economy or an artificial one that has been set up to conform to a certain set of constraints. The approaches we will discuss are based on the latter model. That is, we will assume that the Grid system in question can be made to obey a certain set of economic principles *before* execution begins. Moreover, Grid computational economies can be restarted, whereas the "real" economy cannot. This distinction turns out to be an important one (as we discuss in Section 5), but it necessarily implies that the economy in question is artificial.

Computational economies, defined in this way, can be broadly categorized into two types. *Commodities markets* treat equivalent resources as interchangeable. A purchaser of a resource (such as a CPU of a particular type) buys one of those that are available from a pool of equivalent choices without the ability to specify which resource exactly will be purchased. Alternatively, in *auction markets* purchasers (consumers) bid on and ultimately purchase specific resources provided by *producers.* Each formulation is characterized by theoretical as well as practical advantages and disadvantages. While it is an open question as to which is most suitable for future Grid settings, in this chapter we describe early work (presented more completely in [46]) that addresses these characteristic differences.

The process of conducting computational economic research is, itself, an interesting research topic. With a few exceptions [32, 44], most investigations of various market formulations and mechanisms are simulated. Simulated consumer agents buy resources from simulated producers under various conditions so that the overall economy can be observed. The benefits of this approach are obvious (repeatability, controlled experimentation, etc.) but the model for the interaction of the agents with the system they are using is rarely included. That is, agents interact "out-of-band" when they negotiate resource usage in most simulations. In Grid settings, the load introduced by agents in the market will itself affect resource availability. The extent of this effect is hard to model for the same reasons that control theory is difficult to apply in Grid settings: few stochastic models that match observed performance response are available.

Empirical studies offer an alternative to simulation, although they are labor intensive to conduct. More importantly, they are difficult to design. In particular, Grid resources and testbeds are expensive to maintain, often supporting active "production" users concurrently with Grid research. Additionally, a valid study must motivate users to attach real value to resources and applications. In most experimental environments [39] the subjects realize that they are under study and that they are operating in an artificial economy. While it possible to motivate them to compete economically, this motivation is often unconvincing. In this chapter, we also discuss our experiences pursuing empirical economic results in Grid settings. We have discovered several key issues that must be addressed to make such results convincing.

The remainder of this chapter is organized as follows. In the next section, we present a basic set of assumptions that must be made in order to support a Grid resource economy, and discuss the advantages and disadvantages of different market formulations. Section 4 covers the simulation of Grid economies and presents a methodology we have used to generate early results as a case study. In Section 5 we outline some of the issues associated with effective empirical studies, using our own work-in-progress as an example. Finally, in Section 6 we conclude and summarize as a way of pointing to future research.

2 Computational Economies and the Grid

While a variety of schemes have been proposed as economic approaches to Grid resource allocation, there are a fundamental set of assumptions upon which any "true" economic system must rest. It is often difficult to understand whether a particular resource allocation system

- can be predicted to behave as an economic system, and
- can be analyzed as an economic system.

The former characteristic is beneficial because it permits reasoning about the overall "state" of Grid allocations and the likely changes to that state that may occur. The latter characteristic is desirable because rigorous analysis often leads to explanatory insight.

For a resource allocation mechanism to be economic in nature, it fundamentally must rest on the following assumptions. If it does not, or it is not clear whether these assumptions hold true, the system is not, at its core, a computational economy.

The most fundamental assumption concerns the relationship between supply, demand, and value.

> **Assumption #1**: The relative worth of a resource must be determined by its supply and the demand for it.

In any economic system, the relationship of supply to demand determines value. It is interesting to note that we will not insist that these relationships be *monotonic* in the sense that raising the price will always cause supply to increase and demand to decrease, although this characteristic seems to be an intuitive and natural restriction. Philosophically, we do not include monotonicity as a condition because it is not necessary for the economic results to which we will appeal. The only condition of this type that is mathematically necessary is that if the price of a commodity is allowed to increase without bound, the supply will eventually overtake the demand. That is, from a theoretical point of view, supply must exceed demand for a sufficiently large price. More pragmatically, there are instances in which consumers may increase their demand in the face of rising prices (for example, to purchase on the principle that "you get what you pay for"). That must be modeled correctly.

A second, related assumption concerns the relationship of currency, price, and value.

> **Assumption #2**: The price of a given resource is its worth relative to the value of a unit resource called "currency."

There is nothing inherently special about money in an economic system with the possible exception of its portability. In fact, currency may be "tied" to a specific commodity; for our purposes, however, we will assume that currency is a separate commodity which has utility value for all agents. If currency cannot be used to purchase resources universally (e.g. there is a different currency for each resource type and no way to translate between them) the system (however effective) is not an economic one. Taken in conjunction with assumption *#1*, the result is that price is a function of the relationship between supply and demand, and its units are the units of currency.

Notice that systems which use terms from economics to describe the way allocation decisions are made are not necessarily economic systems. For example, consider a system based on lottery scheduling [42]. Lottery scheduling is a methodology in which resource providers hand out some number of "tickets", each representing the right to use a fixed part of the resource for a given time duration. For each time period (e.g. time slice), a ticket is drawn at random, and its holder is permitted to use the resource. The proportion of time a particular consumer will be allocated is, therefore, proportional to the number of tickets that consumer possesses relative to other consumers.

It is tempting to think of tickets in a lottery-scheduling scheme as currency, and the proportion of tickets each consumer holds as a measure of relative wealth. Unless the proportions are determined through some aggregate function of supply and demand, this simple system is not an economic one under our stricter set of assumptions. The use of currency, by itself, does not necessarily imply economic behavior. Even if each user chooses only to "bid" a fraction of its job's tickets, that fraction controls directly the proportion of the occupancy the job can expect. That is, lottery scheduling as described here is a powerful and effective mechanism for establishing execution priorities, but a system that uses it cannot be expected necessarily to act as an economic one.

This example also illustrates another potential point of confusion with respect to resource brokering schemes. In particular, the use of bidding as a negotiative mechanism does not, by itself, ensure that economic principles govern the behavior of an allocation scheme. Some resource brokering schemes [7, 1] are based on the idea that applications announce their computational requirements to a broker, and resource providers "bid" (based on their processor type, load, etc.) for the outstanding jobs. The job is then "awarded" to the resource that is best able to execute it according to some affinity function. Notice that the decision to acquire a resource is not dependent on its worth as measured by its price, nor does the decision to supply a resource hinge on the possible affinities it may have for some jobs.

However, a slight modification to the lottery-scheduling example results in a system that obeys (and can be analyzed using) economic principles. If each job is allocated some number of tickets (even based on who its user is) and the user is allowed to *retain* unused tickets from one job for use with another, then the overall system can be treated as an economic one. Supply (CPU time slices, in this case) is constant, but the fraction of CPU time that a job may purchase is now a function of the demand for that fixed supply.

> **Assumption #3**: Relative worth is accurately measured only when market equilibrium is reached.

Another way of stating this assumption is that a given price is an accurate measure of value only when the available supply of a resource at that price is equal to the demand for it at that price. The relationship of equilibrium to value is important when evaluating the efficiency of a particular economic system. If the price of a resource is very high, causing little demand for it (i.e. the market is in equilibrium at a high price point), the allocation mechanism is working efficiently from an economic point of view even if many of the resources may be idle. In this case, price accurately captures the value of the resource as measured by the willingness of a producer to sell it *and* the willingness of a consumer to buy it. Now consider what happens if the same amount of resource is available, but the price that is specified (due to the workings of some price-setting mechanism — see the next subsection) is lower than the equilibrium price. In this case, demand would exceed supply, leaving some consumers unsatisfied.

2.1 Price-Setting Mechanisms: Commodities Markets and Auctions

Broadly speaking, there are two categories of mechanism for setting prices: commodities markets and auctions. In both formulations, consumers and producers appeal to a trusted third party to mediate the necessary transactions. In a commodities market setting, the third party (often termed "the market") sets a price for a resources and then queries both producers and consumers for a willingness to sell and buy respectively at that price. Those wishing to participate agree to transact business at the given price point and an exchange of currency for resource takes place. The market observes the unsatisfied supply or demand and uses that information (as well as other inputs) to set a new price. Price setting and transactions may occur in distinct stages, or may be concurrent and asynchronous. The key feature, however, that distinguishes a commodity market from an individual auction is that the consumer does not purchase a "specific" commodity, but rather takes one of many equivalents. For example, in the soybean market, a buyer does not purchase a specific lot of soybeans from a specific grower, but rather some quantity of soybeans of a given quality from the market.

Alternatively, prices may be set through an auction. In this case, the third party (termed "the auctioneer") collects resources and bids, but determines the sale of an individual resource (or resource bundle) based on the bids. Only one bidder ("the winner") is awarded a resource per auction round, and the process is repeated for each available resource.

Taken another way, commodity markets and auctions represent two ends of a spectrum of market formulations. On the commodity market end, an attempt is made to satisfy *all* bidders and sellers at a given price. At the other end — the auction end — *one* bidder and seller is satisfied at a given price. Obviously, it is possible to consider market organizations that are between the extremes. While there may be limited theoretical value in doing so, from an implementation perspective it may be that satisfying some limited number of transactions between price setting "rounds" is a useful organization. While we have studied the feasibility of commodity markets and auctions in Grid settings (see Section 3) in the extreme, much work addressing the practical implications of different market organizations remains.

2.2 Pricing Functions

The way in which a price is determined for a particular transaction is, in some sense, independent of how the market is organized. In auction settings, a variety of price determining rules can be used such as an *English auction*, *Dutch auction*, *First-price-sealed-bid*, *Vickery*, etc. Each of these mechanisms is designed to maximize a different objective function with respect to the market. For example, the most familiar auction style is the English auction in which an auctioneer solicits ascending public bids for an item (i.e. each bid must be larger than the current largest) until only one bidder remains. This style of auction is designed to maximize the revenue available to the seller by drawing out the highest possible price for the item on sale. Notice, however, that revenue maximization in this way does not guarantee that the price paid accurately represents the value of the good. In particular, English auctions are susceptible to the "winner's curse" in which the buyer typically over bids for the good. That is, statistically, the winning bid for a good higher than the good it actually worth since the distribution of bids is likely to straddle the "true" value.

In a Grid setting, the problem with this over pricing is that it may translate to inefficiency. Grid application users at a scientific computer center probably do not want their resource allocation mechanisms wasting resources to over bid each other. If currency can be exchanged for resources, than excess currency expended is equivalent to lost resource time. As a result, frugal users will either collude to keep prices low (which may extinguish supply) or will simply refuse to participate. In either case, users are "encouraged" by the market to understate the value of the resource because of the possibility of over paying. Worse, the market may fluctuate wildly as users search for the true value of a resource with their respective bids.

From an economic perspective, the term "incentive compatibility" refers to the incentive a market participant has to state his or her valuations truthfully. English auctions are not incentive compatible with respect to buyers. Second-price Vickery auctions, however, are buyer incentive compatible. In a Second-price Vickery auction (or Vickery auction, for short) each buyer submits a single sealed bid that is hidden from all other buyers. The winning bidder pays the price specified by the second highest bid. For example, if a CPU were being auctioned to a set of Grid schedulers, and the highest bid were $G100 (100 "Grid bucks"), but the second highest bid were $G90, the bidder who bid $G100 would win, but only pay $G90. It can be proved that this scheme induces buyers to bid truthfully since the possibility of winner's curse has been eliminated.

Unfortunately, it can also be proved that no auction price setting scheme is incentive compatible for both buyers and sellers. As such, proofs about the global stability of an auction-based Grid resource allocation scheme (or indeed any auction-based scheme, Grid or otherwise) remain elusive. Auctions "work" in the sense that they are used in many different economic settings. They are attractive because they are easy to understand and efficient to implement (given a single, centralized auctioneer configuration). They are not, however, easy to analyze. Indeed, the problem of determining an optimal winner in combinatorial auction settings (where buyers and sellers trade currency for combinations of items) is, itself, NP-complete [45, 17]. By itself, this result does not rule out auction systems as a market formulation for Grid settings, but it it almost certainly the case that Grid resource allocators (human or automatic) will need to purchase different resources. Doing so from a number of single-unit auctions can lead to inefficiency, as we describe later. If combinatorial auctions are used, then the auction protocols that can be brought to bear offer only heuristic guarantees of optimality at best.

For commodity market formulations, the theoretical results are more attractive, but the implementation complexity is higher and the intuitive appeal less. Note that for our purposes, we will assume that *all agents, producers and consumers, in the economy are* price takers – *that is, none of the agents represents a large enough market share to affect prices unilaterally, and therefore their decisions are made in response to a price which is given to them.* This assumption, also called "perfect competition" in the literature, is far from harmless, but it typically holds, at least approximately, in large-scale market economies.

The basic result of interest is a rather old one from general equilibrium theory. It says that, for a market in which multiple goods are exchanged, if the aggregate functions governing supply and demand are homogeneous, continuous, and obey Walras' Law, then there exists a way to assign a price to each good so that the *entire* market is brought into equilibrium. A more complete technical description of these assumptions can be found in [46] — we will attempt only an outline here. Homogeneity is the property that the difference between demand and supply at any given price point (termed *excess demand*) is not a function of the units. (In other words, the excess demands will be the same whether the prices are reported in Grid bucks, Grid sawbucks, Grid talents, or Grid dimes.) Continuity is the mathematical definition. Finally, Walras' Law can be roughly paraphrased as "the sum of the excess demands is zero." That is, for any given set of prices, the total value supplied by all producers in the economy is equal to the total value demanded. (This may seem a bit strange if one does not bear in mind that we are treating our currency as just another commodity. Thus a possible action of a "consumer" is to provide a supply of currency whose value is equal to the sum of the values of the commodities that this consumer demands.)

Under these assumptions, several algorithmic methods have been proposed that provably will produce a sequence of prices converging to an equilibrium [35, 23, 14, 36]. From the perspective of Grid resource allocation (but also, perhaps, in real-world markets) the biggest theoretical drawback is the need for the excess demand functions to be continuous. Clearly, supply, demand, and price are discrete quantities, regardless of their units. Further, there is an implementation issue concerning the realization of commodities markets for the Computational Grid in that the price-setting algorithms assume access to the global excess demand functions and possibly their partial derivatives. The Newton-type algorithm proposed by Smale in [36], which can compute an equilibrium price with very low computational complexity, for example, needs the partial derivatives for all excess demand functions as a function of price. In a Grid economy, it is unlikely that individual users will even be able to state their own excess demand functions reliably. If they could, it is not clear that these functions would be time invariant, or that they could be aggregated effectively into global functions for the market as a whole. For these reasons, little work has focused on the use of general equilibrium theory in computational economic settings.

On the other hand, one might reasonably expect that in a large Grid economy, the aggregate excess demand functions would become "continuous enough" in the sense that the granularity caused by indivisibility of commodities is small compared to the magnitudes of these functions. Also, there are well-understood "quasi-Newton" or secant methods which have good numerical stability properties and operate in the absence of explicit knowledge about partial derivatives.

It is another question whether a given (approximate) equilibrium that we compute satisfies any desirable optimality conditions which can be formulated (or even observed). This is a difficult question, and even in an economy which perfectly satisfies our above criteria, the best that we can say is that the equilibrium is *Pareto optimal* in the sense that no change in price can occur without making at least one of the agents worse off (in terms of that agent's utility). Note, then, that the optimality properties of the market equilibrium tend to be "spread out" over the whole economy rather than seeking, say, to maximize cash flow, which is in the interest of the producers. One might hope that a market equilibrium is one that maximizes total utility, but we cannot say this. In any event, the concept of "total utility" is difficult to define in the absence of a well-defined unit of utility ("util"). Moreover, different equilibria will arise from different starting points (initial allocations) of the agents, and some of these may be judged under some criteria to be socially "better" than others. This said, however, it is natural to look at market equilibrium, which after all arises from the desires of the agents, as a useful benchmark for the performance of an economy.

3 An Example Study: G-commerce

As mentioned previously, using economic principles as the basis for Grid resource allocation has several advantages. In this section, we describe a previous study (detailed more completely in [46]) that illustrates two: the opportunity to leverage rigorous economic theory profitably, and the "natural" fit resource economies seem to have intuitively for Grid systems.

The goal of this work is to investigate the feasibility and potential efficiency of different market formulations for Grid resource economies. *G-commerce* refers to the economic mechanisms *and* policies that will need to be put in place to create an effective Grid allocation system. Much of the work in Grid computing that comes before it or is concomitant with it [9, 7, 16, 1] has been focused on the mechanisms necessary to support computational Grid economies. While these mechanisms (many of which take the form of elaborate brokering systems) will assuredly play an important role, G-commerce attempts to study both the necessary mechanisms and the effect of different economic policies in Grid settings.

As an example, consider the potential trade-offs between an auction-based economy and one based on a commodities market approach. Auction systems have been widely studied in other computational settings [40, 32, 12] and as discussed previously, and much of what is known about them rigorously is restricted to a microeconomic level. Alternatively, a commodities market approach has strong macroeconomic properties but important microeconomic characteristics such as incentive compatibility are often taken for granted.[1] The following study attempts to address two questions about Grid performance under different market formulations (i.e. auctions and commodity markets).

The first question is: *Given automatic program schedulers that can react at machine speeds such as those described in [4, 29, 11], what is the effect on resource allocation stability of auctions versus commodities markets?* If a particular economic formulation results in allocators constantly changing their allocation decisions, the overall system will waste much of its available resource on the work necessary to constantly adjust resource allocations.

The second question is: *What is the effect of choosing a particular market formulation on resource utilization?* While it may be necessary for the performance evaluation community to develop new metrics for measuring Grid efficiency, and these metrics may ultimately rely on computational economic principles, it is certainly enlightening to assess the potential impact of a computational economy in terms of well-understood metrics like utilization.

Our initial study uses simulation. While, we are currently developing a methodology for validating these simulation results empirically, we motivate the credibility of the study in terms of the issues discussed in Section 2 of this chapter.

3.1 Producers and Consumers

To compare the efficacy of commodities markets and auctions as Grid resource allocation schemes, we define a set of simulated Grid producers and consumers representing resource providers and applications respectively. We then use the same set of producers and consumers to compare commodity and auction-based market settings.

We simulate two different kinds of producers in this study: producers of CPUs and producers of disk storage. That is, from the perspective of a resource market, there are two kinds of resources within our simulated Grids: CPUs and disks (and therefore, counting currency, our market has three commodities). While the results should generalize to include a variety of other commodities, networks present a special problem. Our consumer model is that an application may request a specified amount of CPU and disk (the units of which we discuss below) and that these requests may be serviced by any provider regardless of

[1] Generally, such microeconomic considerations are well-behaved for an idealized market economy; in fact, it is proven in [21] that a market mechanism, *i.e.* a way of redistributing goods, is incentive compatible for all agents if and only if it is perfectly competitive.

location or network connectivity. Since network links cannot be combined with other resources arbitrarily, they cannot be modeled as separate commodities. We believe that network cost can be represented in terms of “shipping” costs in more complicated markets, but for the purposes of this study, we consider network connectivity to be uniform.

3.1.1 CPU Producer Model

In this study, a CPU represents a computational engine with a fixed dedicated speed. A CPU producer agrees to sell to the Grid some number of fixed “shares” of the CPU it controls. The real-world scenario for this model is for CPU owners to agree to host a fixed number of processes from the Grid in exchange for Grid currency. Each process gets a fixed, pre-determined fraction of the dedicated CPU speed, but the owner determines how many fractions or “slots” he or she is willing to sell. For example, in our study, the fraction is 10% so each CPU producer agrees to sell a fixed number (less than 10) of 10%-sized slots to the Grid. When a job occupies a CPU, it is guaranteed to get 10% of the available cycles for each slot it consumes. Each CPU, however, differs in the total number of slots it is willing to sell.

To determine supply at a given price-point, each CPU calculates

$$mean\_price \quad = \quad revenue/now/slots \tag{1}$$

where *revenue* is the total amount of Grid currency (hereafter referred to as \$G which is pronounced “Grid bucks”), *now* is an incrementing clock, and *slots* is the total number of process slots the CPU owner is willing to support. The *mean_price* value is the average \$G per time unit per slot the CPU has made from selling to the Grid. In our study, CPU producers will only sell if the current price of a CPU slot exceeds the *mean_price* value, and when they sell, they sell all unoccupied slots. That is, the CPU will sell all of its available slots when it will turn a profit (per slot) with respect to the average profit over time.

3.1.2 Disk Producer Model

The model we use for a disk producer is similar to that for the CPU producer, except that disks sell some number of fixed-sized “files” that applications may use for storage. The *mean_price* calculation for disk files is

$$mean\_price \quad = \quad revenue/now/capacity \tag{2}$$

where *capacity* is the total number of files a disk producer is willing to sell to the Grid. If the current price for a file is greater than the *mean_price*, a disk producer will sell all of its available files.

Note that the resolution of CPU slots and file sizes is variable. It is possible to make a CPU slot equivalent to the duration of a single clock cycle, and a disk file be a single byte. Since our markets transact business at the commodity level, however, we hypothesize that any real implementation for the Grid will need to work with larger-scale aggregations of resources for reasons of efficiency. For the simulations described in Section 4 we choose values for these aggregations that we believe reflect a market formulation that is currently implementable.

3.1.3 Consumers and Jobs

Consumers express their needs to the market in the form of jobs. Each job specifies both a size and an occupancy duration for each resource to be consumed. Each consumer also sports a budget of $G (pronounced "Grid bucks") that it can use to pay for the resources needed by its jobs. Consumers are given an initial budget and a periodic allowance, but they are not allowed to hold $G over from one period until the next. This method of budget refresh is inspired by the allocation policies currently in use at the NSF Partnerships for Advanced Computational Infrastructure (PACIs) [27, 25]. At these centers, allocations are perishable.

When a consumer wishes to purchase resources for a job, it declares the size of the request for each commodity, but not the duration. Our model is that job durations are relatively long, and that producers allow consumers occupancy without knowing for how long the occupancy will last. At the time a producer agrees to sell to a consumer, a price is fixed that will be charged to the consumer for each simulated time unit until the job completes.

For example, consider a consumer wishing to buy a CPU slot for 100 minutes and a disk file for 300 minutes to service a particular job. If the consumer wishes to buy each for a particular price, it declares to the market a demand of 1 CPU slot and 1 disk slot, but does not reveal the 100 and 300 minute durations. A CPU producer wishing to sell at the CPU price agrees to accept the job until the job completes (as does the disk producer for the disk job). Once the sales are transacted, the consumer's budget is decremented by the agreed-upon price every simulated minute, and each producer's revenue account is incremented by the same amount. If the job completes, the CPU producer will have accrued 100 times the CPU price, the disk producer will have accrued 300 times the disk price, and the consumer's budget will have been decremented by the sum of 100 times the CPU price and 300 times the disk price.

In defining this method of conducting resource transactions, we make several assumptions. First, we assume that in an actual Grid setting, resource producers or suppliers will commit some fraction of their resources to the Grid, and that fraction is slowly changing. Once committed, the fraction "belongs" to the Grid so producers are not concerned with occupancy. This assumption corresponds to the behavior of some batch systems in which, once a job is allowed to occupy its processors, it is allowed to run either until completion, or until its user's allocation is exhausted. Producers are concerned, in our models, with profit and they only sell if it is profitable on the average. By including time in the supply functions, producers consider past occupancy (in terms of profit) when deciding to sell. We are also assuming that neither consumers nor producers are malicious and that both honor their commitments (i.e. perfect competition). In practice, this requirement assuredly will be difficult to enforce. However, if consumers and producers must agree to use secure authentication methods and system-provided libraries to gain access to Grid resources, then it may be possible to approximate.

This last assumption, however, illustrates the weakness of a commodities market approach in terms of incentive compatibility. An incentive compatible auction would naturally induce consumers to act "fairly" because it would be in their best interest to do so. In a commodities market setting, the system would need to rely on external policies and mechanisms to enforce good consumer behavior (e.g. non-collusion). A key realization for the Grid, however, is that such a requirement already exists for security. For example, it is quite difficult to ensure that collaborating users do not share user-ids and passwords. The primary method of enforcement is policy based rather than mechanism based. That is, users have their access privileges revoked if they share logins as a matter of policy rather than by some automatic mechanism.

3.1.4 Consumer Demand

The consumer demand function is more complex than the CPU and disk supply functions. Consumers must purchase enough CPU and disk resources for each job they wish to run. If they cannot satisfy the request for only one type, they do not express demand for the other. That is, the demand functions for CPU and disks are strongly correlated (although the supply functions are not). This relationship between supply and demand functions constitutes the most difficult of market conditions. Most theoretical market systems make weaker assumptions about the difference in correlation. By addressing the more difficult case, we believe our work more closely resembles what can be realized in practice.

To determine their demand at a given price, each consumer first calculates the average rate at which it would have spent $G for the jobs it has run so far if it had been charged the current price. It then computes how many $G it can spend per simulated time unit until the next budget refresh. That is, it computes

$$\begin{aligned} avg\_rate &= \frac{\sum_i total\_work_i * price_i i}{now} && (3) \\ capable\_rate &= \frac{remaining\_budget}{(refresh - now)} && (4) \end{aligned}$$

where $total\_work_i$ is the total amount of work performed so far using commodity i, $price_i$ is the current price for commodity i, $remaining\_budget$ is the amount left to spend before the budget refresh, $refresh$ is the budget refresh time, and now is the current time. When $capable\_rate$ is greater than or equal to avg_rate, a consumer will express demand.

Unlike our supply functions, the consumer demand function does not consider past price performance directly when determining demand. Instead, consumers using this function act opportunistically based on the money they have left to spend and when they will receive more. They use past behavior only as an indication of how much work they expect to introduce and buy when they believe they can afford to sustain this rate.

Consumers, in our simulations, generate work as a function of time. We arbitrarily fix some simulated period to be a "simulated day." At the beginning of each day, every consumer generates a random number of jobs. By doing so, we hope to model the diurnal user behavior that is typical in large-scale computational settings. In addition, each consumer can generate a single new job every time step with a pre-determined probability. Consumers maintain a queue of jobs waiting for service before they are accepted by producers. When calculating demand, they compute *avg_rate* and *capable_rate* and demand as many jobs from this queue as they can afford.

To summarize, for our G-commerce simulations:

- All entities except the market-maker act individually in their respective self-interests.
- Producers consider long-term profit and past performance when deciding to sell.
- Consumers are given periodic budget replenishments and spend opportunistically.
- Consumers introduce work loads in bulk at the beginning of each simulated day, and randomly throughout the day.

We believe that this combination of characteristics captures a reasonable set of producer and consumer traits as evidenced by the current allocation and usage patterns at large-scale compute centers. It does not, however, attempt to capture what the behavior Grid users in an e-commerce or peer-to-peer setting. This narrowing of focus is an important one. It may be that different producer and consumer settings will require different market formulations — clearly an avenue for future study.

3.2 Commodities Markets

Our model is an example of an *exchange economy*, that is, a system involving *agents* (producers and consumers), and several commodities. Each agent is assumed to control a sufficiently small segment of the market. In other words, the individual behavior of any one agent will not affect the system as a whole appreciably. In particular, prices will be regarded as beyond the control of the agents. Given a system of prices, then, each agent decides upon a course of action, which may consist of the sale of some commodities and the purchase of others with the proceeds. Thus we define supply and demand functions for each commodity, which are functions of the aggregate behavior of all the agents. These are determined by the set of market prices for the various commodities.

The method for price determination is based on Smale's technique [36] for finding general market equilibria. A motivation for its choice and a complete explanation of its application are available from [46]. Here, we limit or description to a discussion of the implementation issues as they pertain to the economic principles embodied by equilibrium theory.

In particular, it is not possible to use Smale's method directly for a number of reasons. First, any actual economy is inherently discrete, so the partial derivatives that the method requires do not exist, strictly speaking. Second, given the behavior of the producers and consumers described above, there are threshold prices for each agent that bring about sudden radical changes in behavior, so that a reasonable model for excess demand functions would involve sizeable jump discontinuities. Finally, the assumptions in Smale's model are that supply and demand are functions of price only and independent of time, whereas in practice there are a number of ways for supply and demand to change over time for a given price vector.

Obtaining the partial derivatives necessary to carry out Smale's process in an actual economy is impossible; however, within the framework of our simulated economy, we are able to get good approximations for the partials at a given price vector by polling the producers and consumers. Starting with a price vector, we find their preferences at price vectors obtained by fixing all but one price and varying the remaining price slightly, thus achieving a "secant" approximation for each commodity separately; we then substitute these approximations for the values of the partial derivatives in the matrix $D_{\mathbf{z}}(\mathbf{p})$, discretize with respect to time, solve for a price vector, and iterate. We will refer, conveniently but somewhat inaccurately, to this price adjustment scheme as *Smale's method.*

The First Bank of G: The drawback to the above scheme is that it relies on polling the entire market for aggregate supply and demand repeatedly to obtain the partial derivatives of the excess demand functions. If we were to try and implement Smale's method directly, each individual producer and consumer would have to be able to respond to the question "how much of commodity x would you buy (sell) at price vector $\mathbf{p}$?" In practice, producers and consumers may not be able to make such a determination accurately for all possible values of $\mathbf{p}$. Furthermore, even if explicit supply and demand functions are made into an obligation that all agents must meet in order to participate in an actual Grid economy, the methodology clearly will not scale. For these reasons, in practice, we do not wish to assume that such polling information will be available.

A theoretically attractive way to circumvent this difficulty is to approximate each excess demand function z_i by a polynomial in $p_1, p_2, ..., p_n$ which fits recent price and excess demand vectors and to use the partial derivatives of these polynomials in place of the actual partials required by the method. In simulations, this method does not, in general, produce prices which approach equilibrium. The *First Bank of G* is a price adjustment scheme which both is practicable and gives good results; this scheme involves using *tâtonnement* [43] until prices get "close" to equilibrium, in the sense that excess demands have sufficiently small absolute value, and then using the polynomial method for "fine tuning." Thus, the First Bank of G approximates Smale's method but is implementable in real-world Grid settings since it hypothesizes excess demand functions and need not poll the market for them. Our experience is that fairly high-degree polynomials are required to capture excess demand behavior with the sharp discontinuities described above. For all simulations described in Section 4, we use a degree 17 polynomial.

While this analysis may seem quite detailed, it illustrates the linkage between theoretical economic results and their potential application to Grid economies. At the same time, the work also provides a practical implementation of a price-setting mechanism. This combination of theory and practice is critical to the realization of effective Grid economies.

3.3 Auctions

Auctions have been extensively studied as resource allocation strategies for distributed computing systems (e.g. [12, 40, 26, 6]). See [45] for a particularly illucidating discussion of the relevant issues.

When consumers simply desire one commodity, for example CPUs in Popcorn [26], auctions provide a convenient, straightforward mechanism for clearing the marketplace. However, the assumptions of a Grid Computing infrastructure pose a few difficulties to this model. First, when an application (the consumer in a Grid Computing scenario) desires multiple commodities, it must place simultaneous bids in multiple auctions, and may only be successful in a few of these. To do so, it must expend currency on the resources that it has obtained while it waits to obtain the others. This expenditure is wasteful, and the uncertain nature of auctions may lead to inefficiency for both producers and consumers.

Second, while a commodities market presents an application with a resource's worth in terms of its price, thus allowing the application to make meaningful scheduling decisions, an auction is more unreliable in terms of both pricing and the ability to obtain a resource, and may therefore result in poor scheduling decisions and more inefficiency for consumers.

To gain a better understanding of how auctions fare in comparison to commodities markets, we implement the following simulation of an auction-based resource allocation mechanism for Computational Grids. At each time step, CPU and disk producers submit their unused CPU and file slots to a CPU and a disk auctioneer. These are accompanied by a minimum selling price, which is the average profit per slot, as detailed in Section 3.1.1 above. Consumers use the demand function as described in Section 3.1.3 to define their bid prices, and as long as they have money to bid on a job, and a job for which to bid, they bid on each commodity needed by their oldest uncommenced job.

Once the auctioneers have received all bids for a time step, they cycle through all the commodities in a random order, performing one auction per commodity. In each auction, the highest-bidding consumer gets the commodity if the bid price is greater than the commodity's minimum price. If there is a second-highest bidder whose price is greater than the commodity's minimum price, then the price for the transaction is the second-highest bidder's price. If there is no such second-highest bidder, then the price of the commodity is the average of the commodity's minimum selling price and the consumer's bid price. When a consumer and commodity have been matched, the commodity is removed from the auctioneer's list of commodities, as is the consumer's bid. At that point, the consumer can submit another bid to that or any other auction, if desired. This situation occurs when a consumer has obtained all commodities for its oldest uncommenced job, and has another job to run. Auctions are transacted in this manner for every commodity, and the entire auction process is repeated at every time step.

Note that this structuring of the auctions means that each consumer may have at most one job for which it is currently bidding. When it obtains all the resources for that job, it immediately starts bidding on its next job. When a time step expires and all auctions for that time step have been completed, there may be several consumers whose jobs have some resources allocated and some unallocated, as a result of failed bidding. These consumers have to pay for their allocated resources while they wait to start bidding in the next time step.

While the auctions determine transaction prices based on individual bids, the supply and demand functions used by the producers and consumers to set ask and bid price are the same functions we use in the commodities market formulations. *Thus, we can compare the market behavior and individual producer and consumer behavior in both auction and commodity market settings.* It is this comparison that forms the basis of the economic study. Since the individual producers and consumers use the same rules to decide when to supply and demand in each market setting, we can observe the effects of market formulation on the overall economy.

4 Simulations and Results

We compare commodities markets and auctions using the producers and consumers described in Section 3.1. Recall that our markets do not include resale components. Consumers do not make money. Instead, $G are given to them periodically much the in the same way that PACIs dole out machine-time allocations. Similarly, producers do not spend money. Once gathered, it is hoarded or, for the purposes of the economy, "consumed." We present the under-demand case which corresponds to a Grid economy in which the supply easily exceeds what is necessary to complete all the conumers' jobs. In the *over-demand* case (described in [46]), consumers wish to buy more resources than are available. New jobs are generated fast enough to keep all producers almost completely busy, thereby creating a work back-log.

Using our simulated markets, we wish to investigate three questions with respect to commodities markets and auctions.

1. *Do the theoretical results from Smale's work [37] apply to plausible Grid simulations?*

2. *Can we approximate Smale's method with one that is practically implementable?*

3. *Are auctions or commodities markets a better choice for Grid computational economies?*

Question (1) is important because if Smale's results apply, they dictate that an equilibrium price-point must exist (in a commodity market formulation), and they provide a methodology for finding those prices that make up the price-point. Assuming the answer to question (1) is affirmative, we also wish to explore methodologies that achieve or approximate Smale's results, but which are implementable in real Grid settings. Lastly, recent work in Grid economies [1, 16, 7] and much previous work in computational economic settings [13, 31, 5, 41] has centered on auctions as the appropriate market formulation. We wish to investigate question (3) to determine whether commodities markets are a viable alternative and how they compare to auctions as a market-making strategy.

4.1 Market Conditions, under-demand case

We present only the under-demand case from [46] (where maximum consumer demand is quickly extinguished by the maximum available supply) as an illustration of the results our simulation system has generated. A discussion of the over-demand case, particularly with respect to the presence of multiple equilibria, is important as well, but beyond the scope of this work. Again, the details are available from [46].

Figure 1 shows the CPU and disk prices for an unrealistic-but-correct implementation of Smale's method in our simulated Grid economy over $10,000$ time units. The implementation is unrealistic (even though it exists) because it relies on the ability to stop the entire economy and poll all producers and consumers completely for their individual supply and demand functions. It is, therefore, able to verify that the discretization that is necessary to represent Grid economies does not "break" the effectiveness of Smale's method. It does not, however, imply that Smale's method is implementable in realistic way for Grid settings. A diurnal cycle of consumer job submission that we deliberately introduce is evident from the price fluctuations. Every 1440 "minutes" each consumer generates between 1 and 100 new jobs causing demand and prices to spike. However, Smale's method is able to find an equilibrium price for both commodities quickly, as is evidenced in Figure 2. Notice that the excess demand spikes in conjunction with the diurnal load, but is quickly brought near zero by the pricing shown in Figure 1 where it hovers until the next cycle. The "speed" with which Smale's method finds equilibrium is predicted by his work, but the simulation verifies that discretization does not prevent the pricing mechanism from responding faster than the agents in the market.

In Figure 3 we show the pricing determined by our engineering approximation to Smale's method — the First Bank of G. The First Bank of G pricing closely approximates the theoretically achievable results generated by Smale's method in our simulated environment. The Bank, though, does not require polling to determine the partial derivatives for the aggregate supply and demand functions. Instead, it uses an iterative polynomial approximation that

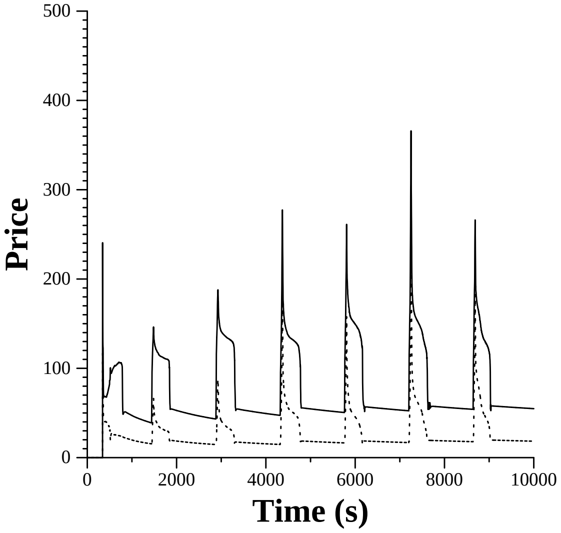

Figure 1: Smale's prices for the under-demand case. Solid line is CPU price, and dotted line is disk price in $G

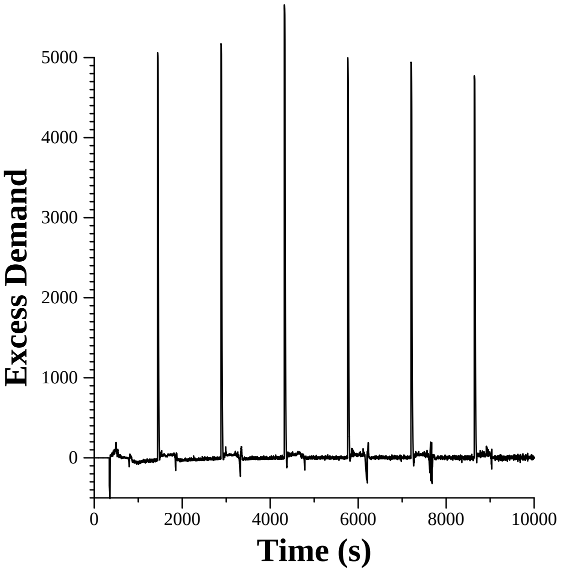

Figure 2: Smale's CPU excess demand for the under-demand case. The units are CPU slots.

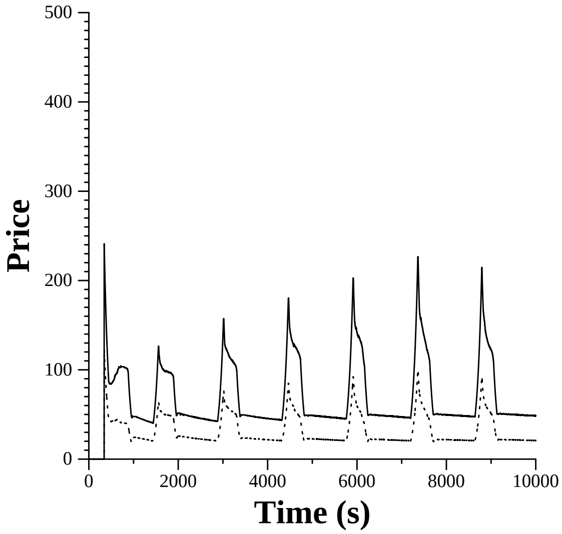

Figure 3: First Bank of G prices for the under-demand case. Solid line is CPU price, and dotted line is disk price in $G

it derives from simple observations of purchasing and consumption. Thus it is possible to implement the First Bank of G for use in a real Grid setting without polling Grid producers or consumers for their supply and demand functions explicitly.

The pricing determined by auctions is quite different, however, as depicted in Figures 4 and 5 (we show CPU and disk price separately as they are almost identical and obscure the graph when overlayed). In the figure, we show the average price paid by all consumers for CPU during each auction round. We use the average price for all auctions as being representative of the “global” market price. Even though this price is smoothed as an average (some consumers pay more and some pay less during each time step), it shows considerably more variance than prices set by the commodities market. The spikes in workload are not reflected in the price, and the variance seems to increase (i.e. the price becomes less stable) over time.

The use of an average price in these results further illustrates the difficulty in analyzing auction systems at the macroeconomic level. Each consumer, potentially, pays a unique price, making overall supply and demand relationships difficult to gauge. This drawback by itself, however, does not imply that auctions are ineffective — only that they are harder to treat analytically on a global level. Given that the average (i.e. smoothed) price is significantly more unstable in an auction setting, however, indicates that equilibrium theory (despite its own inherent drawbacks) is a potentially useful tool with which to build a Grid economy.

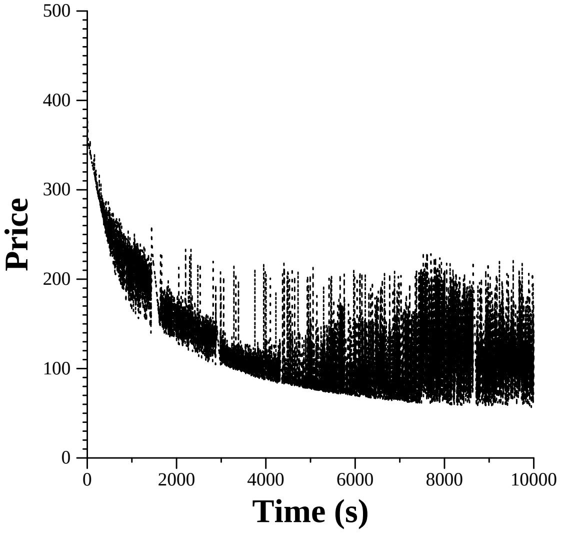

Figure 4: Auction prices for the under-demand case, average CPU price only, in $G

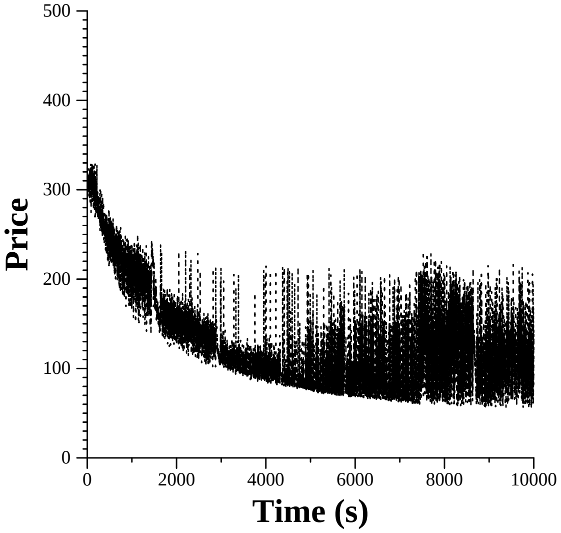

Figure 5: Auction prices for the under-demand case, average disk price only, in $G

efficiency metric	under-demand
Smale **consumer jobs/min**	0.14 j/m
B of G **consumer jobs/min**	0.13 j/m
auction consumer jobs/min	0.07 j/m
Smale **CPU** utilization %	60.7%
B of G **CPU** utilization %	60.4%
auction **CPU** utilization %	35.2%
Smale **disk** utilization %	54.7%
B of G **disk** utilization %	54.3%
auction **disk** utilization %	37.6%

Table 1: Consumer and Producer efficiencies

4.2 Efficiency

While commodities markets using Smale's method of price determination appear to offer better theoretical and simulated economic properties (equilibrium and price stability) than auctions do, we also wish to consider the effect of the two pricing schemes on producer and consumer efficiency. To do so, we report the average percentage of time each resource is occupied as a utilization metric for suppliers, and the average number of jobs/minute each consumer was able to complete as a consumer metric. Table 1 summarizes these values the under-demand case.

In terms of efficiency, Smale's method is best and the First Bank of G achieves almost the same results. Both are significantly better than the auction in all metrics except disk utilization in the over-demanded case [46]. Since CPUs are the scarce resource, disk price may fluctuate through a small range without consequence when lack of CPU supply throttles the system. The auction seems to achieve slightly better disk utilization under these conditions. In general, however, Smale's method and the First Bank of G approximation both outperform the auction in the simulated Grid setting.

5 Discussion

In computational economic terms, this study has various strengths and weaknesses. One of its most obvious weaknesses is that it is unvalidated. At the time of this writing, there are no available infrastructures for implementing general Grid economies of which we are aware. Systems such as Nimrod-G [1] and GrADSoft [3] implement various components, but as of yet no mechanism exists for empirical studies of incentivized users using arbitrary programs. There are two immediate problems with building such a system that are particular to Grid economics.

The first problem is that it is difficult to incentivize credibly "real" Grid consumers and producers in an experimental setting. Even if users are induced to participate, the decisions they make may be based on on the inducement and not on their perceived value in the system. For example, in the Laboratory for **M**iddleware and **A**pplications **Y**ielding **H**eterogeneous **Environments** for **M**etacomputing (MAYHEM) [22] at the University of California, Santa Barbara we have developed an infrastructure for implementing Grid resource economies that includes the First Bank of G as a price-setting mechanism. The system is compatible with a variety of Grid execution infrastructures including Globus [15], NetSolve [10], Condor [38], and Legion [20]. Designing effective empirical studies using the infrastructure has proved challenging. In particular, graduate students (when used as test subjects) are prone to skewing the results (either explicitly or inadvertently) according to the perceived wishes of their advisor. Our current work focuses on developing experiments that incentivize participants based on the value of resources to the computations at hand and not the computational economics results that may be generated.

A second related problem concerns starting and re-starting the economy. A key requirement for Grid economies in general, and Grid economics research in particular, is that Grid systems will need to be partially or entirely "reset." Software upgrades (particularly with respect to middleware services), security concerns, hardware maintenance, etc. can cause substantial reconfigurations. The economics research community has little guidance to offer with respect to starting an economy from scratch (or reset, in the case of the Grid). In an empirical setting, start-up can be critical since the amount of time that can be expected from test subjects may be limited. If the behavior under test will not emerge for months of years because the economy has been started in the wrong "place" it may not be extractable. Again, our early attempts to design credible empirical experiments have been plagued by this problem.

Another important consideration in any economic study concerns the availability of what are usually hidden functions in a "real-world" economy. In particular, some economic research hypothesizes the presence of utility functions which (in their simplest form) measure an agent's "happiness" in terms of other economic quantities. These abstractions (sometimes termed *utility functions*) are useful for lending theoretical rigor, but are often difficult or impossible to determine in any practical way.

For G-commerce, these functions take the form of the consumer and producer decision-making algorithms that control supply and demand. Recall, that our simulated consumers used a stochastic model that was biased toward opportunistic purchasing while producers were stochastic bargain hunters. We designed them based on our real-world experience with Grid users (and potential Grid users) in the PACI environment. Still, despite their non-linear and discontinuous definition, they remain a viable motivation for skepticism.

Notice, though, that even though the individual producer and consumer functions are ill-behaved with respect to the assumptions underlying general equilibrium theory, they compose in a way that allows the basic equilibrium results to emerge. This observation about the composition of agent behavior is a strong-point of the work.

6 Conclusions

In this chapter, we outline many of the features that characterize artificial resource economies as they pertain to the Computational Grid. Effective resource discovery and allocation mechanisms for Grid systems remain a critical area of research. By structuring such systems according to economic principles, it is possible to apply a considerable body of work from economics to analyze and predict Grid behavior. Doing so requires a keen understanding of the underlying assumptions governing allocation decisions.

In particular, it is important to understand, fundamentally, the relationship between price, supply, and demand that fluctuates as users acquire and release Grid resources. The choice of pricing protocol can also affect both the implementation complexity and the type of analysis that is possible. Auction systems, for example, are popular because they are easy to implement and well-understood in terms of their effect on the individual market participants. They are difficult to analyze globally, however, and the optimal use of various formulations (e.g. combinatorial auctions) result in the need to solve NP-complete problems. Alternatively, commodity markets offer attractive global properties such as stability at the expense of useful features at the microeconomic level (e.g. incentive compatibility).

As an example illustrating many of these issues, we present G-commerce — a set of implementable policies and mechanisms economies for controlling resource allocation Computational Grid settings. Using simulated consumers and producers we investigate commodities markets and auctions as ways of price setting. Under similar conditions, we examine the overall price stability, market equilibrium, producer efficiency, and consumer efficiency. The results of this study, that appeal in large part of the work of Smale [37] show that commodity markets (despite their potential drawbacks) outperform auctions in our simulated Grid settings.

Designing and executing an empirical verification of this work remains a work-in-progress. Early experiences with accurately incentivized consumers illustrates several important research questions. In particular, unlike real-world economies, Grid resource economies may need to be restarted. Little formal theory exists for the analysis of the start-up and shut-down properties of economic systems. As a result, Grid economies may require new theoretical results in economics to be truly effective.

Despite this risk, the intuitive appeal of economically based resource allocation mechanisms combined with the theoretical results that are available make computational economies a viable and active research area. Clearly, there is much more work to do.

References

[1] D. Abramson, J. Giddy, I. Foster, and L. Kotler. High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid? In *Proceedings*

of the International Parallel and Distributed Processing Symposium, May 2000.

[2] O. Arndt, B. Freisleben, T. Kielmann, and F. Thilo. Scheduling parallel applications in networks of mixed uniprocessor/multiprocessor workstations. In *Proceedings of ISCA 11th Conference on Parallel and Distributed Computing*, September 1998.

[3] F. Berman, A. Chien, K. Cooper, J. Dongarra, I. Foster, L. J. Dennis Gannon, K. Kennedy, C. Kesselman, D. Reed, L. Torczon, , and R. Wolski. The grads project: Software support for high-level grid application development. Technical Report Rice COMPTR00-355, Rice University, February 2000.

[4] F. Berman, R. Wolski, S. Figueira, J. Schopf, and G. Shao. Application level scheduling on distributed heterogeneous networks. In *Proceedings of Supercomputing 1996*, 1996.

[5] J. Bredin, D. Kotz, and D. Rus. Market-based Resource Control for Mobile Agents. Technical Report PCS-TR97-326, Dartmouth College, Computer Science, Hanover, NH, Nov. 1997.

[6] J. Bredin, D. Kotz, and D. Rus. Market-based resource control for mobile agents. In *Second International Conference on Autonomous Agents*, pages 197–204. ACM Press, May 1998.

[7] R. Buyya. Ecogrid home page `http://www.csse.monash.edu.au/~rajkumar/ecogrid/index.html`.

[8] R. Buyya. economygrid home page `http://www.computingportals.org/projects/economyManager.xml.html`.

[9] R. Buyya. *Economic-based Distributed Resource Management and Scheduling for Grid Computing*. PhD thesis, Monash University, 2002. `http://www.buyya.com/thesis/thesis.pdf`.

[10] H. Casanova and J. Dongarra. NetSolve: A Network Server for Solving Computational Science Problems. *The International Journal of Supercomputer Applications and High Performance Computing*, 1997.

[11] H. Casanova, G. Obertelli, F. Berman, and R. Wolski. The AppLeS Parameter Sweep Template: User-Level Middleware for the +Grid. In *Proceedings of SuperComputing 2000 (SC'00)*, Nov. 2000.

[12] B. Chun and D. E. Culler. Market-based proportional resource sharing for clusters. Millenium Project Research Report, `http://www.cs.berkeley.edu/~plink/papers/market.pdf`, Sep 1999.

[13] B. N. Chun and D. E. Culler. Market-based proportional resource sharing for clusters. Millenium Project Research Report, Sep. 1999.

[14] B. C. Eaves. Homotopies for computation of fixed points. *Mathematical Programming*, 3:1–22, 1972.

[15] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, 1997.

[16] I. Foster, A. Roy, and L. Winkler. A quality of service architecture that combines resource reservation and application adaptation. In *Proceedings of TERENA Networking Conference*, 2000.

[17] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions. In *Proceedings of Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

[18] J. Gehrinf and A. Reinfeld. Mars - a framework for minimizing the job execution time in a metacomputing environment. *Proceedings of Future general Computer Systems*, 1996.

[19] The dynamically-updated request online coallocator (duroc), 2002. available from `http://www.globus.org/duroc`.

[20] A. S. Grimshaw, W. A. Wulf, J. C. French, A. C. Weaver, and P. F. Reynolds. Legion: The next logical step toward a nationwide virtual computer. Technical Report CS-94-21, University of Virginia, 1994.

[21] L. Makowski, J. Ostroy, and U. Segal. Perfect competition as he blueprint for efficiency and incentive compatibility. *UCLA Economics Working Papers #745*, October 1995.

[22] The MAYHEM laboratory, UCSB. `http://pompone.cs.ucsb.edu`.

[23] O. H. Merrill. Applications and extensions of an algorithm that computes fixed points of certain upper semi-continuous point to set mappings. Ph.D. Dissertation, Dept. of Ind. Engineering, University of Michigan, 1972.

[24] H. Nakada, H. Takagi, S. Matsuoka, U. Nagashima, M. Sato, and S. Sekiguchi. Utilizing the metaserver architecture in the ninf global computing system. In *High-Performance Computing and Networking '98, LNCS 1401*, pages 607–616, 1998.

[25] The National Computational Science Alliance. `http://www.ncsa.edu`.

[26] N. Nisan, S. London, O. Regev, and N. Camiel. Globally distributed computation over the Internet — the POPCORN project. In *International Conference on Distributed Computing Systems*, 1998.

[27] The National Partnership for Advanced Computational Infrastructure. `http://www.npaci.edu`.

[28] D. Parkes and L. Ungar. Iterative combinatorial auctions: Theory and practice. In *Proceedings of 17th National Conference on Artificial Intelligence*, 2000.

[29] A. Petitet, S. Blackford, J. Dongarra, B. Ellis, G. Fagg, K. Roche, and S. Vadhiyar. Numerical libraries and the grid. In *Proceedings of SC01*, November 2001.

[30] R. Raman, M. Livny, and M. Solomon. Matchmaking: Distributed resource management for high throughput computing. In *Proc. 7th IEEE Symp. on High Performance Distributed Computing*, 1998.

[31] O. Regev and N. Nisan. The popcorn market - an online market for computational resources. First International Conference On Information and Computation Economies. Charleston SC, 1998. To appear.

[32] O. Regev and N. Nisan. The Popcorn market — online markets for computational resources. In *First International Conference On Information and Computation Economies*, Charleston, SC, 1998.

[33] T. Sandholm and S. Suri. Optimal clearing of supply/demand curves. In *Proceedings of 13th Annual International Symposium on Algorithms and Computation (ISAAC)*, 2002.

[34] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Cabob: A fast optimal algorithm for combinatorial auctions. In *Proc. of 17th International Joint Conference on Artificial Intelligence (IJCAI)*, 2001.

[35] H. Scarf. *The Computation of Economic Equilibria.* Yale University Press, 1973.

[36] S. Smale. Price adjustment and global Newton methods. *Frontiers of Quantitative Economics*, IIIA:191–205, 1975.

[37] S. Smale. Dynamics in general equilibrium theory. *American Economic Review*, 66(2):284–294, May 1976.

[38] T. Tannenbaum and M. Litzkow. The condor distributed processing system. *Dr. Dobbs Journal*, February 1995.

[39] H. Varian. *Broadband: Should We Regulate High-speed Internet Access?*, chapter The Demand for Bandwidth: Evidence from the INDEX Experiment. Brookings Institute, Washington, DC, 2002. James Alleman and Robert Crandall, editors.

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

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

[42] C. A. Waldspurger and W. E. Weihl. Lottery scheduling: Flexible proportional-share resource management. In *First Symposium on Operating Systems Design and Implementation (OSDI '94)*, pages 1–11, November 1994.

[43] L. Walras. *Eléments d'Economie Politique Pure*. Corbaz, 1874.

[44] M. Wellman. *Market-oriented Programming: Some Early Lessons*, chapter 4. World Scientific, 1996.

[45] M. Wellman, W. Walsh, P. Wurman, and J. MacKie-Mason. Auction protocols for decentralized scheduling, 1998.

[46] R. Wolski, J. Plank, T. Bryan, and J. Brevik. Analyzing market-based resource allocation strategies for the computational grid. *International Journal of High-performance Computing Applications*, 15(3), 2001.

[47] R. Wolski, J. S. Plank, J. Brevik, and T. Bryan. G-commerce: Market formulations controlling resource allocation on the computational grid. In *International Parallel and Distributed Processing Symposium (IPDPS)*, April 2001.