Market-Based Approaches to Utility Computing

Colin Low, Andrew Byde
Internet Systems and Storage Laboratory
HP Laboratories Bristol
HPL-2006-23
February 27, 2006*

utility computing, market-based control

Utility Computing is a means to purchase computational resources on demand. We believe the fluctuating nature of demand for these services makes it appropriate to analyse the problems of dynamic resource provision using concepts and techniques derived from microeconomics. This document is an overview of our approach and is intended for the non-technical reader.

* Internal Accession Date Only
Approved for External Publication

Market-Based Approaches to Utility Computing

Colin Low and Andrew Byde

{Colin.Low, Andrew.Byde}@hp.com

Hewlett Packard Laboratories,
Filton Road,
Stoke Gifford,
Bristol BS34 8QZ
UK

Abstract

Utility Computing is a means to purchase computational resources on demand. We believe the fluctuating nature of demand for these services makes it appropriate to analyse the problems of dynamic resource provision using concepts and techniques derived from microeconomics. This document is an overview of our approach and is intended for the non-technical reader.

1. Introduction

Utility Computing, described in more detail below, is a new approach to providing customers with bulk computer processing capabilities by making these resources available using high-speed networking. Many individuals and business already rent space on remote, externally managed WWW servers, and pay for a variety of services such as email, so the concept itself is not novel. What is novel is the scale of Utility Computing and its potential to meet the needs of a new range of compute and storage intensive applications for enterprise customers.

How should one sell bulk computing resources such as processing time, storage and network bandwidth? A simple method is to publish a tariff of prices: most retail outlets sell goods and services this way. Fixed prices are straightforward for consumers and suppliers to deal with (that is, either the price is acceptable or not), but they present drawbacks due to the fact that there is no flexibility in matching buyers and sellers. If the price is set too high, then the buyer cannot buy the goods, and if too low, then demand will exceed supply – either way, the seller does not make his maximum profit. These effects are minimized when there are many different buyers and sellers and where demand is in equilibrium with supply.

When there are many buyers, many sellers, or many goods for sale, fixing prices allows an exchange economy to function well, if not optimally. At the other end of the complexity spectrum, consider a fine-art auction, in which there is exactly one good for sale, one seller, and a handful of buyers. Since there is only one good, the seller only has one opportunity to determine the price of that good; getting the price right is critical to the feasibility of a sale, and the seller's profitability. The auction is a procedure for *price discovery*: working out the right price point at which the demand of buyers matches the supply of sellers.

This document outlines research carried out in HPLB to examine possible mechanisms for various compute-resource markets that we expect to emerge in the near future. We begin from the assumption that fixed tariffs may not be the correct answer; we envisage compute utility markets as something closer to spectrum rights auctions or bulk electricity exchanges than retail supermarkets, and look at technology to support dynamic pricing and resource-use optimisation.

2. Utility Computing

The premise behind Utility Computing is that the scale of computing resources required to support our lives will continue to grow, and many enterprises will be unable to justify the capital and technical expertise required to provide these resources in-house. It is already the case that companies choose to outsource the provision of IT services such as hardware maintenance, and in the future they will outsource the provision of the underlying resources – processing power, communication

bandwidth and data storage. Utility Computing is a vision of a world in which this type of outsourcing is painless and transparent. Potential suppliers of this vision (such as HP) are talking about a future in which large installations, capable of serving the computation needs of hundreds or thousands of customers, deliver IT services of all flavours: from processor-intensive fluid dynamic simulations for aerospace companies, through digital media processing, to web-sites for family photo albums, all from the same resource base.

The primary justification for this new approach is economic: large Datacenters can realize economies of scale in management and running costs. The maturation of the computer industry means that computing services have become pervasive in all areas of life, and many enterprises have huge computing needs, but do not see running computers as part of their core competence (any more than they do building services, site catering or site security).

There are two technology enablers: the maturation of the Internet provides high-bandwidth network access to remote resources at economic prices, and security technologies have reach a point where it is possible to conceive of extending in-house resources to include dynamically-assigned external resources.

We see three main players in the Utility Computing Market. Fabric Providers (FP) provide bulk processing, storage and bandwidth resources. Service Providers (SP) purchase capacity from Fabric Providers, purchase licenses from Application Providers, and sell computing service *bundles* to end-users, who are also the paying customers. This is illustrated in Figure 1.

The scale on which customers might purchase virtual resources is potentially very large. For example, rendering a Hollywood movie, or sequencing a genome, could use hundreds or thousands of processors.

Utility Computing is still in gestation, and there are many important business questions to which one would like answers. What market segments will be most important, and what differentiating factors will be important in determining segments? What kinds of transactions will take place between market players? What kinds of enforceable agreements will be used? What are the important timescales for resource transactions: days, weeks, or months?

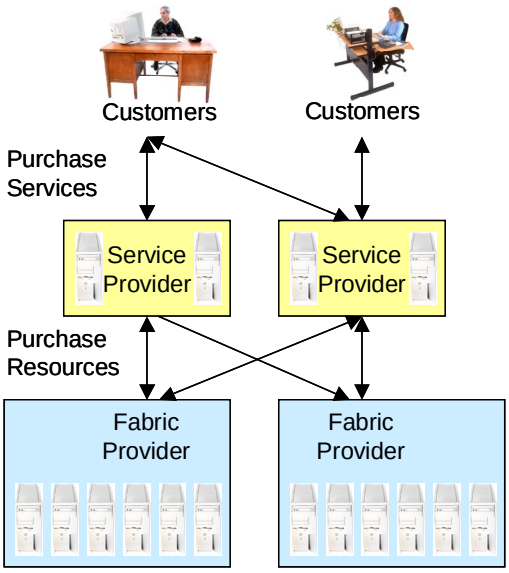

Figure 1: Market Players

It is the business drivers that will determine how resources are sold. Our research cannot provide answers to these questions at this time. What one can do is to propose hypotheses and examine the technological consequences. An observation we can make is that certain services, such as WWW and media services, have peak hour demands that can be ten times greater than quiet-time demand[1]. Applications such as finite element modeling, geological survey analysis, genome sequencing and film rendering can require enormous resources within a defined time window, and then nothing outside of that window.

The assumption that demand of every type can be aggregated to produce the steady, stable demand that would be *required* to support stable price tariffs does appear to be an act of faith. We have chosen to explore the alternative idea that demand will be variable and unpredictable, and under this assumption dynamic pricing models become attractive.

3. Application Models

Some kinds of application have large and variable resource requirements. For example, WWW sites can experience 10-1 variations in load between busy and quiet times, and even larger variations on special occasions (sporting fixtures, news events, promotions, seasonal surges etc).

[1] Andrzejak, Artur; Arlitt, Martin; Rolia, Jerry; *Bounding the Resource Savings of Utility Computing Models*, HP Laboratories Technical Report HPL-2002-339.

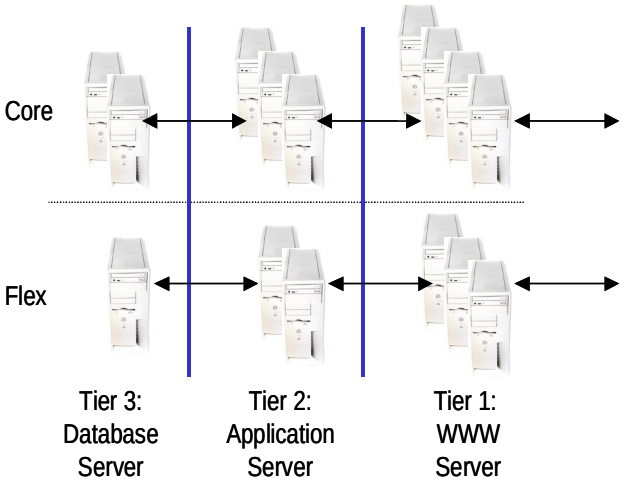

Figure 2: 3-Tier WWW Application Template

The Three-Tier WWW application template is shown in Figure 2, and provides an abstract resource model for a large class of WWW services. A pool of WWW servers provide the first line of interface to customers, and communicate transactions to a pool of Application Servers, which in turn access a pool of Database Servers. One way for a system like this to respond to large load variations would be to grow and shrink the respective pool sizes so that the time taken to execute the average customer transaction remains constant in spite of changing load. This suggests that a typical 3-tier service would specify a minimum **core** size for each pool, and would **flex** the pool sizes in a way that kept key application performance metrics within acceptable levels. A closed-loop design that achieves this using a market to trade resources is described in Section 6.1 below.

A second kind of application we have studied is frame rendering for film animation and special effects. An animation frame is described by a geometric model, textures, lighting and many kinds of special treatments, and these are transformed into the finished output frame. As films normally have 25 frames for every second, and frames can be rendered into high resolution for large-screen projection, the processing requirements can be massive. This application differs from 3-tier in that it is difficult to define a metric that the application must satisfy to create the kind of closed-loop control described above. Frames can vary by orders of magnitude in complexity, and so rendering a single frame could take seconds or several hours. Unlike 3-tier, where the minimum core pool might be one tenth of the flex resource, this kind of application has a large pool of core resources, and may have little or no flex resource. The need to reserve and hold a large pool of resources for a contiguous and bounded period of time makes this an interesting application for study.

4. The Role of Markets

The essence of Utility Computing is the on-demand supply of resources to customers as an economic alternative to ownership. It is inevitable that one would want to consider the structure of this market. It is possible (and easy, even within this document!) to confuse this with Market Based Resource Allocation (MBRA). MBRA is an application of microeconomic theory to resource allocation, and in many cases MBRA can be regarded as a decoupled and distributed control mechanism [3]– that is, there are control problems involving resources, and one way to address these problems is to use the vocabulary and theory of microeconomics. The market becomes an artificial creation used to solve a problem, in the same way as Lagrange Multipliers are variables introduced to solve constraint problems in applied mathematics.

An important quantity in a market is the price p_i paid for good x_i. Prices are the primary observables in a market, and serve as a source of information about the scarcity of different goods. Prices also provide a control function in that they transform income into resources. The first and second Welfare Theorems of microeconomics show that under reasonable mathematical conditions, there exist equilibrium states that are fair (relative to the *a priori* distribution of income) – that is, each player is as well off as they can be, and none can be better off without disadvantaging someone else.

This global good can be achieved without coordination, and in spite of each player in the market acting selfishly. It is this result that suggests that market-based methods should be

useful in distributed computing, where any kind of agreement or consensus protocol is expensive to implement.

Another aspect of markets we have had to consider is *mechanism design*, the design of specific markets that allow players to purchase and trade resources, while achieving useful social goals. This falls outside pure MBRA techniques, but is an important part of the customer interface to Utility Computing.

Our work has considered the use of "markets" in two very different contexts, characterized by the market owner:

- Fabric Provider markets for bulk commodity computing resources.
- Service Provider internal markets for efficient resource partition between applications.

In the first case we have studied *market mechanisms* to permit market users to express preferences for reserving and purchasing resources. In the second case we have used a pure MBRA approach within a SP to *maximize a global good*, namely the revenue associated with a pool of resources.

Both these scenarios describe price-based economies, driven by "buyers" that have money but no resources, and "sellers" that have resources to sell, and want more money. A buyer has a certain demand for resources at each price, defined as the number of resources he would be willing to buy if the price were at a given level. Likewise a seller has a supply function that describes the number of resources he would like to sell at a given price. When the buyers and sellers adjust their bids and offers for goods according to observed trading, they converge to the *equilibrium* price at which the amount of goods demanded by the buyers and the amount of goods supplied by the sellers is equal.

If prices are too low, buyers will demand more resources than sellers are willing to provide, and so some will be willing to pay slightly more in order to avoid getting nothing at all, which leads to a price rise. Likewise if prices are too high, sellers will want to sell more than buyers are willing to buy, and those sellers that are left out have an incentive to lower their prices rather than not sell at all.

The central question of market mechanism choice is: how does one discover the equilibrium price? This question is important from the perspective of a supplier of utility computing because if prices are too low, demand will exceed supply and paying customers will be turned away, and if prices are too high, resources will sit idle. In both cases income is less than it should be.

In consumer markets – PCs for example– prices are determined by the sellers, who make educated guesses regarding likely demand for their products, and set prices accordingly. This process works well when supply and demand is adequate and slowly changing relative to the speed at which prices can be changed.

In exchanges like the London Stock Exchange, buyers (and sellers) of a security enter orders that consist of a number of units, and a maximum (minimum) price. When buy and sell prices overlap, a trade occurs. This type of order book (and related mechanisms) is common for stock markets, and leads to efficient exchange – even if supply and demand are rapidly changing – so long as there is enough *market liquidity*, i.e. enough orders on the book.

The internet has led to a renaissance for auctions of all sorts. Auctions allow price discovery for goods for which there is only limited supply. Although it may one day be the case that Utility Computing is bought and sold on high volume exchange markets like the London Stock Exchange, the demand and supply for these resources is not likely to reach sufficient volumes for some time, and so such mechanisms are not yet applicable. In the meantime, the best way for utility computing suppliers to adapt to rapidly changing demand for resources may well be for them to organize futures and spot auctions.

Our research in HP Laboraties deals with suitable mechanisms for such auctions, including issues such as, how resources are bundled for sale; how buyers express their demand; what are the protocols for communication between the various parties; and what strategies are agents likely to pursue. This last issue is inextricably linked with the previous issues: buyers will adapt their behaviour to the market mechanism, and the market mechanism should be adapted to the behaviour of the agents.

5. Resource Use Optimisation

The revenue obtained from renting out a resource depends on the amount of time the resource is in use, and on the rental charge. The key measure is the average *rate* at which a resource earns revenue.

A problem with the provision of public transport is the huge demand surge during rush hour, and the pressure from commuters to provide sufficient capacity to meet peak demand. This is why public transport systems are rarely profitable: buses and trains are sitting idle for most of the day. If insufficient

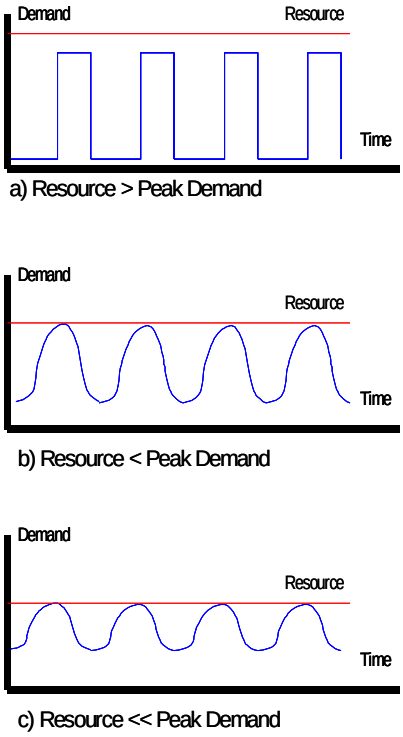

Figure 3: Demand Spreading

capacity is provided to meet peak demand, admission control comes into play, and in public transport this simply means you cannot board a bus or train because it is full. What happens in practice is that commuters time-shift their commute to less busy periods. The less capacity is available to meet demand, the more commuters are forced to travel outside of peak hours, so that in some areas the evening rush hour now lasts from 3.00 pm until 8.00 pm.This is shown in Figure 3. In 3a, the resource/capacity is in excess of what is required, and the demand can be accommodated. In 3b, capacity is less than demand, and some demand has been shifted unto less popular times. In 3c, there is even less capacity, and the natural demand curve in a) has been forced into a very different shape.

This kind of behaviour is similar to that of animators rendering their scenes into images. Animators typically prefer to work during the day, and render their work overnight. If rendering resources were unlimited, one might see the behaviour in a). As the rendering capacity is reduced in b) and c), animators are forced to render at less convenient times.

There needs to be an incentive to time-shift one's commute or rendering task. In the case of commuting, the incentive is a more comfortable journey with reduced journey time. In the case of rendering, the incentive would be a price differential between popular and less popular times. This example is instructive in that it shows how an admission control mechanism shifts demand to less popular times, and has the effect of increasing resource utilization. It also serves as a reminder that profitable resource utilization and customer satisfaction have contradictory objectives. When resources are fixed, and demand is variable, it is not obvious how to strike a balance between the two objectives.

6. Research at HP Laboratories

Our research in Hewlett Packard has looked at two circumstances under which one might use Market-Based Control, which one might term *cooperative* and *competitive* scenarios.

In the cooperative scenario all players use the same mechanism/framework for evaluating their resource needs, and are honest in translating their evaluations into bids for resources. Although players appear to be decoupled and decentralized, they have an implicit connection in their use of an identical evaluation framework, and in their commitment to deal honestly for the common good. This economic scenario is suitable for the context of internal markets for resource allocation within a Service Provider

In the competitive scenario the only point of connection between players is the resource auction, and we can assume that players will use arbitrary amounts of ingenuity to deal advantageously. If a market mechanism has design flaws, we must assume they will be exploited. This scenario is applicable to the markets Fabric Providers will organize to sell their bulk utility resources.

Both the cooperative and competitive scenarios are useful. The cooperative scenario has been used as a devolved and distributed control mechanism, and an interesting application of this is central heating control for an office complex[2]. The competitive scenario is more

[2] Huberman & Clearwater, *A Multi-Agent System for Controlling Building Environments*, ICMAS 95. It is worth reading this in conjunction with the analysis of Ygge & Akkermans, in *Decentralised Markets*

generally useful as it makes fewer assumptions, but suffers from the potential complexity of the market mechanism and theoretical difficulties in understanding how markets of this kind work in practice.

We have applied these two variants of market-based control to two distinct levels in the Utility Computing market: as a means to regulate the use of resources within a single Service Provider, and as a way for Service Providers to trade resources with Fabric Providers.

A Service Provider will have a pool of resources bought from Fabric Providers, allocating resources to end-customers to provide a specified quality of service. The Service Provider's goal is to minimize the use of resources consistent with the Service Level Agreements it has with its customers. The Service Provider can embody each Service Level Agreement within an agent (a Service Manager), and agents can use cooperative market-based control to ensure that each application receives precisely the level of resource it needs.

The Service Provider will acquire resources from Fabric Providers. A core of these resources may be acquired via a long-term leasing arrangement, but we are exploring the possibility that a substantial part of these resources may be acquired dynamically via two types of competitive auction, the reservation market and the spot market, bidding against other Service Providers.

These two uses of Market Based Control are described in sections 6.1 and 6.2 below.

6.1. Customer-Service Provider

We assume that the contractual basis between a customer and a Service Provider is a Service Level Agreement (SLA), defining metrics relating to the performance of a given application. A payment tariff is agreed that depends on the relationship of the actual measured performance to benchmark performance. In most cases the customer will define a target performance level, and the essence of the SLA is that the SP is paid for providing that performance level. The SP may incur penalties for substantial underperformance.

There is no economic or contractual reason why a SP should over-resource an application. The marginal utility from adding additional

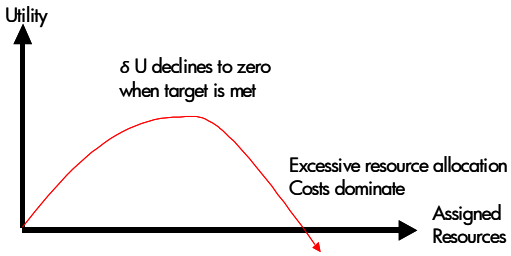

Figure 4: Marginal Utility of Resource Allocation

resources to an application will decline to zero at some point, and then go negative as the cost of power and cooling dominate any additional income. This is shown in Figure 4. If an SP is running applications on behalf of many customers, it is desirable to balance resources between applications so that the global utility of all resources is maximized – that is, each resource is being used as well as it can be.

Market-Based Control provides a simple, elegant and intuitive method for balancing resources across applications. A management component, the Service Manager (SM), is associated with each running application. The SM monitors application performance metrics and feeds them into an SLA Manager module that compares current with target performance. It uses these measurements to calculate a bid price for each resource type used by the application.

The SM sends its prices to a centralized

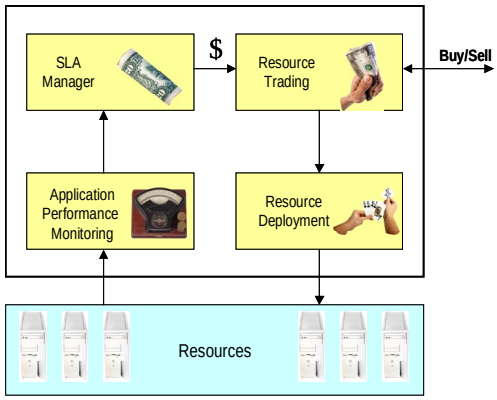

Figure 5: Closed-loop Resource Control

Resource Manager, which compares the marginal utility, represented by the bid price for each resource type in its resource pool, and then makes a decision about how to move resources between applications. If necessary it could acquire more resources from a FP, or release resources back to a FP – the bid price provides a simple mechanism to link internal

versus Centralised Control: A Comparative Study, Journal of AI Research, 11, 1999.

resource provision within a Service Provider to external markets for resources, so that there is a direct link from the portfolio of customer SLAs through to dynamic resource trading using real money.

If the resource set of an application changes as a result of trading, the SM re-deploys application components, the effects of which feed into performance measurements. This closed control loop is shown in Figure 5.

6.2. Service Provider-Fabric Provider

We assume that while a SP might have a stable core of resources leased from FPs on a long-term basis (many months), there will be a means to add and reduce capacity in real-time by buying and selling resources in an open market. A **good** is defined as "a specific type of computation resource for a period of time". Customers will typically want several computer resources for a period of time – that is, they will want a bundle of goods.

We are studying two types of auction mechanism. The first is a **reservation** market, where rights to goods are sold in advance of use. The second market is a **spot** market where resources are sold for immediate use. We anticipate that auctions of both types will be available, so that SPs can both reserve goods in advance as a response to known customer needs using the reservation market, and cover unexpected shortfalls in real time using the spot market.

In the following descriptions, we assume a single FP is selling goods to a number of SPs. This could be generalized to a market where multiple FPs trade with multiple SPs, but we have not yet modelled this case.

Reservation Market

The reservation market enables an SP to purchase a bundle of goods in advance of use. A percentage of unsold FP resources are auctioned, and participants have the opportunity to express their demand for bundles of goods covering multiple time periods. For example, a typical purchase request might be:

"I want to purchase the rights to 15 'computation units' beginning at 18.00 on Friday, for 8 hours."

Only a percentage of available (unbought) goods would be sold at each auction. This is to prevent a "land rush" when the market opens. Goods purchased can be re-entered into either the reservation market or spot market. Goods re-sold in this way earn money for the SP rather than the FP, although the FP could deduct a seller's fee. We anticipate that a futures market auction will be held at regular, relatively infrequent intervals (e.g. daily).

Market Mechanisms

Multiple units of computation in different time periods are sold in the reservation market, and so there are many ways in which such a market can be organized. For example, the goods for different time periods could be sold in separate parallel ascending auctions; the goods could be sold in a **sealed-bid** auction in which the participants place bids for bundles of goods, and when the auction closes, bids are compared and a set of winners is determined. Participants might have to bid for specific resources, or might be allowed to bid for satisfaction of a constraint such as:

"$45 for any 50 units of computation between Friday 5pm and Monday 9am."

Some mechanisms are easier for humans to comprehend and bid in; some are more computationally tractable, for the auctioneer; some are proven theoretically to lead to efficient outcomes. Balancing these interests so as to determine the best mechanism for selling these computation futures is a research topic in its own right.

Spot Market

The spot market sells short-duration goods for immediate use. For example, one might hold auctions every hour for goods that last two hours. The purpose of the spot market is to allow last-minute demand to be met by resources that have not yet been sold, and would go to waste if they were not sold immediately. All goods that are currently unused are auctioned. As mentioned above, goods bought on the reservation market can also be entered into the spot market for immediate sale and use.

As before, there are multiple possible mechanisms for the spot market.

Bidding Agents

In addition to the question of how to organize the underlying market, there is the issue of what tools can be provided that enable participants to express their demand in a convenient manner. eBay for example, provides "proxy bidders", since it is impractical to continuously monitor a

continuous ascending auction that lasts several days: instead, you tell the proxy bidder your maximum willingness to pay, and it bids on your behalf. We are looking to develop such helper agents, not only to alleviate participants from the dull business of bidding, but also to handle strategic decision-making in complex auction environments.

Revenue Optimisation

The FP's goal is to maximise its revenue from selling computation resources. The outcome of a series of reservation auctions is that a schedule is created which allocates resources to specific customers at specific times. Customers must be chosen so that their concurrent resource needs do not conflict (that is the point of creating a schedule) and customers should be chosen so that revenue is the maximum possible given the various bids that have been received.

The class of problems to which this type of optimization problem belongs is known to be NP-complete (except in special cases), and we anticipate that formulating the various auction mechanisms so as to be computationally tractable will be a challenge. However, there are powerful approximation techniques for solving problems of this type. They are extremely complex, and one would normally consider using a commercial solver (e.g. CPLEX). An approximation technique using multiple bidding agents is also being evaluated.

Arbitrage & Speculation

Arbitrage is the exploitation of price differences in time and space for risk-free profit; speculation is buying and selling in the *expectation* of profit.

Speculation and arbitrage appear to be a parasitic activities, in that they perform no useful economic function. This is not the case. Arbitrageurs can help to stabilise prices by mopping up surpluses and by supplying demand surges. If someone buys goods, and then does not need them and wants to resell, the price obtained will depend on the buyers. If there is a pool of arbitrageurs who buy purely for profit, the price will never fall too low, because a low price represents a profit opportunity and arbitrageurs will bid against each other for the right to buy. Likewise, when many people want to buy, and the price begins to rise, arbitrageurs will want to realise profits and begin to supply the market, preventing prices from rising too high.

It is almost impossible to prevent arbitrage and speculation, and the only issue is whether it is overt or covert. Financial and commodity markets encourage it, and one aspect of legitimising an activity is that it can be monitored, regulated, and a profit can be taken from it.

The Schedule

The result of sales in the futures and spot markets is the formation of a concrete schedule. The schedule maps abstract resources to customers at each point in the future. The schedule has the following properties:

- The number of abstract resources allocated cannot exceed the number of physical resources available, and so
- customers are guaranteed the resources they have paid for.

The schedule is read by the FP Resource Manager and used to create a runtime allocation of physical resources. This is illustrated in Figure 6.

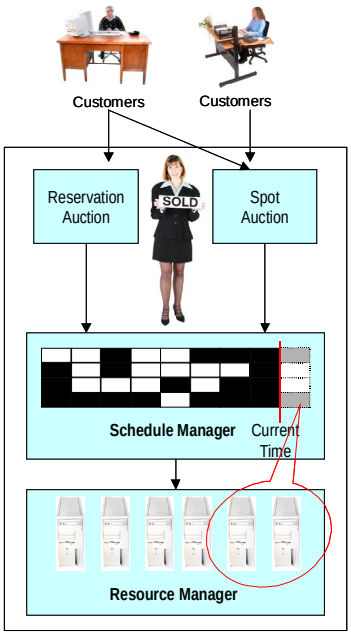

Figure 6: Schedule Creation & Resource Allocation

Issues

Customers will almost certainly want to express preferences in the time and resource domains. Preferences involving time would define a time interval during which a task could be executed, and might include a latest

completion time or a maximum-cost specifier. Preferences involving resources would specify the composition of bundles – minimum, optimal and maximal resource sets required for a task.

One would like to find the highest-value set of bundles and pack them into the schedule at the highest density while avoiding resource conflicts, so maximising the rate of revenue given the current set of customer preferences.

Bundle pricing is known to be a difficult problem. If customers can bid for bundles in an auction, these bundles are likely to overlap, and so a single resource has no unique value – it has a different value in every bundle in which it is present. When one attempts to find the winners in this kind of auction one must consider an exponentially growing number of combinations. In fact, the winner determination problem in combinatorial auctions is known to be NP-complete.

The scheduling problem has similar complexity, and is also NP-complete.

What this means is that calculating the optimal resource allocation to satisfy a given collection of customer resource and time preferences can be intractibly complicated, and one is forced to consider approximations. What one sees in related problem areas is a large number of special-purpose approximations that attempt to exploit some constraint on the general problem. Our current research is exploring two approaches. The first is to formulate the problem for a commercial, state-of-the-art non-linear optimiser (CPLEX) in order to explore the scaling limits of this approach[3]. The second is to use multiple parallel ascending auctions to acquire bundles of goods.

6.3. Micro-Optimisation

We have explored the use of Market-Based approaches for micro-optimisation within a Data center. One of us (AB) has investigated bidding for resources with a cost function that includes location, bandwidth and temperature factors. This provides an elegant way to exert soft control over the allocation of resources – for example, resources in cool racks cost less than resources in over-heating racks. One of us (CL) has investigated barter between resource users to achieve broadly comparable goals.

7. The Grid

The Grid shares many concepts with Utility Computing, and in the future may share standards-based architecture and interface specifications. There is a different emphasis however. The Grid is primarily an environment for scientists to share resources for exceptionally demanding tasks, and until now this has been done in a collegial way. There is definitely an understanding that charging for resource use will be required at some point, but the commercial imperative to maximise return on assets is not currently there.

An extensive survey of past and current research on Grid economics can be found in Buyya's thesis [5].

8. Conclusion

The value proposition behind Utility Computing is that it can meet a customer's performance, management and security requirements and still provide computing resources at a lower cost than purchase and self-management. It could be articulated as "We can do *at least* as good a job as you, *and* it will cost you less". A Utility Computing vendor has to prove two things. The first is that they can do "at least as good a job". Having proved that they meet the minimum requirement even to be considered as a supplier of computing resources or services, a key selling point will be cost savings.
This will place pressure on vendors of Utility Computing to examine revenue and cost structures with great care.
The essence of our work in HP Laboratories in Bristol has been the maximization of value from goods, where a good in this case is a resource deployed for a period of time. We have looked at the maximization of value using microeconomic theory. We have found that focusing on maximization of genuine economic value has a ripple effect through every resource allocation decision – that is, we find it difficult to conceive of Utility Computing without studying it from a MBRA perspective.

9. Further Reading

For an introduction to microeconomic theory, see

[1] Varian, H. R. (2002), *Intermediate Microeconomics: A Modern Approach*, 6th Edition, Norton

and

[3] Andersson, Tenhunen & Ygge show that the winner determination problem in a combinatorial auction can be formulated as a mixed integer programming problem, and that a commercial solver (CPLEX) is competitive with the best special algorithms. See *Integer Programming for Combinatorial Auction Winner Determination*, Fourth International Conference on Multiagent Systems, Boston, 2000

[2] Katz, M. L. and Rosen, H. S. (1998), *Microeconomics* 3rd Edition, McGraw-Hill.

A selection of papers illustrating the use of market-based techniques can be found in

[3] Scott H. Clearwater (ed.), *Market-Based Control: A Paradigm for Distributed Resource Allocation*, World Scientific, Singapore 1995.

An introduction to the traditional theory of scheduling, in particular, *job-shop scheduling*, can be found in

[4] Richard W. Conway, William L. Maxwell, Louis W. Miller, *Theory of Scheduling*, Dover 2003.

For extensive references to prior work on market-based scheduling and Grid economics in general, see

[5] Rajkumar Buyya, *Economic-based Distributed Resource Management and Scheduling for Grid Computing*, Ph.D Thesis Monash University, Melbourne, Australia April 12, 2002, available at http://www.cs.mu.oz.au/~raj/thesis/