Pricing Computing Resources: Reading Between the Lines and Beyond

Junko Nakai*

Advanced Management Technology, Incorporated
NASA Advanced Supercomputing Division
NASA Ames Research Center

NAS Technical Report: NAS-01-010

Revised: January 15, 2002
First Version: November 2001

Abstract

Distributed computing systems have the potential to increase the usefulness of existing facilities for computation without adding anything physical, but that is realized only when necessary administrative features are in place. In a distributed environment, the best match is sought between a computing job to be run and a computer to run the job (global scheduling), which is a function that has not been required by conventional systems. Viewing the computers as "suppliers" and the users as "consumers" of computing services, markets for computing services/resources have been examined as one of the most promising mechanisms for global scheduling. We first establish how economics can contribute to scheduling. We further define the criterion for a scheme to qualify as an application of economics. Many studies to date have claimed to have applied economics to scheduling. If their scheduling mechanisms do not utilize economics, contrary to their claims, their favorable results do not contribute to the assertion that markets provide the best framework for global scheduling. We examine the well-known scheduling schemes, which concern pricing and markets, using our criterion of what application of economics is. For any of the schemes examined, we could not reach the conclusion that it is driven by economic motivations.

*Correspondence address: NASA Ames Research Center, Mail Stop 258-6, Moffett Field, CA 94305-1000 (phone: 650-604-4359, fax: 650-604-8669, e-mail: nakai@nas.nasa.gov). The author thanks the INR staff members for their comments.

1 Scheduling in Distributed Computing Systems

A distributed computing system brings about opportunities for enhancing the efficiency in computing, and hence, for increasing the value of existing facilities for computation. At the same time, a new system necessitates a new administrative structure for supporting its innovative features. One of the indispensable functions in a distributed environment, but not found in conventional systems, is to determine which computer in the system should run the computing job. For exploiting the full potential of a distributed system, which consists of computers with comparable but different capabilities and availability, a mechanism is required to gather and compare such information on various computers in the system and assign each job to the most appropriate computer. This mechanism, termed global scheduling, may take the most primitive form of providing the users with minimal hardware information of each computer.[1] Alternatively, it may be sophisticated enough to inform the users of the state of computer availability at the time of job submission, and concurrently, give priorities to jobs to which the service would be most valuable. Any global scheduling that utilizes the distributed nature of the systems would not be able to avoid the complexity that stems from the heterogeneity of the computers and the jobs. Viewing the computers as "suppliers" and the users as "consumers" of computing services, markets for computing services/resources have been examined as one of the most promising mechanisms for global scheduling.

We first establish how economics, a concept broader than markets, can contribute to scheduling. We do so by examining the factors that they have in common. We rely on the foundation of economics that economic agents are all utility maximizers (or all actions are aimed at attaining the state that is of greatest possible value to them) and, subsequently, we elucidate what constitutes the economic problem and when it comes into being. Where the economic problem leads us, and how pricing affects the outcomes, are briefly discussed. We further define the criterion for a scheme to qualify as an application of economics, which is based on the idea of economic problem. Many studies to date have claimed to have applied economics to scheduling. If their scheduling mechanisms do not concern the economic problem, then parts of scheduling functions that have been named bidding, auctions, etc. (which we regard as mechanisms that are meant to solve the economic problem) do not merit such naming; they simply obfuscate the issue.

We examine the well-known scheduling schemes, which concern pricing (the term primarily used before distributed computing systems became prominent) and markets (the term frequently used for distributed systems) with the use of our criterion of what application of economics is. Our conclusion is that none of the schemes examined can be said to make full use of economics. In most cases, the participation of truly economically-motivated agents in the scheme would not result in the desirable outcomes as often asserted by the studies. In other cases, it is unclear whether there were any economic decisions for the participating agents to make. The aim of this paper is to provide arguments for the above assertions, and ultimately, to aid in constructing a successful global-scheduling mechanism for distributed computing systems.

[1] The mechanisms of global scheduling, currently in use, are more elaborate than the most primitive one described above, but not by a big margin.

2 Scheduling and Economics

The core problem we face in scheduling computing jobs is exceedingly similar to that in the economy. If there were an unlimited amount of computing resources, jobs would be executed as soon as they are submitted, eliminating the need for scheduling. As there is a limit to the availability of resources, scheduling of computing jobs becomes a problem of allocating limited resources. We may envision a situation where the users of computers are required to give up the limited resources they are endowed with in exchange for access to the computing resources. Further, if the endowment could be used for more than one item or occasion, which result in outcomes differing in importance, then the allocation of computing resources is precisely the economic problem. Modern economics is recognized as the science that studies human behavior as a relationship between scarce means which have alternative uses, as asserted by Robbins [37].[2]

When the so-called pricing of computing resources started to attract increasing attention in the late 1960s, it was quickly acknowledged that pricing can be seen as a kind of scheduling mechanism and that both concern allocation of resources [35]. Pricing stepped into the limelight because the most common default scheduling mechanism, first-come, first-served, was often combined with additional rules, indicating that it alone was not meeting needs [35]. The advent of networks of computers did not alienate scheduling from economics, and scheduling for distributed computer systems has also been recognized as an activity under resource management [4, 26]. In the context of operation of fixed-capacity real-time databases [23], pricing has been termed as a device for natural admission control and overload management, and markets for computing resources have been embraced by many as the key components in the operation of distributed systems [3, 6, 7, 13, 18, 25, 28, 33, 41, 44, 45, 48, 49, 50].

2.1 The Economic Problem

An additional insight to the economic problem, as defined by Robbins, was provided by Hayek [14]. He made it clear that the problem was how to allocate resources, but not in an arbitrary manner. The goal is to allocate resources so that they are used in the best way possible. Inasmuch as the best uses are known only to individuals, the economic problem becomes "a problem of the utilization of knowledge which is not given to anyone in its totality" [14]. Certainly, this is also the problem posed to the group of users who are to share a set of computing facilities. The computing capabilities should be shared, and for that reason, information on the resource requirements for each job and the value of its successful execution, is required. Unfortunately, this information is usually available only to the user. In fact, utilization of users' private information has been recognized as an advantage of pricing over other types of scheduling ("the users themselves determine the value of immediate service and produce a service order upon which they mutually agree." [35]). In paraphrasing the economic problem in terms of utilization of private information, Hayek gave a refined definition of the ultimate objective of an economy.

[2]The basic activity in an economy is an exchange of goods and services. Whenever there is an exchange of goods or services, a price is established, which is simply the rate of exchange of goods or services involved. Thus, when we refer to one of the three, an economy, an exchange, and a price, the other two necessarily exist. Markets are forums for exchanges of goods and services, where exchanges are voluntarily initiated. Markets do not prevail in all types of economies, and economies are not synonymous with markets. For example, the planned economies of communist regimes were not market economies.

2.1.1 Beneficiaries of Gains from Trade

One of the potential advantages of pricing computing resources, i.e., utilization of private information, is linked to another aspect of pricing as the above discussion suggests: the possibility of maximizing the value of resources used (the value to the individual agents, and potentially, that to the system as a whole). However, Hayek did not spell out how we should solve a frequently encountered difficulty in resource allocation problems: whose best use should be realized, when one's best use becomes feasible only at the expense of another's.[3] The principle of maximization of the values to individuals leaves unanswered the question of division of gains from trade, or whose value should be maximized when that entails lowering of others' values.[4]

Maximization of the values to individuals (or users) is not necessarily equivalent to that of the value to the entire system (or administration). "The value of resource use" in the literature seems to have meant the sum of the users' and/or the administration's values in the allocation of computing resources; "[the] ordering [determined by prices] will maximize the value to the organization of the computing actually performed" [35], or "when preferences are uniformly expressed in terms of price, the strategy of allocating resources to those willing to pay the highest price insures the maximization of total utility realized by the use of these resources" [9]. Profit maximization or cost recovery (for which profits should be sought) as an overall goal for a computing system are cases of value maximization, where the administration's or the computer vendor's value is given priority.[5]

What is true, regardless of the answer to the question of division of gains from trade, is that pricing mechanism or policy is one of the most critical determinants in the division. When resources are allocated to the users who need them most, the users' value is maximized if they are provided free, other things being equal. The service provider's value is maximized if each user pays the amount equal to his/her willingness to pay for each unit of service, other things being equal. Various pricing policies may be adopted for meeting different goals.

2.1.2 Scarcity of Resources, Alternative Uses, and Value

We now turn to what is often overlooked when discussing a system that consists of value (or utility) maximizing agents: the importance of scarcity of resources, including budget, and that of the existence of alternative uses of resources in the formation of value itself. Robbins' definition of what economics is as a discipline, given above, involves "scarce means, which have alternative uses", indicating that there are always competing needs for resources and budget in economies. Such a variety of needs, which cannot all be fulfilled and each of which leads to an outcome that is different in importance, is what brings the economic problem into its existence.[6] If resources are obtainable whenever desired and/or their uses

[3] Theoretically speaking, when an economy consists of utility-maximizing agents, there often exist multiple Pareto-optimal solutions, if any exists. See, for example, Sections 17.D-F of *Microeconomic Theory* [29].

[4] In most circumstances, a trade or an exchange takes place only when all parties involved agree to it, and that is when all of them consider the trade beneficial to themselves. Thus, barring fraud and other similar schemes, a voluntary exchange always brings gains to all involved. However, how much each party would gain depends on the terms of trade, among others.

[5] If a computer is "on lease" to an institution and the computer vendor sells computing service, but not the computer itself, then the vendor would adopt a pricing policy which is usually aimed at profit maximization or cost recovery.

[6] Although Robbins' quote given earlier is considered the definition of modern economics, one necessary ingredient of the economic problem has been mentioned only earlier in his essay: difference in outcomes

result in outcomes identical in value, there would be no question of allocating the resources for their best use or to maximize value from use. In Nielsen's [35] words, "[a]n object takes on value only when it is scarce."

We distinguish two types of scarcity. It may take the straightforward form of a finite limit to the amount available, an amount smaller than is required to fulfill all needs, in which case the multiplicity of possible uses of the resource with outcomes that are distinguishable in importance, leads to the economic problem. If a limited amount of a resource is available during a particular time period (which cannot be used any other time) and that resource is designated for only one use, there is a unique possibility: allocation of the entire resource for that single use. In short, there is no economic problem under scarcity of resources if there are no alternative uses, which lead to different utility levels. Scarcity may take another form: a resource having dual effects, each of which is associated with a positive or a negative value to the users of that resource, depending on the amount allocated. The second type of scarcity also serves the function provided by alternative uses, which vary in importance of attainable outcomes, in the first kind. We will call the first kind of scarcity *absolute scarcity* and the second kind *inseparability scarcity*. Both absolute scarcity, coupled with the existence of alternative uses with different, resultant utility levels, and inseparability scarcity make it necessary for resource users to evaluate and compare the values of various allocations.

Pricing, therefore, plays an insignificant role in the exercise of value maximization, be it the value to individuals or that to the system, absent scarcity. For a multiple service-class network, Cocchi *et al.* [8] showed that prices did not have much effect in maximizing the overall value of computer usage if the work load was light. For fixed-capacity, real-time databases, Konana *et al.* [23] showed that there were bigger net system benefits and consumer surplus when job arrival rates were higher.[7] The economic problem comes into being only when resources have multiple positively and differently valued uses and the resources are scarce, or when resources' positively valued uses cannot be dissociated from negatively valued ones, where both effects vary in levels.

2.2 Price Related Issues in Scheduling

We briefly discuss two issues related to pricing: accounting and prioritization. Accounting in economic terms assumes the existence of price on every item to be accounted. In fact, the desire to record the usage of various computing resources in a consistent manner appears to be a drive behind pricing. Prioritization of jobs is often adopted together with a price for every priority level, and we argue below that prioritization is a special case of what is usually called pricing [9].

with respect to importance. Robbins wrote (the emphasis is from the original text): "But when time and the means for achieving ends are limited *and* capable of alternative application, *and* the ends are capable of being distinguished in order of importance, then behavior necessarily assumes the form of choice. Every act which involved time and scarce means for the achievement of one end involves the relinquishment of their use for the achievement of another. It has an economic aspect." We consider scarce means to include time, because we usually have time limits to whatever we do.

[7] A user was assumed to have an instantaneous value for a data request if granted, which depended on the realized data-flow rate. The net system-benefit was defined as the aggregate value to the users minus the delay cost to all users. Consumer surplus is the difference between the price consumers are willing to pay and the price actually paid, summed over all consumers.

2.2.1 Accounting

A trade would be complete upon exchange of goods and services, if parties involved are willing to give to the others exactly what is desired. However, if there is no such double coincidence of wants, a unit for assessing the trade or a medium of exchange (whose unit serves as that for trade assessment) would be necessary. If a physical exchange of the medium does not accompany the exchange of goods and services, a record of the exchange of goods and services must be kept: accounting.[8] It is often the case that several exchanges need to be summarized as if it were one, which requires a common unit for exchanges. In making economic decisions, a common unit is a must if there exists more than one input and the inputs could be combined in various ways to produce different outputs [34].

In computing systems, the need for a common unit for exchanges is a variant of the above situation. The users do not have anything to offer to the resource suppliers that is of direct value to them (e.g., computing resources that the suppliers could use themselves or sell). What the users are supposed to give to the suppliers in exchange for the resources is their resource-use allowance, and this condition necessitates a unit for converting allowance into charges for computing resources, or a common unit for both of them. The spirit of distributed computing systems is to confer users the ability to access any resource available in the system. This, in turn, requires conversion factors for resource-use allowance and resource-charge of all computers in the system, or a common unit for all of them. If there is only one type of resource to be charged for and the quality of that resource is uniform in the system, it suffices to define computer-use allowance in terms of unit of that resource (e.g., CPU time). When more than one type is to be accounted for (e.g., CPU time and memory) and different combination in quantities would be in use, accounting for overall use would require a unit which could measure uses of all resources concerned. Price is often considered a natural candidate for such a common unit.

Pricing, by itself, does not guarantee a common unit for resources, which may be used for trade assessment. A price of a certain good, in essence, is the amount of other goods, a unit of the good in question would fetch. If all resources are exchangeable with, at least, one common good, then that good may serve as the common unit. It is not imperative for the unit good to have any intrinsic value; a bill as we know today is a medium of exchange, and has little value of its own, that of a small piece of paper. Moreover, a common unit is not the only necessary element for establishing a good accounting system. A viable accounting system is reproducible, equitable, auditable, and understandable [9]. Moreover, users should be charged for the resources that were made unavailable because of their job execution, probably including the resources that were not used by the job but whose use was blocked [9, 16]. Any of these characteristics are not guaranteed by a common unit alone.

[8] Note that the contemporary definition of accounting in the field of accounting itself usually takes a more applied view, with the existence of shareholders in mind. For example, Horngren *et al.* [17] defined accounting to be "the process of identifying, recording, summarizing, and reporting economic information to decision makers." According to Stickney *et al.* [43], "[a]ccounting is a system for measuring the results of business activities and communicating those measurements to interested users." Edmonds *et al.* [11] defined an accounting event to be "an economic occurrence that causes changes in an enterprise's assets, liabilities, and/or equity." Sidebotham [40] drew on the history of accounting, and arrived at a broader definition: "At each stage of development men have employed accounting, according to their needs, to enumerate and control assets, as a reporting device for stewards and tax-gatherers, as evidence of trade, for the control of production, or the management of business."

2.2.2 Prioritization

We describe below the relationship between pricing and prioritization of jobs, a method often employed in scheduling. Priorities are set by service providers, and, are expressions of preferences by the providers. When a certain priority is chosen by a user, that is an expression of preferences by the user. Offering and accepting a certain price is analogous to setting and choosing a particular priority. When a price is offered, it reveals the preferences of the offerer, and when it is accepted, it reveals those of the accepting party. The proximity of pricing and prioritization in nature is also inferred from the fact that each priority is often distinguished from one another not only by policy but also by price.

The difference between the two lies in the accuracy of expression. A priority mechanism allows a choice among a finite number of priorities to the users, whereas a pricing mechanism usually allows any number to be picked. Therefore, under a priority mechanism, users would choose the priority level that matches their preferences best, which may not be optimal had other priorities been available. In terms of price, there can be exactly as many prices as there are priorities at one time. Under a pricing mechanism, however, the price that suits the preferences more closely could be chosen.

On priority mechanisms, Cotton [9] observed that they are based on the assumption that users are homogeneous and static, leading to suboptimal allocations in view of maximization of user utility. There are advantages to a priority mechanism, such as reduced disutility from a job's waiting in a queue, inexpensive cost of administration, and possible control of quality by the users [9]. A general pricing mechanism has the potential to differentiate resources more dynamically compared to a priority mechanism, but at the cost of increased computation for setting prices and uncertainty in price.

2.3 Application of Economics to Scheduling: Criterion

We have laid out above the problem an economy faces. Simply put, the basic economic activity is the exchange of possessions. The possibility of exchange exists when the exchange is positively valued by all parties involved. The potential value of an exchange is evaluated with care, only under the knowledge that one cannot obtain all that is desired, and such evaluations are borne privately. There is no economic problem for a computer user if the budget is more than one needs, or restricted to be used for a single service and that single use incurs only positive values.[9]

We do not assert that actual economic systems perfectly utilize private information to allocate resources for their best possible uses. The economic problem has given rise to numerous tools and institutional arrangements that we observe in the economy. However, many of them deliver not only the intended, desirable effects, but also undesirable ones. Stockmarket crashes, where a seemingly well-functioning market unpredictably ceases to be so, and environmental pollution, where firms take advantage of what is at their disposal for little cost, attest to this assertion. Hence, despite the similarity between the scheduling and the economic problems, we do not know *a priori* whether any economic device is preferable over other mechanisms for solving the problem of computing-resource allocation. In addition, any successful scheduling method will be characterized by the following features: ease of use and maintenance [16, 39], autonomy of local entities [5, 39], short calculation

[9] In theoretical economics, a representative consumer may have only one good to purchase. However, that is usually in a dynamic context, where a good bought at a certain time is distinguished from the same good to be bought later.

times for determining the schedule (in comparison to time intervals between job arrivals), transparency of mechanism to users [16, 35, 39], fairness as agreed by the users concerned [39], and effectiveness of its user education program [35].

In the following sections, we analyze various pricing schemes for computing resources, with the existence of the economic problem as the necessary condition for determining whether a scheme is an application of economics. When the participants in the scheme maximize utility (or equivalently, value), given constraints, by choosing among needs that cannot be simultaneously fulfilled and are different in achievable utility, the economic problem exists, and therefore, the necessary condition is satisfied for calling the scheme an application of economics to scheduling.[10] We divide the necessary condition for any scheme to qualify as an application of economics into: participating agents' utility maximizing behavior; scarcity of resources; (either of the two scarcities, as described earlier in the section) and, the existence of alternative uses (to be precise, the existence of alternative allocation levels, in case of inseparability scarcity), whose resultant utilities are not identical. If a computing-service provider interacts with users in the process of determining service allocation, the criterion must be satisfied not only by the users but also by the provider. Roughly speaking, a scheme is *incentive incompatible* when the desired outcome cannot be attained through utility-maximizing behavior of the participants.[11]

As is clear from the above, the knowledge of participating agents' utility functions is indispensable for checking whether the necessary condition is met. This requirement is a drawback for schemes which do not employ artificial agents; it is difficult to define utility functions for human users, while artificial agents cannot do without well-defined rules for their behavior. Our focus is on the most recent development in the use of economics in scheduling, and we examine the latest, representative schemes, all of which involve or strongly suggest involvement of artificial agents. We also examine pricing examples for non-distributed systems from the pre-1990 period, which do not make use of artificial agents.

3 History of Pricing Computing Resources

3.1 Functions of Pricing: from late 1960s to mid-1980s

During the early days of computing, most of the computing processes were sequential and in batches. Hence, the possible functions of pricing were narrowed to user-initiated dispersion of job submission over time, profit maximization, and cost recovery.[12] In the case of dispersion of job submission over time, the ultimate, but implicit, goal was to enhance the value of computing resources by differentiating resources based partly on user information revealed shortly before their possible execution.

Among the first efforts to price the computing resources, the case reported by Sutherland in 1968 [46] has served as the focal point [18, 35, 48]. Sutherland's scheme aimed at

[10] When the attainment of the allocation desired is impossible without economic behavior, the sufficient condition is satisfied for calling the scheme an application of economics to scheduling.

[11] A pattern of behavior is incentive compatible "if no participant finds it advantageous to depart from his behavior pattern so long as the others do not" [20].

[12] According to Cotton [9], resource allocation and cost recovery represent a dual nature of prices. He further asserted that pricing "satisfies dual objectives" (allocation and cost recovery). It is a misleading assertion, because allocation itself is not usually considered an objective, but only with some qualification such as efficient allocation and equitable allocation. In addition, as Cotton himself cited Nielsen [35], "there is no such thing as 'no allocation.'" An allocation is achieved as long as users gain access to computing resources in some ways. Finally, pricing does not guarantee cost recovery.

attaining only the first function, namely, dispersion of job submission over time. His users needed to interact with their programs; there were hours that were commonly considered more convenient, and therefore, more popular than others for running jobs. Based on the assumption that jobs have various requirements with respect to completion time, differentiation of resources was sought through varying the price of computer usage for each time segment. Two years later, Nielsen [35] presented brief descriptions of two pricing schemes that also aimed at congestion alleviation alone. Hootman [16] discussed the difficulties in determining the appropriate pricing policy, especially those introduced by the time-sharing feature, which became available already in the 1960s. His premise was that computer vendors sold computing services to the users, but the problems examined and their analyses also apply directly to cases in which the computers are owned by the institutions and it is the administrators who set the pricing policies. On a different note, Mendelson [32] conducted a theoretical investigation on the relationship between user-value maximization and other objectives put forth by the administration or the computer vendor (profit maximization and cost recovery). His analysis assumed a lower value for the user of a computer, when a submitted job could not be executed immediately and waited in the queue longer before its execution. In what follows, we discuss these four papers.

3.1.1 Sutherland's Scheme

Sutherland's [46] remains one of the few schemes of computing-resource pricing that have been implemented on a full scale and have attained practicality. We note that neither alternative uses existed for users' budget, nor dual effects of computer use on users' utility were documented; the necessary condition for qualifying as an economic application was violated. The service provider was not directly involved in resource allocation. The futures market and auctions in the scheme did not function like those in actual economies. The conclusion is that the scheme is not an application of economics.

Budget Allocation

Sutherland's scheme was implemented for a single computer at Harvard University. A budget was allocated to each user in accordance with the importance of the user's project, and it was used for bidding on time segments. Each time segment was a quarter of an hour long, whose beginning and end were fixed by the administration. The identity of bidders was disclosed through their initials which they wrote on the bidding sheet as indications of bids. The sheet was accessible to all users of the computer. Bidding for time segments on a certain day was allowed until 9 a.m. on the previous day.[13] A bid was required to be in an integer, denominated by the unit for the amount of budget or that for "currency."

The users' budgets were restored to the full after completion of each job or conclusion of an unsuccessful bidding, and no saving beyond the amount of predetermined full-budget was permissible. The allocated budget was usable only for running jobs on one computer. Moreover, nothing in the paper suggests that the use of computer had both negative and positive effects on users' utility. The scheme did not satisfy the necessary condition to be called an economic application. Sutherland justified his budget policy on the grounds that it prevented the computer from sitting idle. The advantage of limited, versus unlimited, budget lies in its encouragement of users to demand according to the intensity of their needs. Such utility-maximizing behavior increases the total value of the computing resources to

[13]We do not know how long in advance the users were allowed to start bidding.

users compared to that when bids are made without considering the nature of the job. Sutherland's scheme did not make use of this process.

Functionalities of Futures Market and Auction

Bidding for time segments was closed more than 15 hours before the intended time for computer use. In other words, the computing resources were never allocated on the spot and only agreements for future uses were permissible. Thus, naming the resultant trading mechanism, with the lack of a spot market, a futures market is rather misleading; the function of a futures market is to supplement that of a spot market.[14]

Each time segment was given to the highest bidder by soliciting buying prices, and hence, the scheme attempted to make use of a prominent feature of auctions: possible revelation of users' preferences (in this case, those for time segments), which are private information, without direct inquiries. However, owing to the budget policy adopted as described above, the users were not required to compare different uses of their budgets to decide on the bidding amount; the implemented auction was not guaranteed to be one in a proper sense. We conclude that the scheme does not exploit the functionalities of a futures market or an auction.

3.1.2 Nielsen's Examples

Nielsen [35] provided two implementations of pricing, one each for a large and a small community. The details of neither of them is known, preventing us from judging whether they were economic applications or not.

Scheme for a Large Community

In his paper published in 1970, Nielsen gave a brief report on two cases of pricing computing resources. One involved computing services provided to a body of students, faculty members, and researchers at Stanford University, about 5,000 of them, for the purpose of general education and research. It appears that the operation cost had to be covered by the charges for computing service, for which funds in dollars were given to the users by the university. It is not clear whether charges were supposed to cover the installation cost as well. There was a basic rate structure, which was adjusted approximately quarterly and consisted of prices on 18 types of services (terminal rental, leased communication lines, card punching, batch processing service, etc.). The paper implies that the pricing system was functional, but there is no indication as to whether the goal of cost recovery was met. We also do not know how the prices were determined and how superior the implemented pricing was compared to other resource allocation methods.

Scheme for a Small Community

Nielsen's second example concerns a smaller community of about 200 physicists at the Stanford Linear Accelerator Center where most of the jobs were batch-processing. A pricing scheme was planned at the time of reporting, most likely for user-initiated dispersion of job submission over time. The planned base-rate schedule consisted of six rates, including CPU cycles, memory space-time, Input/Output operations, and disk mounting. In addition,

[14] A futures contract is an agreement to buy or sell a certain quantity of a particular commodity, currency, or security at a certain date in the future at a certain price. Futures markets, where futures contracts are made, decreases price uncertainty and enhances trading by risk averse traders who would be less active were there only spot markets.

priority levels, four for batch-processing service and two for printing, were proposed; higher prices for the same computing, if the job was to be executed ahead of the place in queue given by the standard scheduling order. Other components of Nielsen's pricing system, such as price determination, were not given in the paper.

3.1.3 Hootman's Analysis of Pricing

Hootman [16] summarized the pricing situation in the late 1960s as "rang[ing] all over the lot" because of the new computing feature that had become available: time-sharing. Based on his observations, we arrive at some of the inevitable and undesirable characteristics of pricing policy for computing resources.

Complications in Pricing

Hootman identified three commonly adopted objectives of pricing: cost recovery (i.e., pricing "[b]ased upon cost"), profit maximization (i.e., pricing "[b]ased upon [']what the market will bear[']"), and undercutting the competitors' pricing (i.e., pricing "[b]ased upon competitive alternatives"). Competitive pricing is possible only if multiple, would-be service providers exist, which is not the case with the other three studies of the same period examined in this paper. His impression was that *ad hoc* pricing schemes, not based on any of the principles above, were becoming the most prevalent.

Besides the problem of generating sufficient revenue, which is at the root of the objectives mentioned above, Hootman listed three important pricing issues. One is the demarcation between resources the users have control over and those they do not. The demarcation problem can be further divided into: how to measure the overhead (i.e., what the users do not have control over), and how much of the overhead the users should be considered responsible for, in terms of charge. The assumption is that users would not object to paying for resources whose usage they can hardly deny or conceal, but would object if they are asked to pay for resources whose users cannot be easily defined and identified. Another important issue in pricing concerns demurrage, i.e., how to charge for resources which become unavailable simply due to the use of other resources. Finally, he listed understandability of the pricing system by the users as the third of the issues. Among these problems, his analysis focused on: measurement of overhead; and, decision with respect to what portion of overhead should be charged to the users.

Consequences of Overhead and Lack of Information

Having discussed the pricing problem in general, Hootman analyzed the specificities of the problem for major hardware components (e.g., memory, auxiliary memory) and for software (e.g., proprietary programs). There appears to be no easy solution to these problems, not to mention one that could be agreed to by the majority of the computing community. The above three difficulties in pricing is compounded by the fact that little is known about the use of different components of computers and user behavior. We also have the restriction that the amount of resources required for setting the price should not be too large; otherwise, the purpose of pricing resource use is defeated.

These analyses lead to two, rather undesirable, characteristics any pricing policy with a potential to succeed cannot free itself from. First, some arbitrariness is unavoidable, since there exists, most likely, no universally acceptable solution to any of the problems above (e.g., how to measure overhead, how much overhead a user should be held responsible for). Second, a pricing policy, which is easy on the resources and the users, would be based on

easy-to-identify resource usages. Pricing always alters user behavior towards using less of highly priced resources. If the resources, whose uses are more easily measured compared to others, are not the ones which can be considered as representative of the entire computer use, the pricing scheme may skew user behavior in an undesirable and unpredictable way. More seriously, it would not be supported by the users.

3.1.4 Mendelson's Analysis of Pricing Objectives

Mendelson [32] theoretically examined the objectives of pricing, in particular, the relationship between maximization of users' value and the objectives often adopted by the administration of computing facilities or the sole computer vendor (in the case where computing service, and not the computer which provides the service, is sold to the users), i.e., profit maximization and cost recovery. Users had to decide whether to increase the computing load by taking into account disutility from a job's waiting in a queue, which increased along with the waiting time; inseparability scarcity existed. We first summarize Mendelson's results with respect to the service provider's goals of profit maximization, cost recovery, and maximization of users' value: The last one is incompatible with either of the first two. Subsequently, we discuss the reason why we cannot judge whether Mendelson's scheme is an application of economics, as well as practicality of the proposed pricing. The service provider's utility was set equal to the aggregate utility of the users, and the provider's behavior was that of utility maximization. Since the individual users' utility was undefined, we may say that their behavior was that of utility maximization only under certain conditions.

Profit Maximization, Cost Recovery, and Users' Value

One of Mendelson's results is that the value to users would not be maximized if access to computing resources is not charged. In other words, queueing is not self-regulating, therefore, the so-called free access leads to overloading. If usage is priced by a profit-maximizing authority instead (the situation is equivalent to monopoly pricing, in the most common case of one service provider per system), it leads to reduced utilization of the capacity, below the level that maximizes the net value to the users. He showed, in addition, that the investment decisions of a monopolist would result in a capacity lower than that which maximizes users' value, given any utilization rate.

Another reason for adoption of a pricing system is the administration's desire to recover costs through collecting charges. Cost recovery, like profit maximization, does not sit well with users' value maximization. If there are no queueing effects and other externalities (i.e., changes in utility levels, caused by other users' activities), and if there is no change over time in cost per unit-time for each service capacity, the net aggregate user-utility is maximized when revenue is equal to cost, i.e., when budget is balanced, in Mendelson's framework. However, if there are queueing effects, the disutility from or the cost of queueing would not figure into the provider's budget. That is, net user-value maximization, in the presence of disutility from queueing, would not result in a balanced budget. Finally, he claimed that from the point of view of maximizing users' value, seemingly low utilization rates are often optimal, because of disutility caused by queueing. In sum, aggregate users' value is not maximized when profit is maximized, or when cost recovery or high utilization rate is aimed at.

The Type of Users

Naturally, Mendelson's analysis was based on a set of assumptions regarding system behavior. The assumption that deserves special attention is the value of computer services. The first derivative of the aggregate users' (gross) value of computing services was given as follows: $V'(\lambda) = p + v \cdot W$, where λ is the arrival rate of jobs to the system, p is the price per standardized job, v is the delay cost per unit of time per job (or users' willingness to pay for obtaining the processing results one time-unit earlier), and W is the time during which a job is expected to remain in the system. While this value function is meant for the system as a whole from the users' perspective, the motivation behind the function is that of a single user. By defining the aggregate value of computing services as above, Mendelson assumed the existence of a representative user whose value was completely in line with that of the entire body of users.[15] Mendelson did not propose individual users' utilities, hence, we cannot judge whether the existence of a representative user was derived from an admissible formulation of individual preferences and utilities.[16] Since the individuals' utilities and their views of each other are unknown, we do not know whether each users' utility was maximized through the scheme, preventing us from concluding that the scheme satisfied the condition to qualify as an economic application.

Price Determination
The paper suggested determining the price for running a job from the users' value (V) and other variables (v and W) in the above equation, where the value of users' time was to replace V. There are two problems with this approach. One is that the value of users' time does not include the value from running a job, which is a major component in the value to be obtained from using a computer. Another is that the price is set so that the marginal aggregate-utility (i.e., $V'(\lambda) - p - v \cdot W$, the marginal aggregate-gross-value minus the marginal aggregate-cost) is equal to zero. If the price is determined before job submissions, there is no guarantee that the user body would be such that its V' is identical to one that was used to calculate p. If the price is determined after job completion, there would be no such discrepancy, but a critical element for a scheduler's success, i.e., user autonomy [5, 39], would be eroded.

3.2 Markets and Computer Networks: from early 1980s to present

The word "markets" started to take the place of "pricing" in the 1980s, when computer networks became more common. When the users had access to only a single computer, pricing of computing resources meant price-setting by the sole service provider, who wished to alleviate congestion or collect expenses incurred by the computer. In contrast, a computer network consists not only of multiple users, but also of multiple service providers accessible at the same time, appearing much more complex [44][17] and closer to a market as we know from everyday life [3, 13, 18, 25, 33, 48]. The seeming ease with which markets allocate

[15] In other words, the existence of a normative, representative consumer was assumed, which is a justifiable assumption only if certain conditions are met. For detailed discussions on this matter, see, for example, Section 4.D of *Microeconomic Theory* [29].

[16] Accepting the aggregate-value function, the simplest situation, in which such a representative user can exist, is when the users are identical in all attributes (e.g., budget, jobs to run, and willingness to pay for obtaining the results earlier).

[17] According to Stonebraker *et al.*, "[t]he difficulty in scheduling distributed actions in a large system stems from the combinatorially large number of possible choices for each action, expense of global synchronization, and requirement for supporting heterogeneous systems. Complexity is further increased by the presence of a dynamically changing environment, including time varying load levels for each site and the possibility of sites entering and leaving the system."

resources, albeit its complexity [3, 13, 25, 33, 44], and the solid existence of theoretical microeconomics (particularly, the general equilibrium theory), whose results are derived mathematically [13, 25, 33, 44], have captured the imagination of researchers in the field of networks, including distributed computing systems.

3.2.1 Systems of Focus

We can distinguish two types of networks of computers, which may be termed distributed computer systems [42],[18] but are different in terms of application of economics. The distinction is based on the type of services networks provide.

The first type concerns services that are owned and provided by parties other than the users or the administrative organizations that oversee and regulate user activities. The services in this category are mainly data retrieval and transmission, as in commercial applications accessed through a Web browser. The users have no knowledge of, control over, or interest in which computer is involved in the provision of service. The second type deals with services which are defined and owned by the users or the administrative organizations for user activities. The services include not only data retrieval and transmission, but also computation, which is usually the most important component. Computation is architecture sensitive, thus the users are interested in and given control over the choice of computer to be used.

For the first type, there are usually many possibilities as to routing, but the capacity of information transmission is fixed, at least in the short run. The primary purpose of pricing the services for such networks is to control the data traffic so that the value from use of the infrastructure is maximized [26]. Routing configuration is one of the most important attributes in this case [8, 24].

The question of pricing for the second type does not arise exclusively because of heavy traffic along the routes. The second type of network was conceived on the assumption that traffic would remain reasonably light and that sending a job to a geographically remote computer for execution would result in its earlier completion, than submitting a job to a local computer. Since the gains from early, and possibly faster, executions are supposed to outweigh the inevitable increase in the amount of communication, the second kind of network makes the configuration of networks of much smaller importance than for the first type. What markets are expected to do for the second kind is to make the best match between the job to be run and a computer, known as global scheduling. Such networks are used mainly for research purposes, and pursuit of profits is very often considered an objective that runs counter to a productive research environment. The first type, on the other hand, are very often operated by commercial entities, whose survival depends on the profitability of providing services; pricing has an additional role of providing information for investment decisions [26].

Based on the above discussion, we conclude that the two groups of distributed systems aim for different goals through application of economics, and thus, should be considered separately.[19] We focus below on the second type of computer networks, or distributed computing systems [21], which are represented by the Information Power Grid [22], the

[18] Stankovic defined a distributed computer system to be "a collection of processor-memory pairs connected by a communications subnet and logically integrated in varying degrees by a distributed operating system and/or distributed data-base system."

[19] For issues related to the Internet and flow control, see References [8], [10], [23], [24], [26], [31], [38], among others.

National Technology Grid [36], and the EuroGrid [12]. We note that one of the most important problems in distributed computer systems emerges from the fact that jobs, as well as resources available at different times, are not homogeneous.

3.2.2 Why Markets?

A keen interest has been shown in establishing a market of computing resources, which may match jobs and computers in the best way possible. The match arranged by the market is almost always judged desirable, without comparison with other possible matches.

Some argue that we can overcome the difficulty of agreeing on the performance metric for distributed systems by employing markets [13].[20] In addition, markets are claimed to possess many useful characteristics, such as simplicity [13, 18, 25],[21] flexibility [7, 33, 44], efficiency [33] (or the ability to achieve Pareto-optimal allocations under certain conditions [50]), dynamic adjustability [18, 44], scalability [3], sparsity of required communication (which stems from the existence of price) [7, 18, 33, 49], the ability to meet the global objective (when market participants pursue their own local goals) [3, 7, 33], compatibility with object-oriented programming [33, 50], while it has been acknowledged that empirical confirmation of such claims is necessary [49]. Markets' other desirable features are attributed to their decentralized nature. Some consider a decentralized system superior to a centralized system by definition [33], while others see a great possibility of controlling a distributed system using a decentralized method [18, 25, 28, 41, 48, 49, 50]. Decentralized systems are said to be better suited for large systems [13, 33, 45], easy to design and implement [7, 13, 45, 50], scalable [44] (or extensible [41]), devoid of a single point of failure (contrary to centralized systems) [25, 28, 44], speedy [41], and reliable [41]. The representative market models for distributed computing systems are: the Contract Net Protocol [41], the Enterprise [28], a model by Kurose and Simha [25], Agoric System [33], Spawn [18, 48], WALRAS [6, 49, 50], Mariposa [44, 45], and the Grid Architecture for Computational Economy [3].[22] All of these models employ, or are strongly suggestive of using, artificial agents, which acquire computing resources on behalf of the users; they are multi-agent systems. Below we provide discussions on these models.

3.2.3 The Contract Net Protocol

We examine the information exchange process in the Contract Net Protocol [41], which is not too different from that in a non-distributed computing system, if evaluated according to the characterization provided by the author of the protocol. Many of the details necessary for implementation were left unspecified, including agents' utility; we are unable to conclude that the protocol is an economic application.

Bidding and Negotiation

[20] Ferguson *et al.* [13] argued: "Traditional approaches attempt to optimize some system-wide measure of performance (e.g., average response time, throughput). ... The current and future complexity of resource allocation problems described above makes it impossible to define an acceptable system-wide performance metric. ... Most economic models ... The performance criteria of the system as a whole is determined by some combination of the performance criteria of the individual agents."

[21] For some, it is not clear whether "the market approach offers any advantages in overall complexity" [50].

[22] In citing such models, those for database systems that assume light traffic, are often included, e.g., Mariposa.

As summarized by Tilley [47], the Contract Net Protocol "is a conceptual design for a method of task allocation to nodes which can perform the tasks within a distributed computing system." It is a model with artificial agents in which a manager node (a type of artificial agent) broadcasts the task to be carried out and contractor nodes (another type of artificial agents) bid for the task. The manager node evaluates the bids and decides which contractor is to be awarded the task. Such exchanges of information were together termed a negotiation process, which is an unfortunate name choice. If it were a negotiation, a task request should be modified upon rejection by a contractor, in line with the feedback given by that contractor, and resubmitted to the same contractor. Instead, the request is sent to as many contractors as possible, and the sender waits for a bidder who considers the request acceptable. The protocol does not concern negotiation.

Characteristics of the Information-Exchange Process

Smith [41] identified four important components in his information exchange process: (i) lack of centralized control; (ii) two-way nature; (iii) evaluation of information by local entities; and, (iv) finalization by mutual agreement. These features are not unique to Smith's scheme, and most of them are also found in scheduling mechanisms for systems with a single computer. When various time segments are priced differently, communication takes place; the preference of the service provider, with respect to the timing of users' job submission, is conveyed to the users. The users evaluate the information (i.e., prices) in order to decide when to submit their jobs, and by submitting a job for particular time segments they communicate their preferences under that set of prices. Hence, although it may not be as explicit as the one in the protocol, pricing always entails two-way communication and evaluation of information by local entities. Moreover, as it is the users who decide which time segments to demand, given the differentiation of them by the service provider, the allocation of time segments is determined through mutual consent. Therefore, the listed components, with the exception of the first one, are features also shared by a pricing scheme for a system with a single service provider. Note that it is not possible to bypass a computer when there is only one in the system. The supporting organization of such a computer is usually considered the central authority, and to the extent that many decisions involve the caretaker of the sole computer, central control is unavoidable for a system with one computer. Thus, it is quite natural that we do not find Smith's first feature in non-distributed systems.

The Goal of the Protocol

The formats of a task and a bid, as well as their evaluation method, were not specified in the protocol. The resultant matches will vary, depending on their specifications, which are in turn dependent on the objectives of the manager and the contract nodes. How accurate the task description and the bid contents would be, or how much private information would be revealed through communication, is also dependent on the specifications. Thus, it is not clear which goals for distributed systems could be supported by the protocol, although some match would probably be achieved. Additionally, the utility of nodes was not taken into consideration in drawing the scheme, the presence of which is required if we are to confirm that the problem posed to the system is an economic one. We cannot conclude, without further details of implementation, whether the protocol simply borrows terminology from economics (e.g., bidding), and has no further bearing on economics, or it is driven by some economic force.

The Contract Net Protocol is a blueprint for matching mechanisms between jobs to be executed and hardware for their execution. It has shown researchers the possible applications of economics in global scheduling. The novelty lies in that fact; it has directed our attention to a possible use of economics in distributed systems. We need to conduct more research on how various approaches to the scheduling problem, including the protocol, relate to each other [47], before we can conclude that economics is useful in scheduling.

3.2.4 Enterprise

The Enterprise system [28] is a fleshed-out version of the Contract Net Protocol, which connected personal workstations, using a local area network. Although favorable simulation results were reported, we cannot attribute them wholly to the scheme's general features. Neither can we conclude that the scheme is an application of economics, because the utility of the service providers in the scheme was undefined and there was mention of neither dual effects of resource use nor user budget.

Member of the Contract Net Protocol Family

A request by a client (or a manager, in Smith's [41] terminology) for bids contained the numerical priority of the task, special requirements, and information of the task that allowed processing-time estimation.[23] A response by a potential contractor to a request, i.e., a "bid," contained either an estimation of completion time or an acknowledgement message (if it happened to be executing a job at the time of receiving the request). The evaluation criterion was how soon the completion was expected to be; the contractor with the shortest processing-time won the task. That is, the clients' utility was the negative value of expected job-completion time, provided that they maximized utility. For the reported simulation, minimization of the mean flow-time of jobs was chosen as the scheduling objective. Thus, the shortest-processing-time-first scheduling was considered optimal among the available heuristic methods. We note that the scheduling objective has no logical connection with the objective of the contractors (i.e., the service providers), unlike in a non-distributed system where the sole service provider is the scheduler.

In order to prevent the clients from reporting underestimated processing times for obtaining earlier spots for their jobs, tasks were aborted if they exceeded the time specified by the "estimation error tolerance" parameter.[24] A client waited for a certain time after sending a request, and before engaging in an evaluation.[25] If no bid was received by the time of evaluation, the first bidder won the task. A later bid was considered, if it was "significantly better" than that of the tentative winner. These measures were put into place in consideration of unpredicted delays and losses in message exchange. Cancel messages were sent to all bidders who did not win the contract.

A simulation was carried out on ten various configurations, each of which had exactly

[23] As one of the implementation examples, Malone *et al.* [28] reported a task description which contained the following information: the estimated processing time on a "standard" processor, and the names and lengths of the files to be loaded before processing. The estimated processing time was expected to be provided by the user. If not, the default value was employed. The time for file loading was estimated to be proportional to the length of each file.

[24] This measure does not enforce honest reporting of estimates. Rather, it encourages reportings of honest estimates minus the permitted error.

[25] The client's own bid for task execution was not processed for a certain time period so as to give time for bids from other workstations to arrive.

eight units of processing power.[26] Jobs were assumed to be independent of one another, processable on any workstation in the network, and 1,200-75,000 of them arrived according to a Poisson process. Their size was assumed to be exponentially distributed. The examined settings differed with respect to the accuracy of job processing-time estimates, delay in message transmission, the system utilization level, etc.[27] The same sequence of random numbers was used for job generation for each simulation.

Market-Like Task Scheduler

As the title of the paper admits ("Enterprise: A Market-like Task Scheduler for Distributed Computing Systems" [28]), the Enterprise scheduler does not quite involve a market. What was christened as a bidding process involved revelation of local information, or information privately held by the job-executing machines, but there was no reward to the machines from winning the bids; since the service providers' utility was not part of the framework, there was no increase in the utility of the machines, or of the owners of the machines, when a bid was won. No incentive existed for the machines to report the best possible completion time and obtain a contract. Bidding was a form of information exchange, which was not motivated by gains in trade of scarce resources. In short, we cannot conclude that the machines behaved in a utility-maximizing way and that the scheme concerned the economic problem.

Load, Time Estimates, Number of Machines, and Message Delay

We now examine the conclusions from the simulation results. The positive relationship between the system load and the mean flow-time is as expected. A heavier load means that there are fewer machines available at a time and over time, forcing each job to stay longer in the system.

The second conclusion is that the inaccuracy in processing-time estimates has minimal effects on the mean flow-time. Contrary to the implication in the paper, the features unique to the Enterprise were probably not responsible for reducing the effects. If there had been only one computer to execute the jobs, the estimates would have been of crucial importance in the shortest-processing-time-first scheduling. However, the distributed system executes as many jobs as there are workstations in the network at the same time, rendering the priority of smaller importance. If there are n workstations in the network, it is desirable that every set of approximately n jobs is prioritized according to accurate estimates, but how jobs are prioritized within each set is of much smaller importance than it is under a non-distributed system. This holds for any distributed computing system, whose computers are capable of executing any job that arrives.

The third conclusion is that the mean flow-time is not reduced when more than eight to ten machines are added to the network, under the assumption of perfect communication. The reason for hitting the ceiling at eight machines per network was not given.

It was also concluded that message delays have little effect on the mean flow-time, where each delay was introduced as a fixed percentage of the average task-processing-time. All messages were delayed for the same amount of time, hence, the time required for communication was monotonically transformed, resulting in the same order of message arrivals. Thus, the mean flow-time could be delayed only by the extent of the tardiness

[26] One unit of processing power was that of a Xerox 1100 processor.

[27] The rate of system utilization was defined as the expected amount of processing requested per time-interval divided by the total amount of processing power in the system.

of messages, and not because of less-than-optimal scheduling that would be caused by more general kinds of message delays. In sum, the desirable results observed seem to be attributable to the features of the Enterprise that are not related to the economic problem.

3.2.5 Model by Kurose and Simha

We briefly discuss the general equilibrium theory, which the model by Kurose and Simha [25] draws on, with attention to the so-called tâtonnement process and the problematic aspects of the process (i.e., the presence of an auctioneer and incentive incompatibility). While inseparability scarcity existed for the artificial agents representing nodes in the model, they were not utility maximizers, disqualifying the model as an application of economics.

General Equilibrium Theory and Resource-Directed Approach

Kurose and Simha implemented one of what they call the two basic microeconomic approaches, the price-oriented and the resource-oriented approaches, which are better known as a tâtonnement process (for reaching an equilibrium) with pure price adjustment and that with pure quantity adjustment, respectively.[28] Hence, the model by Kurose and Simha applied the general equilibrium theory in economics, a theory that concerns price and quantity determination in equilibrium initiated by Léon Walras in the late 19th century. The economy under consideration in the theory is one in which the number of agents in the system is large enough so that any one of them cannot affect prices by acting alone, and agents do not collaborate. In other words, all participants in the economy, producers and consumers, are price-takers. Such an economy is often interpreted either as a market economy that is perfectly competitive or as a planned economy. The model by Kurose and Simha is based on Heal's [15], which adopted the latter interpretation.

In an exposition of the standard tâtonnement process with pure quantity adjustment, an economy with production is usually considered [15, 29]. There is an auctioneer (or a resource allocator) who informs profit-maximizing producers of their entitled quantities of inputs, and the producers report back the marginal productivities (i.e., the first derivatives of the production functions) at those quantities. Upon receipt of the information from the producers, the auctioneer changes the allocation so that more inputs are allocated to the producers with higher marginal productivities. The process is repeated until an equilibrium in price and quantity is reached.[29] In the context of distributed computer systems, there are no producers, and we need to alter the scenario as follows: The auctioneer informs the utility-maximizing user-agents of their entitled quantities of resources and the agents report back the marginal utilities (i.e., the first derivatives of the utility functions) at those quantities. The auctioneer changes the allocation of resources so that more inputs are allocated to the agents with higher marginal utilities. The process continues until an equilibrium in price and quantity is attained.[30] This is the approach preferred and adopted by the au-

[28] In connection with the two approaches, some information sources in economics were given [2, 15, 20]. However, the two approaches do not appear in the citations by the names given by the authors. Hurwicz's [20] tentative naming of the processes, "price-guided" and "quantity-guided," appear to have been adopted with a slight change.

[29] Since the producers are profit-maximizing, the ratios of marginal productivities (or the marginal rates of transformation) are equivalent to shadow prices (or the prices at which the profit-maximizing producers are willing to accept the proposed allocation of resources) in an economy with production.

[30] Since the agents are utility-maximizing, the ratios of marginal utilities (or the marginal rates of substitution) are equivalent to shadow prices (or the prices at which utility-maximizing agents are willing to accept the proposed resource allocation) in a pure exchange economy.

thors, because all interim allocations are feasible, unlike the price-oriented approach.[31,32] We note that the process does not guarantee convergence to a unique equilibrium, even if one exists, without further restrictions on the economy.

Another attractive feature of the resource-directed approach was reported: "When analytic formulas are used to compute performance, successive iterations of the algorithm result in resource allocations of strictly increasing systemwide utility."[33] Putting aside the issue of incentive compatibility, we may say that optimization by local agents led to an optimal solution for the entire system because the global utility was set equal to the sum of utilities of local agents. The global objective function was a function only of local objective functions increasing in all of its arguments, and unresponsive to the names of the local agents; local optimization coincided with global optimization. The iterations in resource allocation are solutions to the economic problem only if successive allocations do not lead to lower utilities for all agents concerned. If any of the agent's utility is to be reduced through another round of transaction, that agent is better off by not participating in the exchange. Thus, the feature is equivalent to economic feasibility of each iteration in the adopted framework, under the assumption that honest reporting of utility levels is in the interest of local agents.

The two advantages described above, feasibility (in a discrete process) and monotonicity, were labeled two desirable properties of tâtonnement [27]/gradient [20]-based processes for reaching an equilibrium, by Malinvaud. He was concerned about the possibility of slow or "disorganized" convergence to an equilibrium, which implied that the existence of an equilibrium and convergence to one through a tâtonnement process, by themselves, do not guarantee practicality of the process as one in economic planning. Kurose and Simha proposed algorithms which were based on a tâtonnement process with pure quantity adjustment. Theoretical investigation of their most basic algorithm by Heal [15] had shown that it indeed exhibits both properties. For the process to function, however, one condition has to be met: the central authority's knowledge of an initial allocation that is feasible, which is the cost of obtaining the desirable properties (i.e., feasibility and monotonicity) according to Hurwicz [20].

Optimal File Allocation

The distributed system chosen for investigation was a network of nodes, which were assumed capable of communicating with any other in the network. The problem was how to allocate files optimally.[34] We could view each node as an artificial agent of the system. The cost

[31] See Section 3.2.7 for an application of the tâtonnement process with pure price adjustment.

[32] The title of Heal's paper [15], on which the model is based, is somewhat misleading: "Planning without Prices." What Heal meant by "without prices" is that in a planned economy, where all resource-allocation decisions are made by the central planner, the planner works directly with marginal productivities (in a production economy, or marginal utilities in a pure exchange economy) reported by local agents. Heal implicitly assumed producers to be profit-maximizing, in which case the ratios of marginal productivities (or the marginal rates of transformation) are equivalent to shadow prices (or the prices at which the profit-maximizing producers are willing to accept the proposed allocation of resources).

[33] There is also a disadvantage to the approach, vis-à-vis the price-oriented approach. Heal [15] concluded that in the tâtonnement process with pure quantity adjustment (or the resource-oriented approach, according to Kurose and Simha) more information exchange would be required than in the process with pure price adjustment. Hurwicz [20] pointed out that the total information would be of higher dimension in the process with pure quantity adjustment (or the "quantity-guided" mechanism, according to Hurwicz) compared to the process with pure price adjustment (or the "price-guided" mechanism), but also added that whether the difference was significant was "somewhat controversial."

[34] Each node had a local look-up table, which provided information on the file fragment locations so that

of communication to each node was defined to be the average delay in the transmission of messages. The cost of access delay was defined to be the expected time in access delay. In turn, the sum of the cost of communication and the cost of access delay was called the expected cost of access to the file source at a node. Therefore, allocating all files at one node may reduce the cost of communication, but only by increasing the cost of access delay, because that node must handle all inquiries in the system; inseparability scarcity existed. The expected cost of access to the entire network was the sum of the expected cost at each node. The optimal file allocation was taken to be the allocation that minimizes the expected cost of access for the whole network. In order to cast the problem as a utility maximization problem, the utility was set equal to the negative of the expected cost.

The performance of three algorithms for file allocation was examined, using 19 nodes and allowing them to generate access to file resources at a rate determined by a Poisson process (with its parameter equal to unity, the inverse of the number of nodes, etc.). They all employed gradient processes; each node computed the first derivative of the utility function and/or the second derivative, evaluated at a specific point, and sent that information to the central node (or alternatively, to all other nodes). Due to such employment of nodes, the algorithms were characterized as distributed (or equivalently, decentralized).[35] The termination criterion was that the marginal utilities of all nodes in the network be sufficiently close. When the criterion was not met, the files were reallocated so that the difference in marginal utilities would be smaller. The paper concluded that all algorithms, which were discrete-time processes, had the following desirable properties: feasibility in all iterations, strict monotonicity, and fast convergence.

Problems in General Equilibrium Theory

There are some problems in the general equilibrium theory, in connection with tâtonnement processes. One is that it is devoid of a price formation mechanism under the market-economy interpretation [1]. Prices are adjusted by an auctioneer until an equilibrium is reached, but there is no real-world counterpart to such an auctioneer in markets. If we consider a planned economy, the role of the auctioneer can be thought to be presumed by the central planner. The second approach is the one taken by Heal's model [15], on which the model by Kurose and Simha is based. The second interpretation is not free of problems if the system is to be called a decentralized one, as Heal [15] and Malinvaud [27], among others, did. The presence of a central planner makes it difficult to claim that the model is that of a decentralized system, which functions without central directives, as is assumed by many who are engaged in applying market mechanisms to the allocation of computing resources. Heal's model involved reporting of preferences, but no local decision-making with respect to available choices. In other words, it was decentralized only in the sense that information was collected from the local agents. The same holds for Malinvaud's decentralized process. As far as planned economies are concerned, it may be legitimate to name such an economy a decentralized and planned one, as opposed to a planned economy in which all economic activities are determined by the central planner without any feedback from the local agents. The mechanisms employed by Kurose and Simha for adjustment of the system towards an equilibrium are informationally decentralized [20], but that feature is not equivalent to decentralized decision-making.[36] Indeed, the processes do not concern

a request not met locally could be sent to an appropriate node.

[35] The two terms, distributed and decentralized, were used interchangeably by the authors.

[36] An informationally decentralized process is one which has informational requirements that are no greater than those for a perfectly competitive process [19].

local decision-making, just as Heal's model does not. In sum, a decentralized, planned economy does not truly qualify as a decentralized system with advantages such as lack of a single point of failure, as envisaged by many of the researchers in the field of global scheduling.

Another problem in the general equilibrium theory is also carried over to the model by Heal, as well as to that by Kurose and Simha: incentive incompatibility. Although reporting the derivatives of the pertinent functions to the central authority is a standard element in the tâtonnement processes, its incentive compatibility has been established only when the number of traders is infinite for a pure exchange economy, if no forced, initial redistribution of endowments is allowed [20]. Note that individual nodes could have increased the final utility by falsely reporting the levels of marginal utility that were above the actual levels. No mechanism was in place to prevent the nodes from resorting to such an action. However, the local agents in the investigated network acted so as to fulfill the global goal, i.e., minimization of expected access cost to the entire system, by forgoing the opportunity to increase their own utilities. That is, a distributed algorithm is not synonymous with distributed decision-making or a decentralized system, which implies utility maximization by economic agents.

The above discussion also serves as an analysis of whether the resource allocation scheme was driven by economic considerations of the agents in the system. The model relied on nodes' balancing the benefits and the costs of owning a file fragment; inseparability scarcity was present. The fact that the nodes reported their marginal utilities honestly to the central node (or, to all other nodes), in face of feasible cheating and attainment of higher utility, together with the fact that honesty did not factor into the utility defined, indicate that the nodes were not utility-maximizing agents; the first part of the necessary condition to be an economic application was not met.[37] The study neither validates nor invalidates the appropriateness of creating markets for computing resources in distributed systems.

3.2.6 Spawn

Spawn [18, 48] is a resource allocation mechanism for a network of heterogeneous workstations whose agents are sellers (i.e., owners of workstations, who are not using them at any given moment) and buyers of CPU time. Human users do not directly participate, making the system a multi-agent one. The special feature of Spawn is its spawning process or dividing a task into subtasks. Our examination of the roles of funding and pricing indicates that they are unlikely to have supported the agents' efforts to solve the economic problem. The utilities of both buyers and sellers of computing resources were not defined; it is impossible to confirm that the scheme satisfied the necessary condition to be an economic application.

Spawning Concurrent and Independent Tasks

In the reported implementation [48], a buyer bid for an idle machine, which was devoted to a single winning buyer. A sealed-bid, second-price auction was employed,[38] where a

[37] The utility of the auctioneer, an active participant in the resource allocation process, was left undefined. We may say that the algorithms' termination criteria are the goals of the auctioneers. Then, each set of adjustment rules must be such that the utility of the auctioneer increases after each adjustment and reaches its maximum as the termination criterion is satisfied.

[38] A sealed-bid, second-price auction was chosen so that repeated communication, such as that in an English auction, would not be required. Huberman and Hogg [18] further added: "More importantly, this mechanism avoids gaming of the system by the agents—for example, by strategically driving up the winning price so that alternative users exhaust their resources more quickly ...". For this assessment to hold true,

bid contained the length of time desired, the quantity of funds (which is equivalent to the bidding price, as explained below), and a brief task description [48]. The budget for each task was determined by its relative priority [18]. In the most general case of dividing a task, its divisibility must be checked. If divisible, information on how it can be divided to make the resultant subtasks suitable for execution by separate machines, what kind of resources each divided task requires, etc., should be communicated to the system for a successful spawning. These problems were avoided by tailoring the system to handle concurrent and independent subtasks. When subtasks were spawned, the budget for the original problem was transferred to the subtasks at a constant rate, and the budget unused was kept by the subtasks [18]. The simulation examples concerned asynchronous Monte Carlo applications, which were executed concurrently in a network of idle workstations. The number of nodes employed varied from six to 64, and that of trials per second ranged up to 100,000. The overhead was observed to be relatively small.

Bids and Funds as Dynamically Determined Priority

We first examine below how auctions in Spawn functioned. It was optimal for the subtasks to place all funds accumulated as a bid in any auction [18], because there was no other use for funds and because auctions in general do not penalize the losers financially.[39] Therefore, any winning bid was equal to the entire funds owned by the winning subtask at the time of bidding. Indeed, Waldspurger *et al.* [48] reported that the resources were allocated in a way that reflected the funding ratio in all runs, which is a fair allocation, according to the authors. Subsequently, fair allocations were not due to the auction mechanism by itself, but more due to the applicability of funds that was limited to one task.

By allowing unused budgets to accumulate and by setting up the game so that the bidder with the biggest budget would win, the funding did not simply reflect the original relative priority, but one that was dynamically adjusted [48]. This ensured that no unexecuted job is left with a small probability of winning resources when all others are completed [18]. Thus, the role of auctions in Spawn was to communicate the dynamic priority of jobs, which was completely determined by the remaining budget, to resource sellers.

The Role of Price

We now turn to the role of price in Spawn. In the description of the spawning procedure, there is an indication that subtasks were spawned to machines that were close to the originating machine of the task, and not to all machines in the network [48]. It is not clear, however, which course of action was taken upon losing in an auction: whether the subtask waited at the same machine for its fund to accumulate, or ventured further in the network for a machine whose service could be bought with the funds available at that time. Hence, we do not know the mechanism which "permits concurrent Spawn applications to adaptively expand into more machines when prices are low, and forces them to contract into fewer machines when prices are high" [48].

Recall that bids were synonymous with budgets, which were in turn equivalent to pri-

the expected winning bid in the sealed-bid, second-price auction must be lower than that in the English auction for the same item. That is true under restricted conditions. For detailed discussions on this issue, see, for example, "Auctions and Bidding" [30].

[39] Huberman and Hogg [18] described the bid employed by Spawn as follows: "The amount of the bid was based on two pieces of information: the intrinsic speed of the machine to complete that task and how much money the agent had." It was not described how the information on speed could be communicated or in which way it influenced the bid.

orities. Therefore, the winning price depended on the levels of priority given to jobs that asked for the use of a particular idle machine. Moreover, the communication mechanism of Spawn was such that jobs of which priority would be coveting the same idle machine could be revealed only through an auction. That is, there was no knowing beforehand which machine would be offering services cheaply. As a consequence, the situation in which "rich agents were all found to be bidding on very few machines while others were idle" in a sparsely connected network [18, 48] could not be avoided; some of the fundamental structures of the system have to be changed, if such situations were not to be encountered [18].

As was analyzed above, all bids were equivalent to dynamically adjusted priority of jobs or all the funds at hand. Thus, the price at which the resource in question was traded, was always equal to the amount of funds owned by the job with the second highest priority. In other words, price at any time is nothing more than a reflection of two machines that were close to an idle machine and had the highest priorities. Our interpretation of price in Spawn conforms with one of the experiments reported [48]. Instead of funding the spawned jobs equally, the ones which were "running on the cheapest few machines were given the funding." This amounted to increasing the priority levels of jobs (and hence, the amount of bidding) if the previous winning bid had been low. Considering the funding and the auction structures of Spawn, higher rates of fund transfer make a higher winning-bid, and hence, a higher winning-price more likely. This analysis matches with their result of the experiment: "This strategy eliminated the price difference[.]" In other words we are not led to the conclusion that price acted as a guidance for better resource usage.

The Global Objective and the Economic Problem

The global objective of allocating machines to jobs in a fair way, which, in the framework, was to allocate them to jobs with higher priorities [48], can be always met if there are more idle machines than there are subtasks. If the number of idle machines is smaller than that of subtasks, whether Spawn would grant resources to the jobs with the highest priorities depends on how closely located the jobs with highest priorities are in the network. When the highest priority jobs are close to each other, some jobs, which should not be granted resources before other subtasks from the point of global ordering of job execution because of their low priority, may be given resources [48]. Differently put, the attainment of the overall objective has more to do with funding of various subtasks and how the originating machines of the subtasks are located relative to each other than with each subtask's economic problem.

Since funding is but for one subtask, there was no incentive for the bidders in Spawn to search for cheaply priced resources so that the remaining budget could be used for some other purposes. Neither alternative use of budget nor inseparability scarcity was found in the setting. We cannot say that the scheme is an application of economics. It is not known whether Spawn was better than other methods in allocating resources in a fair manner, as defined by the authors.

3.2.7 WALRAS

WALRAS, like the model by Kurose and Simha, is an application of the general equilibrium theory to management of distributed systems [49]. It implemented the tâtonnement process with pure price adjustment, through which an equilibrium in the economy is reached, while

Kurose and Simha opted for that with pure quantity adjustment.[40] We provide a summary of the tâtonnement process in WALRAS, which is different from the standard process, and the implications of the functional form of the users' utility, in terms of the existence of an equilibrium, its uniqueness, and convergence to the equilibrium. There were several goods that affected the user utility (satisfying the existence-of-alternatives condition), but WALRAS is incentive incompatible, as any model based on the general equilibrium theory with a finite number of agents would be (if there exists neither production nor the possibility of redistributing initial endowments); it is unsuitable to be called an economic application. User budget was undefined and resource use in WALRAS did not have dual effects; neither type of two scarcities seems to have existed. Finally, we do not know how well the agents, whose utilities are of the same functional form, may represent a heterogeneous body of users in reality.

General Equilibrium Theory and WALRAS

In the standard tâtonnement process with pure price adjustment, there is an auctioneer who informs agents of the prices, and the agents report back the amounts of goods they demand (or more precisely, demand for goods over and above the amount endowed) at those prices. Such reports are called bids. The auctioneer calculates the new prices, according to the predetermined rules and the amounts of demand reported, and the process repeats until the prices no longer need to be adjusted.

WALRAS differed from the standard tâtonnement process in that demand functions were reported by the agents (instead of a point on a demand function) and that the auctioneer dealt with each good separately [6, 49]. Moreover, not all agents reported their demands for all goods in each time period [6]. Random draws, which were independent across time and agents, determined which bids were submitted. For the unselected combinations of agents and goods, the bids from the previous period were used. The advantage of their asynchronous bidding was small price oscillations [6, 50]. Again, the agents which participated in the biddings were not human users, and hence, the scheme was based on artificial agents. Cheng and Wellman [6] favored their approach over other processes for attaining an equilibrium in the general equilibrium theory, since they saw fewer opportunities for strategic interactions and no trade took place until an equilibrium was achieved (no resource was allocated based on intermediate results, which were by definition not global optima and may have been irreversible if implemented). Equilibria reached through tâtonnement processes in a framework as the one adopted by WALRAS are Pareto-optimal, which are often interpreted as desirable allocations [50].

The utility function of the resource users, $u(x)$, was of constant elasticity of substitution: $u(x) = \left(\sum_{j=1}^{k} \alpha_j (x^j)^\rho\right)^{1/\rho}$, where α_j's were randomly generated coefficients from a uniform distribution, x^j was the amount of good j, and ρ was fixed at 0.5 (for the main simulation). Therefore, the resulting excess-demand functions, for each agent and for the entire economy, had the property of gross substitutability [6], i.e., there were no strong complementarities among the goods [29]. The existence of an equilibrium was guaranteed by the preferences implied from the utility functions (which were continuous, strictly convex, and locally nonsatiated) and non-negative total excess-demand (as Cheng and Wellman implicitly demonstrated with experiments).[41] Moreover, gross substitutability ensured the uniqueness of equilibrium and convergence to that equilibrium point on any price path.

[40] See Section 3.2.5 for an application of the tâtonnement process with pure quantity adjustment.

[41] See Figure 1 in "The WALRAS Algorithm" [6].

While the adaptive learning behavior of the auctioneers justified the rules of WALRAS, users remained simple price-takers who reported their excess-demand functions honestly. The users would not have reported the true demand functions if they acted so as to maximize utilities, as is evidenced by the utility function shown above (which is insensitive to the act of truth telling). This problem was also found in the model by Kurose and Simha.[42]

Convergence and Other Problems

While WALRAS has been used for several distributed multicommodity-flow problems [49, 50], we concentrate here on the most comprehensive results given in "The WALRAS Algorithm" [6]. An examination of 100 randomly generated economies, with five or seven agents of utility as described above (where j is equal to 5), showed that the median behavior of the system was a rapid convergence to the equilibrium at the beginning, and leveling off at a small, positive amount of total excess-demand after 150 iterations. When values other than 0.5 for the substitution coefficient were adopted, convergence was not seen even after 5,000 iterations in some cases.

For the feasibility of the proposed algorithm, it is imperative that it promises convergence to an equilibrium. Singularity of equilibrium is convenient, since it spares us from the need to compare equilibria and further guide the system to the most desirable one. These favorable properties are obtained only under restricted circumstances [6]. Gross substitutability of the aggregate excess-demand function is a sufficient condition, and the authors reported that they could not find a class of utility functions that are not grossly substitutable and yet converge to an equilibrium.

Whatever the necessary conditions may be for convergence, and possibly for uniqueness of equilibrium, the utility functions employed must represent the preferences of users. The framework adopted utility functions with constant elasticity of substitution among multiple goods, but we do not know whether such utility functions reflect the preferences of users in distributed computing systems. WALRAS explored the case where every agent had a utility function of the same type. How it would serve a system of users with various forms of utility functions and incentive compatible strategies, and how it would compare with other scheduling mechanisms, are yet to be seen.

3.2.8 Mariposa

Mariposa is a distributed database and storage system for non-uniform, multi-administrator, wide area networks [44, 45]. We argue below that some of the advantages of markets, which are claimed also as those of Mariposa, do not hold unconditionally, and question the practicality of the objectives chosen for the artificial agents. Some, but not all, of the agents' objectives (and thus, utilities of some agents) were proposed. We could not conclude that the scheme satisfied the necessary condition for being an economic application.

Unconventional Features of Mariposa

The work was motivated by the observation that non-uniform, multi-administrator, wide area networks require the following features [44, 45], which traditional distributed database management systems, according to the authors, cannot inherently possess [45]: scalability, data mobility, lack of global-synchronization requirement, total local autonomy, and

[42] No process that leads to Pareto-optimal equilibria is incentive compatible in a pure exchange economy with a finite number of agents, unless the endowments of the agents may be forcibly exchanged before the process starts [20].

configurability of policies. The adoption of economics was considered desirable because it allowed a large number of sites in the network and locally made decisions (which permitted data mobility and made global synchronization unnecessary). Policy changes were to be made easy with Rush, the language used by Mariposa [45]. Finally, scheduling complexity was to be reduced by applying economics, where an "invisible hand" would make trading of resources "reasonably equitable" [44, 45].

For the "invisible hand" property to be present, there must be enough agents in the system so that each agent correctly sees individual action inconsequential to the condition of the system as a whole. Economic behavior depends on how each agent sees the rest of the system, and therefore, it differs according to the number of the agents in the system. The behavior of the system when there are only two agents would be quite different from that when there are 100 agents. Moreover, a market exhibits the property only under certain assumptions, and the resultant equilibrium would be Pareto-optimal, which is not necessarily equitable.[43]

Goals of Clients, Brokers, and Bidders/Servers

There were three types of entities in Mariposa: clients, brokers, and bidders/servers [44, 45], all of which were artificial agents. We assert below that the necessary condition for qualifying as an economic application cannot be concluded to have been satisfied and that the proposed objective for the broker is unlikely to be practical.

A broker sent subqueries to bidders and chose the best bid on behalf of the client with a query to be performed. Each query had its own budget, and no transfer of budget among queries seems to have been permissible. Neither use for unused budget nor dual effects of resource use was documented. A possible scenario was outlined for data query, in which the broker aimed to maximize the difference between the budget and the cost. Since the brokers worked for the clients, there were two possibilities with respect to their interests: The interests of the brokers and the clients were perfectly in line with each other, or they were not. In case the objectives of the broker and the client coincided, and indeed if each query has its own budget which cannot be shared with other queries, the broker should have used all the budget that could be used for that job to obtain the fastest service possible; minimization of query completion time is a more compelling objective for the broker as an agent for a client than the proposed one. The objectives of the two would not be the same, if, for example, there is some use for unused budget for the brokers, but not for the clients. The brokers would want to maximize the unused budget as proposed in the paper, but that would not have any meaning to the client. The client would rather have the broker choose the bid that would make the fastest query. In this case, the scheme described in the paper would not be supported by the clients; there probably would be a great demand for change of the system.

As for bidders, rough sketches of bidding schemes for data query and storage were provided in the papers [44, 45]. However, the utility functions for the bidders/servers are unknown, and hence, it cannot be concluded whether the bidding strategies were utility maximizing. We do not know if the necessary condition to be an economic application was met.

Contribution of Economics

Experiments regarding the behavior of Mariposa were conducted, with three sites and

[43]For detailed discussions on this topic, see, for example, Section 16.C of *Microeconomic Theory* [29].

databases of the size 96-128 MB for the wide area network case and those of the size 64-160 MB for the local area network case [45]. Data movements for the identical query using Mariposa and a traditional query optimizer were analyzed, although execution times were not compared. The conclusion of the analysis is that Mariposa is superior because it gave more choices, in the form of bids, than the traditional optimizer [45]. In order to establish that the formulation of a data query problem as an economic one is what makes Mariposa desirable, we need to confirm that the best outcome was attained (among the possibilities which are more numerous than in other query systems), thanks to the economic behavior of the agents involved. Such confirmation is impossible, as we do not have enough information to affirm the existence of the economic problem in Mariposa.

3.2.9 Agoric Systems

An agoric system is an intellectual exploration as to what economics may be able to do for distributed computing systems. Without more information than has been provided in the paper [33], we cannot determine whether the system is an application of economics.

Decentralization: Computing Systems and Markets

The paper by Miller and Drexler on agoric systems [33] draws our attention to the evolution of models in computer science: from centralized to decentralized. They argued that while economics and any system with goals, resources, etc., have much in common, the evolution has made the market mechanism, in particular, relevant to computer science. They advanced markets as a mechanism allowing an effective attainment of desirable global goals through locally made decisions, based on some results in the general equilibrium theory.[44] In addition, a market, being a decentralized system, was described as potentially more rational than a centralized system "since it involves more minds taking into account more total information." The authors' definition of an agoric system is: a software system that is intended to take advantage of such workings.

Prices as Seen by Hayek

The paper summarized Hayek's view on price as follows: "This simple, local decision rule gains its power from the ability of market prices to summarize global information about relative values." Although the role of price, as an embodiment of all the relevant information in the economy when making economic decisions, has been an important component in the general equilibrium theory, such prices are available only in an equilibrium [1]. As discussed in connection with the model by Kurose and Simha, how such prices could be formed has not been part of the theory [1], and there is no auctioneer in markets in reality who gathers and disseminates information and adjusts prices and/or quantities. In the domain of auction theory (one of the few theories that is concerned with the process of price determination), McAfee and McMillan [30] asked the following question: "Is it correct, as Hayek asserted, that the price summarizes all of the relevant information about supply and demand?" They answered in the negative for non-competitive environments, which are more frequently encountered in real economies than perfectly competitive ones.[45] Moreover, it has been questioned whether private information, as Hayek saw, is appropriate as that incorporated

[44] For discussions on what properties the equilibria of an abstract economy may have and on under which conditions they are brought about, see, for example, Chapters 16 and 17 of *Microeconomic Theory* [29].

[45] "If the competition is less than perfect it is in the seller's interest, if possible, to adjust the price after the sale in the light of any new information he obtains about the item's value to the buyer ..." [30].

in the price, for the purpose of supporting the general equilibrium theory [51]. The powerful role of prices and the possibility of their formation need to be shown through simulation and implementation of the system.

3.2.10 Grid Architecture for Computational Economy

The Grid Architecture for Computational Economy (GRACE) aims at incorporating an economic model into a grid system with existing middleware, such as Globus and Legion, on the grounds that the economic viewpoint is better suited than other mechanisms to run a distributed computing system [3]. Below we discuss the effect of proposed objectives of service providers on pricing as well as simulation results. Although utilities for service providers and users were suggested, there was an implication that neither dual effects of resource use nor alternative uses for user budget existed; the scheme does not qualify as an economic application.

Family of GRACE and Its Price Behavior

Buyya et al. [3] distinguished and named seven possible economic models that may be employed by GRACE: commodity market, posted prices, bargaining, tendering, auction, proportional resources sharing or shareholder, and community/coalition/bartering. The service providers were to share the common objective: "to maximize their resource utilization by offering a competitive service access cost in order to attract consumers." The paper listed important points with respect to price, such as the dependency of pricing policy on the objective of the system,[46] the need for careful selection of resources to be charged, and, the necessity of accounting and payment procedures, but did not examine how the proposed objective may affect pricing. Suppose the users are utility-maximizers with a limited amount of budget, which has to be carefully dispensed in order to run all the planned jobs. Then, since the amount of resources that are available for use is fixed in the short-run, the service providers with the above objective would attempt to maximize the utilization rate by altering the resource prices. One possible scenario is that, absent collusion, which follows from the definition of a competitive environment, underbidding of prices by the service providers continues until, at least, one of them provides resources at no cost.[47] If some users are rejected by the provider with free resources due to lack of capacity, they would turn to other providers, who would engage in a similar price war. As long as there are other providers who wish to get as many users as possible and maximize the rate of utilization, there will be undercutting of price until one provider sets it equal to zero. Another possible scenario is that free resources invite congestion and result in alienation of users, who opt for positively priced but not-so-congested resources. In either scenario, price fluctuations may be quite large.

Simulation Results

As a simulation exercise, CPU time was charged at a flat rate, but differently for peak- and

[46]The paper differentiates the formation of pricing in each of their models. In fact, price formation processes are essentially the same unless prices are imposed. For example, what they call a price determined by supply and demand is the same as what they describe as one determined by the objectives of service providers and users; supply and demand are shaped by the objectives of service providers and users.

[47]The case is akin to the Bertrand model of price competition, although the suppliers in the model are profit-maximizers. For the discussion of the Bertrand model, see, for example, Section 12.C of *Microeconomic Theory* [29].

off-peak-periods of a day, using a network of four machines and one cluster of computers. They were located at three geographically separated institutions, and each of them had the capacity of ten nodes. Therefore, although the service providers were supposed to maximize the utilization rate of their resources in GRACE, the resource prices were given to them, leaving no space for their optimization. This setting precludes the exercise from being a simulation of a model with economic application.

The geographical locations of the computers were Melbourne, Chicago, and Los Angeles, and thus, the Australian site and the American sites roughly alternated with respect to peak-period. The scheduler, which assigned tasks to various locations, was configured so that the cost (the paper is suggestive of the overall cost) of running jobs was minimized and all jobs were completed within an hour after each submission. The paper reported that the total cost for the computation of 165 jobs, each of which was "a CPU-intensive task of approximately 5 minutes in duration[,]" was smaller than that when the cost minimization algorithm was not employed. This result should not be unique to GRACE, but the result to be achieved by all properly functioning cost-minimization algorithms. With respect to the necessary condition to be an economic application, the users' utility maximization part was met if the utility of each job, which was only implied, was equal to the negative of the computation cost. Although not described explicitly, it appears that a budget was allocated for each job and no transfer of budget was permitted; the second part of the necessary condition was not satisfied.

4 Scheduling and Economics in Practice

4.1 Have We Applied Economics?

We have examined above the representative scheduling mechanisms for non-distributed and distributed computing systems, which aimed at incorporating economics. We employed the following criterion for judging that a scheduling mechanism was based on economics: confirmation that the outcome was achieved through utility-maximizing behavior of the participants, given absolute scarcity of resources and the existence of alternative uses of resources which lead to different utility levels (or alternatively, given resource use, which is associated with both positive and negative effects on utility that vary in accordance with the level of resource allocated, i.e., inseparability scarcity). The participants included computing-service providers, in addition to service users, if they interacted in the process of resource-allocation determination. If the desirable outcomes were achievable only with economically motivated behavior, that fact was sufficient to conclude that the scheme was based on economics. Further, we may say that a scheduling which utilizes an economic model is superior to other scheduling mechanisms, if the mechanisms are applied to the same situations and the performance of the one with an economic model is better in tune with the scheduling goal. We could not conclude that the criterion for an economic application was satisfied by any of the models examined. Moreover, there was no comparison of their performances with those using other scheduling mechanisms. Hence, no support was provided for establishing that economics is a necessary component for superior performance of distributed computing systems.

4.2 Bidding and Auction

Most of the mechanisms which we examined employed an economic procedure called bidding (Sutherland's, the Contract Net Protocol, the Enterprise, Spawn, WALRAS, Mariposa, and GRACE). We argue below that none of the biddings can be said to exhibit the functionalities of a bidding as we know in auctions.

Bidding is useful when bidders do not have clear descriptions of their own utilities or do not wish to communicate them in order to obtain the best deal. Compared to negotiation, bidding is more suited for multiple sellers and buyers, more open, and probably, less time consuming. An auction, which consists of biddings, is a forum for competing traders to express their desire to trade (i.e., to reveal preferences) so that the trade takes place in terms that are most favorable to them (i.e., to maximize utility from the auction). The expression of their desire is affected by the presence of competing traders or bidders. This is because the expression of unwillingness to trade works in favor of the buyer with respect to the final price, but it works against their establishing a trade due to other traders who are competing for the same trade. In other words, auctions reveal traders' preferences that are influenced by their perception of other bidders' preferences. Since we cannot say that any of the Contract Net Protocol, the Enterprise, Spawn, Mariposa, and WALRAS employed utility maximizing agents, we cannot conclude what they called biddings were those with basis in economics. Strictly speaking, the reporting by the local agents in tâtonnement processes in the general equilibrium theory are not bidding in a distributed computing environment, unless there are an infinite number of traders or redistribution of initial endowments is allowed.

Bidding assumes heterogeneity of bidders. There would be redundancy in collecting bids from all agents in the system if they are homogeneous. In addition, if there are no other uses for what is to be given up in order to obtain the item, the bidding strategy does not reflect the bidder's economic calculation, ridding an auction of its main function of preference revelation. That was the case in Sutherland's scheme and Spawn.

Differentiation of a resource through pricing, which is dependent on the time of the day, expresses the preferences of the price-setter, but does not fall into the category of auctions, if there is only one price at a time in the presence of heterogeneous goods, or if there are multiple prices, each of which is for a mutually unsubstitutable item. The prices were set uniformly across sites on the same continent by the administrative authority in the simulation example of GRACE, and therefore, we may conclude that it was an auction, only under the assumption that the sites on a continent offered identical services.

4.3 Future Direction

We have established that the problem in scheduling can be seen as that in economics under certain conditions. Many of the studies examined justified their use of economics based on the desirable results given by the general equilibrium theory, which is more narrowly focused than the whole discipline of economics. How relevant is the general equilibrium theory to scheduling? More generally, we should question the general equilibrium theory and markets themselves. The actual economy gives the impression of having the capacity to work well on its own, and the general equilibrium theory is often taken as a good abstraction of that seamless working. Does the theory capture what drives the economy to behave well as a system? If not, applying the general equilibrium theory to scheduling would not provide a "mechanism that somehow works successfully without intervention." Conversely,

are markets capable of producing the favorable outcomes as the general equilibrium theory implies (and subsequently, asserted by the architects of market-based scheduling)? We believe that many of these questions can be answered through a careful reading of the general equilibrium theory and its related fields.

Additionally, all the proposals for distributed systems that were examined, used, or implied the use of, artificial agents. Are they necessary and do they have the potential to provide better scheduling than without? Would the use of artificial agents in place of human users alter the workings of computing resource markets? These questions should be answered before we attempt to construct a scheduling mechanism which genuinely applies economics.

References

[1] Arrow, Kenneth J., "Toward a Theory of Price Adjustment," in *The Allocation of Economic Resources*, edited by Moses Abramovitz. Stanford, California: Stanford University Press, 1959.

[2] Arrow, Kenneth J. and Frank Hahn. *General Competitive Analysis*. San Francisco: Holden-Day, 1971.

[3] Buyya, Rajkumar, David Abramson, and Jonathan Giddy, "A Case for Economy Grid Architecture for Service Oriented Grid Computing," presented at the 10th Heterogeneous Computing Workshop, San Francisco, April 23, 2001.

[4] Casavant, Thomas L. and Jon G. Kuhl, "A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems," *IEEE Transactions on Software Engineering*, Vol. 14, No. 2, February 1988, 141-154.

[5] Chapin, Steve J., "Distributed and Multiprocessor Scheduling," *ACM Computing Surveys*, Vol. 28, No. 1, March 1996, 233-235.

[6] Cheng, John Q. and Michael P. Wellman, "The WALRAS Algorithm: A Convergent Distributed Implementation of General Equilibrium Outcomes," *Computational Economics*, Vol. 12, No. 1, August 1998, 1-24.

[7] Clearwater, Scott H., "Why Market-Based Control?", in *Market-Based Control: A Paradigm for Distributed Resource Allocation*, edited by Scott H. Clearwater. Singapore: World Scientific Publishing, 1996.

[8] Cocchi, Ron, Scott Shenker, Deborah Estrin, and Lixia Zhang, "Pricing in Computer Networks: Motivation, Formulation, and Example," *ACM Transactions on Networking*, Vol. 1, No. 6, December 1993, 614-627.

[9] Cotton, Ira W., "Microeconomics and the Market for Computer Services," *Computing Surveys*, Vol. 7, No. 2, June 1975, 95-111.

[10] Edell, Richard and Pravin Varaiya, "Providing Internet Access: What We Learn from INDEX," http://www.path.berkeley.edu/~varaiya/papers_ps.dir/networkpaper.pdf, April 1999.

[11] Edmonds, Thomas P., Frances M. McNair, Edward E. Milam, Philip R. Olds, *Fundamental Financial Accounting Concepts*, 2nd edition. Boston: Irwin, McGraw-Hill, 1997.

[12] Euro Grid, http://www.eurogrid.org.

[13] Ferguson, Donald F., Christos Nikolaou, Jakka Sairamesh, and Yechiam Yemeni, "Economic Models for Allocating Resources in Computer Systems," in *Market-Based Control: A Paradigm for Distributed Resource Allocation*, edited by Scott H. Clearwater. Singapore: World Scientific Publishing, 1996.

[14] Hayek, Friedrich A., "The Use of Knowledge in Society," *American Economic Review*, Vol. 35, No. 4, September 1945, 519-530.

[15] Heal, Geoffrey, "Planning without Prices," *Review of Economic Studies*, Vol. 36, No. 3, 1960, 347-362.

[16] Hootman, Joseph T., "The Pricing Dilemma," *Datamation*, Vol. 15, No. 8, August 1969, 61-66.

[17] Horngren, Charles T., Gary L. Sundem, and John A. Elliott, *Introduction to Financial Accounting*, 7th edition. Upper Saddle River, New Jersey: Prentice Hall, 1999.

[18] Huberman, Bernardo A. and Tad Hogg, "Distributed Computation as an Economic System," *Journal of Economics Perspectives*, Vol. 9, No. 1, Winter 1995, 141-152.

[19] Hurwicz, Leonid, "On Informationally Decentralized Systems," in *Decision and Organization: A Volume in Honor of Jacob Marschak*, edited by C. B. McGuire and Roy Radner. New York: North-Holland Publishing, 1972.

[20] Hurwicz, Leonid, "The Design Mechanisms for Resource Allocation," *American Economic Review*, Vol. 63, No. 2, 1973, 1-30.

[21] Institute of Electrical and Electronics Engineers. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. New York: 1990.

[22] Information Power Grid, Engineering and Research Site, http://www.ipg.nasa.gov.

[23] Konana, Prabhudev, Alok Gupta, and Andrew B. Whinston, "Integrating User Preferences and Real-Time Workload in Information Services," *Information Systems Research*, Vol. 11, No. 2, June 2000, 177-196.

[24] Korilis, Yannis A., Aurel A. Lazar, and Ariel Orda, "Architecting Noncooperative Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 7, September 1995, 1241-1251.

[25] Kurose, James F. and Rahul Simha, "A Microeconomic Approach to Optimal Resource Allocation in Distributed Computer Systems," *IEEE Transactions on Computers*, Vol. 38, No. 5, May 1989, 705-717.

[26] MacKie-Mason, Jeffrey K. and Hal R. Varian, "Pricing the Internet," in *Public Access to the Internet*, edited by Brian Kahin and James Keller. Cambridge, Massachusetts: MIT Press, 1995.

[27] Malinvaud, E., "Decentralized Procedures for Planning," in *Activity Analysis in the Theory of Growth and Planning*, edited by E. Malinvaud and M. O. L. Bacharach. London: Macmillan, 1967.

[28] Malone, Thomas W., Richard E. Fikes, Kenneth R. Grant, and Michael T. Howard, "Enterprise: A Market-like Task Scheduler for Distributed Computing Environments," in *The Ecology of Computation*, edited by B. A. Huberman. North-Holland: Elsevier Science Publishers, 1988.

[29] Mas-Colell, Andreu, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. New York: Oxford University Press, 1995.

[30] McAfee, R. Preston and John McMillan, "Auctions and Bidding," *Journal of Economic Literature*, Vol. 25, No. 2, June 1987, 699-738.

[31] McKnight, Lee W. and Joseph P. Bailey, "Internet Economics: When Constituencies Collide in Cyberspace," *IEEE Internet Computing*, November · December 1997, 30-37.

[32] Mendelson, Haim, "Pricing Computer Services: Queueing Effects," *Communications of the ACM*, Vol. 28, No. 3, March 1985, 312-321.

[33] Miller, Mark S. and K. Eric Drexler, "Markets and Computation: Agoric Open Systems," http://www.agorics.com/agoricpapers.html, 7 July 2000.

[34] von Mises, Ludwig, "Economic Calculation in Socialism," in *Comparative Economic Systems: Models and Cases*, 5th edition, edited by Morris Bornstein. Homewood, Illinois: Richard, D. Irwin, 1985.

[35] Nielsen, Norman R., "The Allocation of Computer Resources—Is Pricing the Answer?," *Communications of the ACM*, Vol. 13, No. 8, August 1970, 467-474.

[36] National Technology Grid, http://archive.ncsa.uiuc.edu/alliance/alliance/GridTech.html, http://www.terena.nl/middleware/grid-general.html.

[37] Robbins, Lionel C. *An Essay on the Nature and Significance of Economic Science*. London: Macmillan, 1984 (originally published in 1932).

[38] Sanders, Beverly A., "An Incentive Compatible Flow Control Algorithm for Rate Allocation in Computer Networks," *IEEE Transactions on Computers*, Vol. 37, No. 9, September 1988, 1067-1072.

[39] Personal communication with Cathy Schulbach, NASA Ames Research Center.

[40] Sidebotham, Roy. *Introduction to the Theory and Context of Accounting*. New York: Pergamon Press, 1965.

[41] Smith, Reid G., "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," *IEEE Transactions on Computers*, Vol. c-29, No. 12, December 1980, 1104-1113.

[42] Stankovic, John A., "A Perspective on Distributed Computer Systems," *IEEE Transactions on Computers*, Vol. c-33, No. 12, December 1984, 1102-1115.

[43] Stickney, Clyde P., Roman L. Weil, and Sidney Davidson. *Financial Accounting: An Introduction to Concepts, Methods, and Uses*, 6th edition. San Diego: Harcourt Brace Jovanovich, 1991.

[44] Stonebraker, Michael, Robert Devine, Marcel Kornacker, Witold Litwin, Avi Pfeffer, Adam Sah, and Carl Staelin, "An Economic Paradigm for Query Processing and Data Migration in Mariposa," http://sunsite.berkeley.edu/Dienst/UI/2.0/Describe/ncstrl.ucb/S2K-94-49

[45] Stonebraker, Michael, Paul M. Aoki, Witold Litwin, Avi Pfeffer, Adam Sah, Carl Staelin, and Andrew Yu "Mariposa: a Wide-Area Distributed Database System," VLDB Journal, Vol. 5, No. 1, January 1996, 48-63.

[46] Sutherland, I. E. "A Futures Market in Computer Time," *Communications of the ACM*, Vol. 11, No. 6, June 1968, 49-451.

[47] Tilley, Kevin J., "Machining Task Allocation in Discrete Manufacturing Systems," in *Market-Based Control: A Paradigm for Distributed Resource Allocation*, edited by Scott H. Clearwater. Singapore: World Scientific Publishing, 1996.

[48] Waldspurger, Carl A., Tad Hogg, Bernardo A. Huberman, Jeffrey O. Kephart, and Scott Stornetta, "Spawn: A Distributed Computational Economy," *IEEE Transactions on Software Engineering*, Vol. 18, No. 2, February 1992, 103-117.

[49] Wellman, Michael P., "Market-Oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems," *Journal of Artificial Intelligence Research*, Vol. 1, August 1993, 1-23.

[50] Wellman, Michael P., "Market-Oriented Programming: Some Early Lessons," in *Market-Based Control: A Paradigm for Distributed Resource Allocation*, edited by Scott H. Clearwater. Singapore: World Scientific Publishing, 1996.

[51] Zappia, Carlo, "The Notion of Private Information in a Modern Perspective: A Reappraisal of Hayek's Contribution," *European Journal of the History of Economic Thought*, Vol. 3, No. 1, Spring 1996, 107-131.