Swing Options: a Mechanism for Pricing IT Peak Demand

Scott H. Clearwater
P.O. Box 620513
Woodside, CA
clearway@comcast.net

Bernardo A. Huberman
HP Laboratories
Palo Alto, CA 94304
huberman@hpl.hp.com

Abstract

Since usage patterns of information technology within organizations can be bursty, the peak demand for IT resources can at times exceed the installed capacity within the enterprise. If providers of such peak capacity emerge, as was the case for electricity and natural gas, the problem arises as to how to efficiently provide and price such peak demand.

We present a swing option mechanism that allows for the efficient pricing of IT resources ranging from CPU usage to storage and bandwidth. This mechanism allows users to buy the right but not the obligation to future peak use. A statistical simulation tool allows the users to price these swings according to their own utilization patterns and to recover some of their costs if the options are not exercised. The provider in turn exploits its ability to statistically multiplex its resources to price peak usage. The use of these swing options serves as an incentive to the users to accurately forecasts of their own needs, thus leading to more efficient utilization of the provider's resources.

Introduction

As computerized tasks and services become more prevalent within companies, a greater burden of expertise falls on those companies whose primary mission may not be information technology (IT) management. In addition, certain enterprise applications suffer from bursty usage patterns, whereby the peak demand IT resources at given times exceeds the installed capacity within the organization. These two trends will lead to the emergence of companies that offer IT peak capacity on demand for a given price, playing a role similar to many utilities such as electricity or natural gas.

The emergence of IT outsourcing has in turn created a number of problems for both providers and customers. On the provider side, the issue of how to charge for peak demand will depend on the statistical multiplexing that results from serving several customers at the same time and the lack of predictability for particular bursts in demand. On the customer side, there needs to be a simple way of figuring out how to anticipate and hedge the need for peak demand, as well as the costs that it will add to the overall IT operations.

The issue of peak use of IT in a non-trivial one for many customers. A number of studies have found that IT-related activities, specifically network traffic, is both bursty and heavy-tailed[5, 10]. Additionally, files systems, video traffic, and software caches have also been found to be extremely spiky in use [2, 4, 12]. Thus, the existence of so many sources of peak demand in computation makes provisioning by a single firm prohibitively expensive, as it would require extra resources that are only intermittently used. It is therefore clear that computational resources are prime candidates for a peak plant service that can accommodate multiple customers with varying requests.

Peaking plants are traditionally used by the power utilities to handle anomalous demand such as occurs on very hot days. Peaking plants are more expensive to run than normal plants but they save money in the long run because they are used infrequently and spare the utility the cost of building very large new power plants that would run at uneconomically low levels of utilization. In the case of IT peaking plants, the convenience of on-demand resources is novel to customers and not taken for granted as in the case of electricity, where customers expect lights and devices to go on every time that they are switched on. Consequently, IT customers would expect to be charged a premium for the use of a peaking service given the undesirable alternative of owning and operating what could be significant resources that for the most part would remain idle.

Peak use occurs in two basic ways. It can either be bursty, infrequent, unpredictable and large, or it may be frequent and small, appearing as a fluctuating signal above a threshold of utilization. These two profiles of peaks are illustrated in Fig. 1.

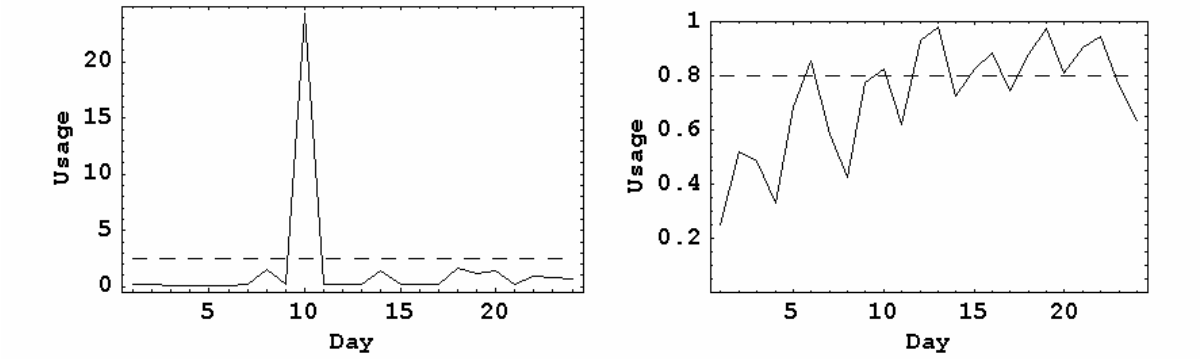

Fig. 1. (left)Bursty peak usage. (right) Noisy peak usage.

The economically feasible computational resource peaking plant we describe in this paper solves both of these problems by relying on its ability to share IT resources among a number of customers. Resource sharing is possible because the IT provider has admission control over who can use the peaking plant resources and may therefore select a portfolio of customers with a desired degree of overlap in their usage in order to manage the risk of a resource shortfall.

To efficiently accommodate customer resource demands the resource provider generates a schedule of swing options representing rights to use computational resources at a particular time without incurring the obligation to exercise the right. Swing options, also called flexible load contracts, are a type of exotic contract often used in electricity and natural gas, as well as other commodity markets[6]. These contracts incorporate flexibility of delivery in a take-or-pay manner. Specifically, swing options address the need by customers to frequently adjust their demand for goods that are not storable, as is the case with IT. As pointed out in [8] a swing option "is of value in any market where the physical transfer of the underlying asset must take place through interconnected networks, and is thus subject to volume constraint." In the case of swing options for IT, the assets are the bits and the programs that manipulate them, including storage, and the physical transfer is realized through a computer network, to and from a disk, or within a computer's memory and processor.

In the context of energy markets there is a literature that discusses schemes for fair swing prices[1, 6]. In [6] in particular, historical prices in a real market along with contract constraints are used to construct a swing option valuation framework. This framework provides a basis for determining fair swing prices under a number of reasonable constraints that are still simplifications of real market conditions.

By contrast, in the swing option service envisioned here, the resource owner sells the initial rights to the swing options to customers. Furthermore, the resource owner can buy back swings at a discount to the original purchase that the customer no longer wants. The discounted buy back depends on how far in advance the customer decides not to exercise

the swing option. The main advantage of swing options is that they act as incentives for predictable usage. This benefits the resource provider because resource planning can be done ahead of time, which is less costly than providing for peak demand of IT on a short-term basis. Predictable usage also provides a higher degree of customer service satisfaction. Thus, customers benefit from predictable usage because they can be assured that resources will be available when they are needed and they don't have the burden of maintaining them when they are not needed.

In the sections that follow we first describe the architecture of a swing option service, followed by the swing option pricing model. Next, we describe an analysis of such a service using simulated data and report on the results. We conclude with a number of issues that need to be addressed in order to make this scheme feasible.

Architecture of a Swing Option Service

Fig. 2 shows a schematic diagram of how the swing option service works between a resource provider and customer. The resource provider has a database of previous customers' usage and charges. Combined with a new customer's historical usage the appropriate prices can be set for this customer or for all the resource provider's customers. The pricing algorithm does a cross-validation simulation study (detailed below) using the historical data to set the appropriate price for services that insure a set profit margin with a high degree of confidence set by the resource provider. A resource scheduler then offers a calendar of available resources to the customers, who in turn pay the published swing contract price for usage and the contracted swing ranges. Individual customer usage variation accounts for differences in the contributions to the resource provider's profit, both from penalties and from swing trading. At the end of a billing period the resource provider calculates all the charges and sends a bill to the customer. The customer has access through a web interface to cumulative charges as well as estimated charges for the current billing period.

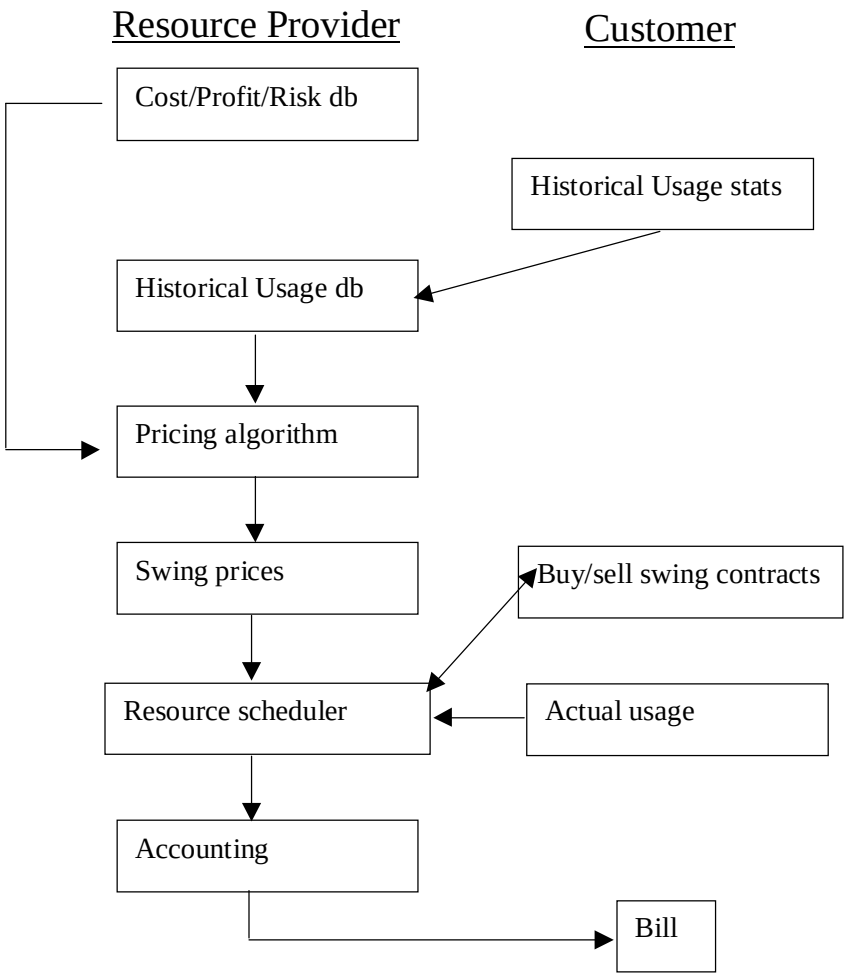

Fig. 2 Swing option service architecture.

Swing Option Pricing

In deriving a pricing model for a swing option service we need to parameterize the costs of providing the resource and the costs of the options so that a resource provider may "dial-a-yield" for the amount of profit margin return they would like to have. For example, the resource provider may derive most of its profit from swing contracts or from swing range violation penalties or swing options reselling (churning). Churning is profitable to the resource provider because a user selling swings back to the resource provider does so at a discount to its original purchase price. The resource provider may then resell that swing option and the future usage it represents to another customer. The resource provider thus can have multiple income streams for the same resource. By moving the profit away from the average usage as in a pure pay-for-use model and over to the "fluctuation" side of the usage, the swing option service is essentially a fluctuation-based or range-based pricing model.

As mentioned above, the inherent fluctuations in demand help to make the swing option services a profitable model. Working against profitability is the non-stationary nature of the demand. To account for changing patterns of usage the service provider has the ability to change its prices to maintain a predictable and consistent profit margin. Users trying to game the provider by providing incorrect historical data would have only a short arbitrage period before the user's new historical data takes precedent and obviates the arbitrage.

In this swing option service for peak demand, a customer is provided with a tool for estimating its cost given a set of historical data as well as the provision for entering the customer's assumptions about being aggressive or conservative with swings. Because the prices for swings are set ahead of time and not by market forces, the forecasting tool provides a powerful "what-if" capability to both the resource provider and the customer for estimating outright costs as well as risk associated with fluctuations in customer demand.

The ranges defined by swing contracts determine consumption boundaries that enable the customer to hedge its resource use[7]. Although the hedge costs money in the short run, it gives the consumer a flexibility in the long run that would otherwise cost more than the hedge did in the first place. The hedge benefits both the resource provider and customer because it incentivizes predictable usage. Namely, the range of the swing hedge is a predictor of future resource usage because going outside the swing range incurs a significant penalty.

A given customer will require a certain number of swings into order to cover the expected range of use within some degree of confidence. This number will depend on its own historical fluctuations in IT use. Less fluctuation in use implies that fewer swings are required to cover the expected usage and therefore a lower swing cost to the user. Higher usage fluctuations imply that more swings will required.

For example, if a customer expects to use 300-500 CPUs but has contracted swings for 400-500, then it will still be paying for 400 even though the usage is sometimes 300-400. Having swings of 300-500 is cheaper than paying for the lower bound of 400. However, due to the escalating cost schedule as the time for swing option exercise approaches, in some cases it may not be worthwhile buying additional swings on the low side if it is at the last minute. Note again that providing accurate historical data is incentivized because going outside the expected swings is more costly to the user than staying within predicted bounds of the swings.

As mentioned above, one of the key issues facing any consumer or producer of any resource is the peak use. From the producer side this means maintaining a potentially large standby reserve to instantly satisfy any burst in consumer demand. Obviously maintaining a large reserve is costly and those costs are passed onto the consumers. In a competitive environment, the provider that is best able to manage peak demands has a big advantage. The swing option service is one means for managing peak demands in a low-risk way. Risk is reduced because of the use of historical usage statistics and in the customers preference for swing contracts that are biased to incentivize more predictable and therefore less risky demands.

From the consumer side demand can vary significantly (a factor of ten) and unexpectedly for example due to changes in deadlines, or due to transient but predictable demands such as end of the month or quarter computations.

Managing computation resources has a big advantage over electrical power management. Specifically, computational demand can often be swapped in an out of the resource grid by the producer rather easily while for an electrical power grid the producer is largely at the mercy of demand and must take special non-automated actions to avert potential problems with small reserves such as public service announcements and calls to large industrial firms to curtail usage. For the computational resource provider, in many cases jobs can be swapped out of processors or the job queue manipulated in an automated way to better maintain system throughput within reserves and with a minimum of customer dissatisfaction.

We can now discuss the specific features of a swing pricing strategy for the resource provider. The swing pricing strategy is key for having a profitable swing option service. The swing *strike* price is the price at which the swing option is offered by the resource provider and is a function of:

$strike(current\_time, start, end, CPU_{min}, CPU_{max})$

with the following qualitative behavior:

strike is an increasing function of: *current_time* (incentivizes early commitment)
strike is a decreasing function of: *end – start* (incentivizes longer commitment)
strike is a decreasing function of: CPU_{max} (incentivizes larger commitment)
strike is an increasing function of: $CPU_{max} - CPU_{min}$ (incentivizes surer commitment)

The penalty function, *penalty*, (paid by the customer to the resource provider) is a function of: CPU_{min}, CPU_{max}, and CPU_{actual} with the following behavior:

if $CPU_{actual} > CPU_{max}$ then penalize proportional to the time and amount of overage;

if $CPU_{actual} < CPU_{min}$ then the user must pay proportional to CPU_{min} of the time spent underutilized.

The ability of the customer to resell unwanted swing options at a price of $resell = r*strike$ is set below *strike*, i..e, $r < 1$, so as to incentivize accurate swing purchases and reduce churning. The discounted rebate to the customer is one mechanism that the resource provider uses to resell unwanted resources rather than letting them go unused.

For some time periods there will be cases where the overbooking of resources leads to a situation with a shortfall of resources. In these cases there must a tie-breaking criteria to see who actually gets to use the resource and who does not. For any given period of time where the resources are overbooked, the resources are granted in the following order of preference with the resources going to the winner of the first criteria that is not a tie between users with a random selection as a default when all the other criteria are tied:

1) highest probability of not canceling reservations (incentivizes customer predictability);
2) earliest reservation (incentivizes early commitment);
3) largest request (incentivizes larger commitments);
4) longest request (incentivizes longer commitments);
5) random selection.

For example, the users are first ranked by their historical probability of not canceling reservations and the resources allocated until they are used up. If there is a tie between users and there are still resources remaining, then the users are ranked by the time of their reservation with the earliest reservation receiving the resources. If the reservations were made at the same time then the tie-breaking criteria goes to the next level, and so on until all the resources are committed or a random tie-breaker is used.

Although the above example and those that follow are given for CPUs, other computational resources, such as disk storage and network bandwidth, can be used. Also, bundles of such computational resources (computons) could be used as well

We now construct the actual swing pricing algorithm. The main idea behind the pricing model is to use dimensionless quantities that can be used as scale factors. Thus, the actual swing price will be the derived scale factor multiplied by the cost per unit of resource. This dimensionless approach returns a profit margin rather than an outright price. Another advantage of a dimensionless approach is that can be used for any resource allocation problem as long as the basic cost of the underlying resource is known.

To begin with, let n_{peak} be the peak number of CPUs needed by a user or users and let n_{avg} be the average number of CPUs expected to be used. The customer would like to pay for only the CPUs used plus the convenience of not having to own and operate a data center. The swing option service contract is designed to fulfill the customers needs for computational convenience and the resource provider's need for a profit. Because the resources are available on a short term notice the resource provider can charge a premium over the long-term rate cost of purchasing the resources, yet the customer can still save money in the long run because there is a much lower rate of wasted resources with the resource provider in a swing option service contract.

The load factor is a term commonly used in the power industry and the analogous quantity for the compute resource market is,

$$load\_factor = f = n_{avg} / n_{peak} \quad . \tag{1}$$

In the special case that the historical usage is found to be completely constant then $f = 1$ and so there is no profit to the swing option service because the customer could just as easily have bought the resources itself rather than utilize the swing option service. However, it's extremely unlikely that this will be the case over the long term. In fact, the heavy-tailed distributions associated with computer use[2, 10, 12] mentioned earlier imply that f can be much less than 1 so that the peaks are an order of magnitude more than the typical usage.

The resource provider provides for each customer's peak usage through resource sharing with other customers, for example through the use of statistical multiplexing[3, 11, 13] or some other means. The resource sharing may involve checkpoint/restart of jobs as well as utilizing unused cycles on reserved machines. Assuming that the peak customer usages are not fully correlated, a feature that can be controlled by an appropriate admission policy, the resource provider only has to provide a fraction of the summed peak usages of all its customers. In fact, the ability to effectively share resources through uncorrelated usage is one of the competitive advantages of IT outsourcing over direct ownership by a consumer. The actual amount of resource needed by the resource provider is thus given by the bounds of single largest peak customer demand for completely anti-correlated demands and by the sum of all the peak demands in the case of completely correlated demands.

Now we turn to calculating swing option prices in the context of our swing option service model. In general, in a swing option service the revenue consists of 4 components:

$$revenue = avg.usage + swing\_options + penalties + churn. \tag{2}$$

where *penalties* are assessed by the resource provider to the customer for going above its swing range and *churn* is the money from the customer making subsequent swing trades. (Note that at this point we are neglecting the loss of revenue due to class of service violations.) For example, if the customer expects a different level of use than that covered by its original swings, then it can buy or sell new swing contracts to make the swing

contract range more financially efficient. In the case where the customer sells back a swing to the resource provider, these resources can be resold by the resource provider to another customer thereby allowing the resource provider to receive revenue multiple times on the rights to the same resources.

One strategy that the resource provider can use is to have the average usage account for the resource provider's operating costs and the swing options for the profits, while the penalties, and churn to balance out the class of service violations. All these quantities can be calculated using historical data.

As part of our dimensionless pricing model we tie the profit margin of the swing option service to the swing usage, ignoring the penalties and churn for the time being. Thus, the overall revenue to the resource provider can be written in terms of the peak and average usage and the swing costs, namely,

$$revenue = cost + profit = c\, n_{peak} (1 + m) = c\, n_{avg} + swing \qquad (3)$$

where c is the cost per unit of computation resource to the resource provider for providing the service and m is the desired fractional profit margin to the resource provider. This equation allows the resource provider to determine the price to charge the customer for swing contracts for a desired profit margin. Namely,

$$swing = c\, n_{peak} ((1\text{-}f) + m). \qquad (4)$$

What remains to be determined is the cost of each swing option to the customer. In a worst (best)-case scenario for the resource provider(customer), the customer's historical data provides perfect predictive power and the customer exercises exactly the minimum number of swing options that it needs to cover its usage. In other words, the user does not have swings in effect that are too high or too low. Practically speaking there will be a granularity in the swing CPU ranges so that there will be some excess capacity that the user is paying for but not using. However, these are likely to be only a few percent which works in the favor of the resource provider since these could in principal be shared with another user. We now incorporate the swing range granularity into our pricing model.

Let

$$\Delta_{swing} = \text{range of CPUs covered by a swing.} \qquad (5)$$

and

$$CPU_{range} = CPU_{max} - CPU_{min} \qquad (6)$$

be the range of CPUs covered by the swing contract. Then the number of swing contracts, s, exercised by a user during a time period (e.g., one day or one week) is given by:

$$s = \langle CPU_{range} \rangle / \Delta_{swing} \qquad (7)$$

where the average is over the user or facility profile usage data. Finally, we have the cost of buying a swing contract, basically the price for the right to exercise the swing, e = *strike*. The idea is to set the cost of the swing such that it corresponds to the pre-defined level of profit of the swing option in (4).

$$swing = s\ e = c\ n_{peak}\ (\ (\ 1\text{-}f\ ) + m\ ), \tag{8}$$

or solving for e,

$$e = c\ n_{peak}\ (\ (\ 1\text{-}f\ ) + m\ )\ /\ s \tag{9}$$

or,

$$e = c\ n_{peak}\ (\ (\ 1\text{-}f\ ) + m\ )\Delta_{swing} / <CPU_{range}> \tag{10}$$

which gives us the cost to the customer for each swing contract.

Now that we have the cost of the swings we can add in the other terms of the profit equation, namely, the penalty and churn profits gives us,

$$profit = swing + c\ \beta\ \Delta_{CPU} + churn \tag{11}$$

where β (>1) is the penalty scale factor for going out of the swing bounds and Δ_{CPU} is the cumulative amount (in cycles, i.e., CPU-hr) that the customer is out of bounds of the swing contract divided by T, the total time of the contract.

To complete the profit picture we need to take into account users buying and selling additional swing contracts, i.e. the churn. A churn is one of four possible actions:

1) a swing cancellation on the high side (a lower upper bound),
2) a swing cancellation on the low side side (a higher lower bound) ,
3) a new swing on the high side (a higher upper bound),
4) a new swing on the low side (a lower lower bound).

The change in the profit to the resource provider assuming a time dependency in the canceling and refund rate, but no change in actual usage:

$$churn = Presale(dt)*strike(dt) - Pcancel(dt)*\ r(dt)*strike(dt) \tag{12}$$

churn = change in revenue to the resource provider because of a change swing contracts
dt = exercise date – cancellation date
Pcancel = probability of a swing being cancelled on the high side or low side.
Presale = probability that a swing is resold after it has been cancelled
r = refund rate as a fraction of the original swing cost for a cancelled swing

Once the swing option service has been in operation for some time there will be historical data on cancellations and resales so the resource provider can decide what the refund schedule should be to preserve the proper level of profit margin. Because $r < 1$, the resource provider always comes out ahead financially on any swing churning. However, it may still be advantageous to the customer to cancel because the customer has decided it needs the usage anyway and therefore there is no need for the downside protection and effectively recoups a refund which is some fraction of the original swing cost. Also, canceling a swing contract on the high side reduces the customers costs while recouping some of the cost of the swing.

We should also point out that swing trading may not always be advantageous to the customer and could lead to higher minimum costs because of usage below a higher *swingLo* (lower bound of a swing range) or due to more penalties because of usage above a lower *swingHi* (upper bound of a swing range) if the usage turns out differently from what the customer expects.

To incentivize more consistent usage by customers the resource provider could also offer a "low fluctuation" discount so that the proportional usage term would be modified to be:

$$avg.usage = c\,(\,1 + \delta\,(\,1\text{-}f\,)\,) \tag{13}$$

where $\delta > 1$ is the discount for higher load factor (i.e., $f \to 1$).

There can also be a term to reward customers for committing early to a contract, for example:

$$1 - b(\,1 - e^{-\gamma \Delta t}\,) \tag{14}$$

where $0 < b < 1$ is the maximum discount for an early commitment, $\gamma > 0$ is the early discount decay rate, and Δt is the time between when the swing was purchased and when it is exercised (used).

Thus, if the actual usage follows the historical data then the resource provider will receive the amount of profit defined in (11). If the usage data has less fluctuations than that during an actual contract then the swing profit will be the same. If the fluctuations are higher than the historical data then the penalty term in (11) will kick in while the resource provider still retains the same profit from exercised swing options. Any rebates the customer receives for returned swings will tend to reduce the overall profit.

What we have done in constructing the swing price model so far is in essence to define the swing price in order to fit the usage profile and to achieve an average level of profit to the resource provider. In practice the gross profit margin is now given by:

$$m_{gross} = (\,s\,e + \beta \Delta_{CPU}\,c + e(\,Presale - rPcancel\,)\,) / (\,c\,n_{peak}\,) - 1 + f \tag{15}$$

In order to avoid penalties, the conservative user may wish to exercise more swing options which will increase the profit to the resource provider from the swings. Thus, in reality, the user will either overestimate the number of swings and incur an extra swing cost or underestimate the number of swings needed and incur extra penalties. Either way, the resource provider benefits because of the existence and unpredictability of the fluctuations.

Fig. 3 shows an example of the historical profile data, the swings, the actual average usage, and the actual lower and upper usages adapted from actual server trace usage data. The first week's times correspond to the profiled data that are used to determine the swing ranges. The actual usage takes place for second week's times that correspond to the profiled days, i.e, profile day "Mon" is the profiled usage for actual day "Mon", and so on. The error bars correspond to the usage and the gray areas to the swings for that day. In this case the customer has chosen swings that are minimally covering the profiled usage. The overall profit (calculated from the positive days) is made from the actual usage (solid line), the swings (gray area), and the over usage penalties (dashed lines outside of gray areas). Thus, the user pays for actual average usage and the protection of the swings plus any penalties. Note that on Monday and Tuesday the user pays a penalty for going out of the bounds of his swing contract.

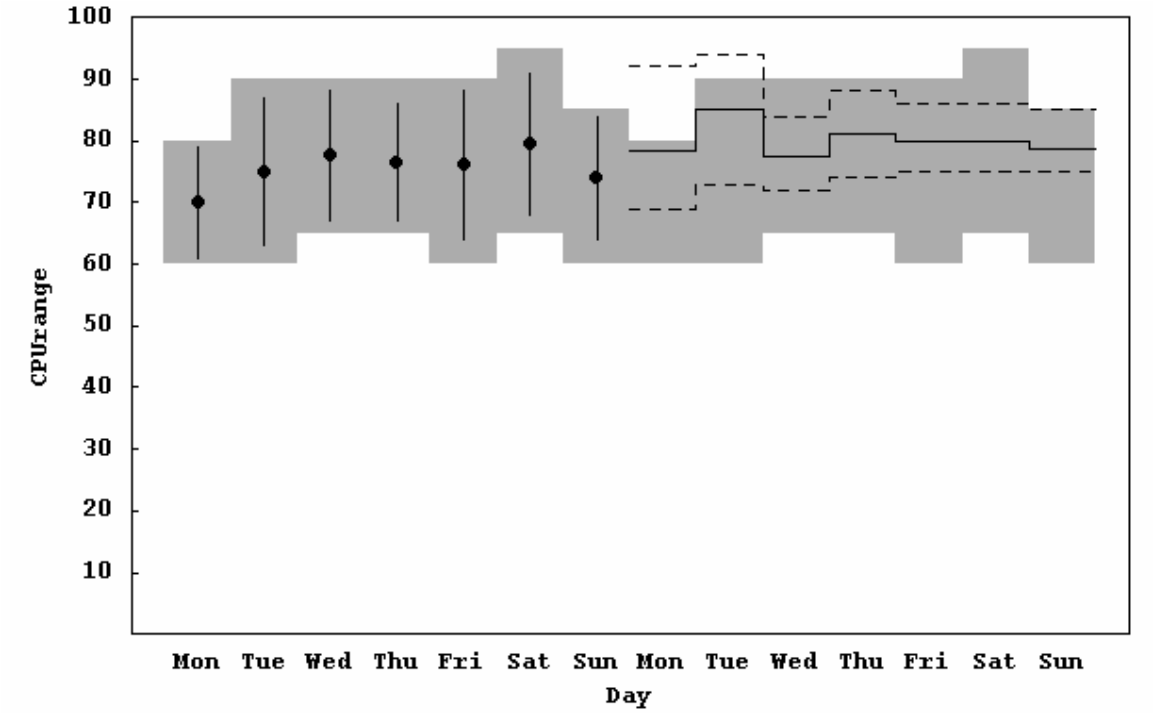

Fig. 3. Profile data (error bars on first 7 days), swings(gray area), actual average usage(solid line on second 7 days), upper and lower ranges of actual usage(dashed lines on second 7 days).

Analysis

In this section we simulate the swing option service model using CPU historical data. Using a cross-validation of historical traces, the swing option service model described above can be tested under various parameter values and the distribution of profits obtained. These distributions can be used by the resource provider and customer to better understand the risk involved in a particular swing pricing and allocation strategy. For this analysis we have assumed that there are class of service violations which would decrease the profit to the resource provider, as explained in more detail below.

We make the following assumptions about the inputs and parameters to the model as shown in Table 1. We assume without loss of generality that the cost rate scale $c = 1$. and that that there are no time-varying quantities. The probabilities of canceling and rebuying a *swingLo* or *swingHi* are uniformly random over time at a rate of 10%/day. We also assume a Normally distributed class of service violation with average $m_{gross}/10$ and standard deviation of $m_{gross}/20$ so that:

$$m_{net} = m_{gross} - service.violation = m_{gross} - N(\, 0.1\, m_{gross},\, 0.05\, m_{gross}\,) \qquad (16)$$

where $N(\, \mu,\, \sigma\,)$ is a random normally distributed number with mean μ and standard deviation σ.

Table 1. Swing Option Service Model Parameters and Values

Parameter	Determined By	Value
n_{avg}	historical usage	80
n_{peak}	historical usage	96
$\langle CPU_{range} \rangle$	historical usage	25
β	swing option service	2
Δ_{swing}	swing option service	5
r	swing option service	0.5
m	swing option service	0.1
s	calculated	5
e	calculated	6
f	calculated	0.8
Pcancel	real-time	0.10
Presale	real-time	0.10
Δ_{CPU}	real-time	5
m_{gross}	real-time	0.2
m_{net}	real-time	0.1

Fig. 4 below shows a simulated daily profit margin distribution based on customer's past usage from an historical log of 49 days. The profile data from which the swing prices were calculated consisted of 21 days and the test data against which the prices were calculated consisted of 14 days. Seven day segments were randomly chosen from the log to constitute the profile and test data and to preserve the day-to-day correlations. A different combinations of the profile and test data contributed to each of the 100 runs shown in the plot which is normalized to give a probability plot.

The desired swing option service profit (set to 10% by the resource provider) is exceeded when there are swing contract violations and churning. The values for *swingLo* and *swingHi* were set such that they were as close a possible to always span the historical profile data for a given run. Thus, the swings were chosen to be minimally covering of the profile data. The figure shows that the most likely profit is 10% as expected, but that there is also a small probability for less than 10% profit either because the actual usage was less than the historical average usage, or there were some class of service violations. Of further note is the existence of the long tail on the high end of the distribution, which indicates the presence of penalties to the user and additional sources of revenue for the resource provider.

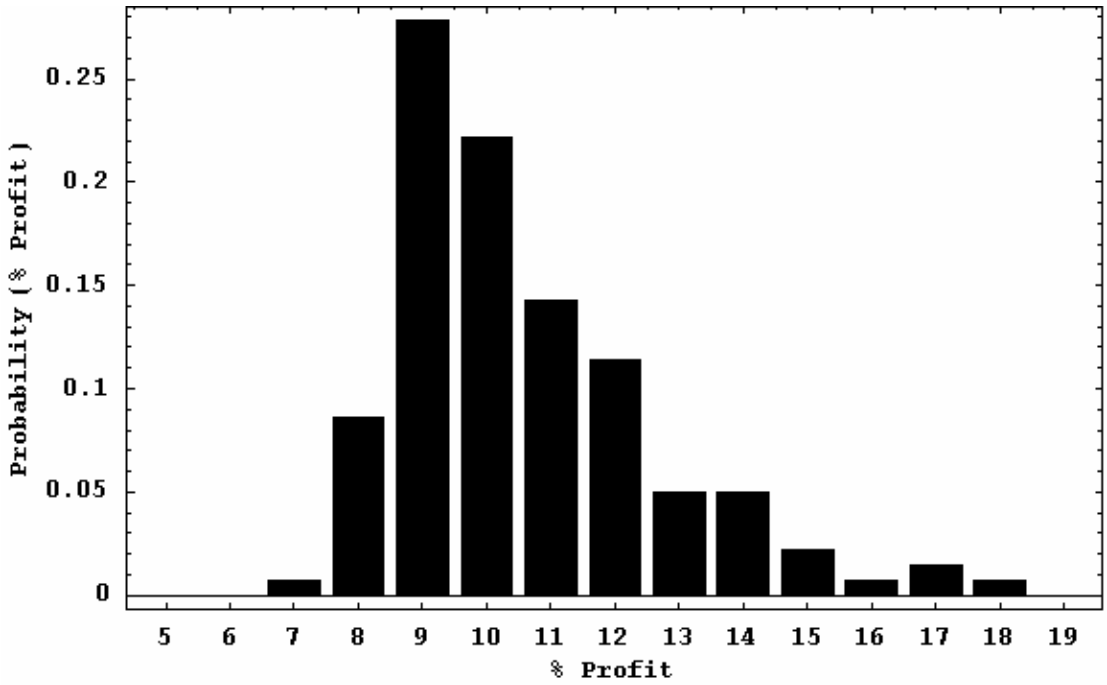

Fig. 4. Profit margin distribution for many simulated runs.

To better understand and control the risk entailed with the distribution of the profit margins, the parameters in the pricing model can be tuned while using a methodology such as "Price-at-Risk"[9].

Fig. 5 shows the breakdown of profit contributions as a function of different runs simulating a distribution over time when swing trading is taken into account. The original

swings were chosen to be the minimum range necessary to cover the profiled usage. The swing trading is modeled by a 10% probability that *swingHi* or *swingLo* will be raised or lowered for that time period by 1 swing contract. This amounts to adding random noise to the swing ranges so that we can simulate its effect on the profit margin.

The average profit is 10.9%, again close to the desired level of 10%; the penalty profit is 1.3%; and the churn profit is 0.4%; the service violations contribute -1.5%, for a total profit of m_{net} =11.1% which is very close to the desired figure of m_{gross} = 10%. Of course with different parameter settings we will see a different distribution of profit but the combination of profit from actual usage and swings should be quite close to the figure selected by the resource provider. Profits above the desired level may be due to excessive penalties or swing trading, or a combination of both.

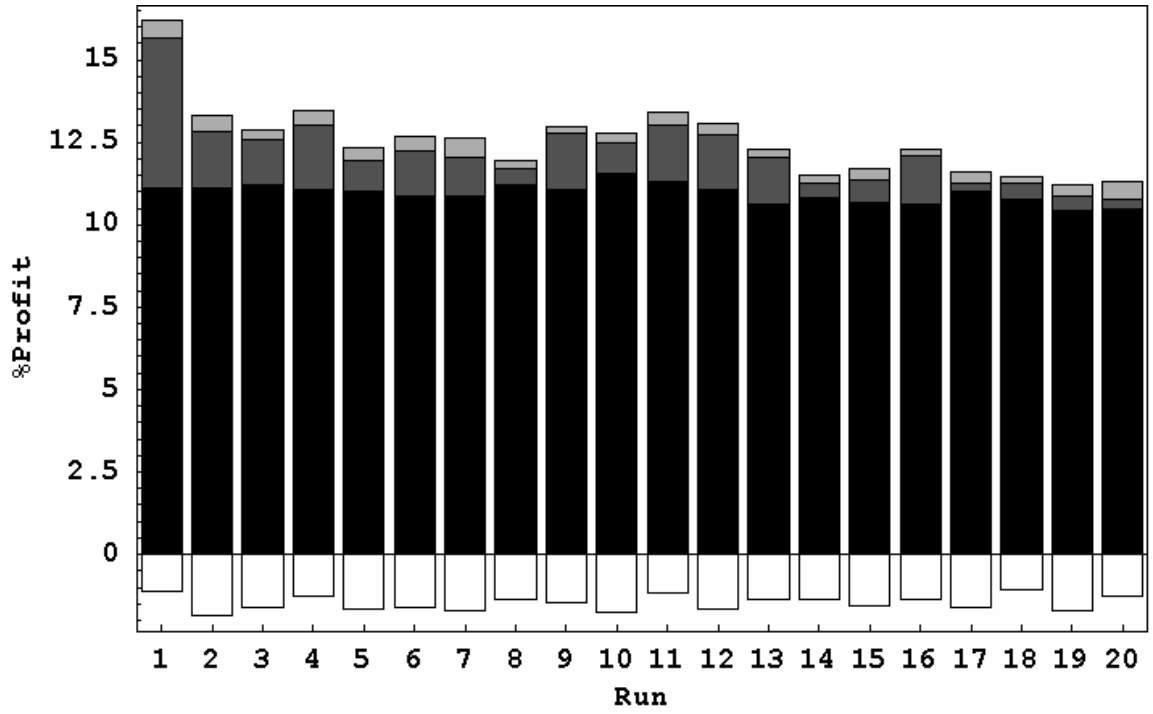

Fig. 5. Fractional profit margin distribution by run for swings(black), penalties(dark gray), swing trading(light gray), service violations(white).

We can also model the effects of different swing strategies by the customer. Specifically, we consider a more conservative swing strategy in which there is one extra swing above and below the minimum range. A more aggressive strategy is modeled by using one less swing contract than the minimum range. These results are shown in Fig. 6. The conservative swing strategy led to a resource provider profit margin of m_{net} =10.0% and the aggressive swing strategy led to a resource provider profit margin of m_{net} = 13.9%. The results show that an aggressive swing strategy on average hurts the customer because of excessive penalty charges. On the other hand, a more conservative swing strategy lowered the profit margin of the resource provider from 11.1% to 10.0% which means that the customer saved 1.1%.

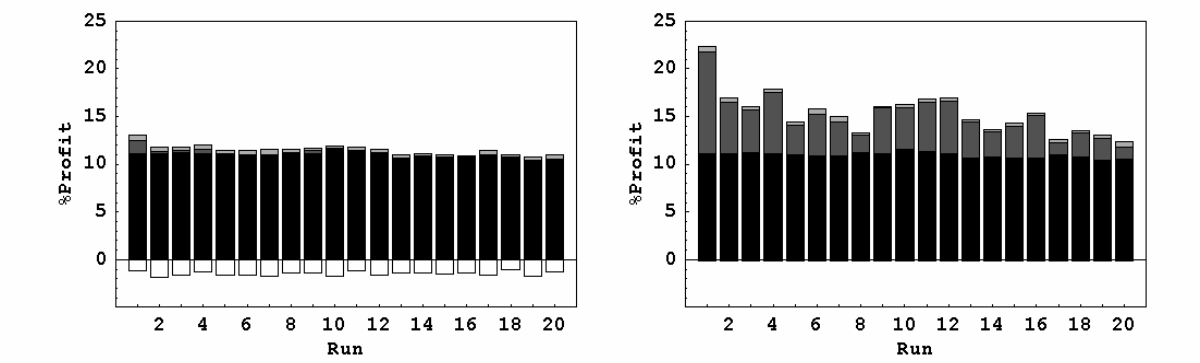

Fig. 6. Comparison of fractional profit as a function of run for conservative (left) and aggressive (right) swing strategies.

What is not shown in the plot however, is the fact that the resource provider would benefit from more predictable usage and thus be better able to accommodate unexpected peaks from its other customers through resource sharing.

Up to this point we have assumed that all the historical data has been aggregated in such a way that the swing costs and other calculated quantities are based on the overall resource provider's customer portfolio. Over time, users with high swing costs, penalties or churn may decide to leave a particular resource provider because they are effectively carrying the resource provider's profit burden disproportionately. By the same token, in a competitive environment with multiple resource providers, customers with very predictable usage will have lower costs and there may be a migration of such users to a specific resource provider, leading to an overall slightly lower profit margin. To deal with this situation the resource provider may have to institute a customer admissions policy that effectively selects for a portfolio of customers that includes a certain percentage of higher profit customers, assuming they are available.

If the resource provider chose to price each customer individually rather than in aggregate, then it may be possible for the customer to game the system for a time. For example, suppose a customer provides an initially noisy set of profile data. This will cause the resource provider to set the cost of individual swings (e) to be fairly low for this customer and then make up the profit with volume. However, the scheming customer could deliberately choose a small swing range knowing that its actual usage data is quite smooth. In this case the small swing range will not hurt the scheming customer because it knows its true usage and it will pay less for the swings and have a lower cost overall. However, this trick only works one time because once the customer begins using the system the resource provider has an historical record and can use this record after the first billing period so that the customer's arbitrage advantage disappears and the customer's profit margin reverts to that of the resource provider's target.

Over time none of the customer's historical records will be stationary so the swing option service will be required to periodically update its prices for swings, penalties, churns, and service violations. The same formalism described above can be used in this time-changing environment. For example, a sliding window of historical usage can be used to

track variations. Also, a time-decayed usage model in which the effects of older data are exponentially decayed could be used. Another alternative to keep track of time varying usage would be to use some sort of time-series analysis in which closely tracking records from the past are used as predictors of future usage.

Discussion

We have presented a novel means for pricing computational resources using swing options. In the swing option service model presented here the resource provider acted as a seller of rights to computational resources as well as buyer of unused ones. We note that in a mature market for IT resources customers could sell the swing options among each other directly in a true market.

A swing pricing model was then derived to obtain a profit margin selected by the resource provider. A cross-validation simulation testing of historical customer data helped to generate the appropriate prices and risks involved with a particular pricing and swing strategy. We then showed that the swing option service framework provides a predictable profit margin to the resource provider under a variety of customer swing strategies. The swing option service also provides the customer with an economical means for accommodating peak demands.

In the early days of electricity generation large firms had their own in-house power plants. These were eventually replaced by the grid of power utilities we have today. Whether computational resources will follow this same pattern remains to be seen, but insofar as peak demand for IT is a problem, our swing option mechanism solves it in ways reminiscent of other utilities.

Acknowledgements

We thank Julie Symons for providing the server data.

References

[1] Angelo Barbieri, Mark B. Gurman "Putting a price on swings" Energy and Power Risk Management, Oct. (1996).

[2] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger, "Long-Range Dependence in Variable-Bit-Rate Video Traffic", *IEEE Transactions on Communications*, vol. 43, no. 2/3/4 Feb./Mar./Apr. 1995, pp. 1566-1579.

[3] R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn "Statistical service assurances for traffic scheduling algorithms" IEEE Journal on Selected Areas in Communications. Special Issue on Internet QoS, Vol. 18, No. 12, pp.2651-2664, December (2000).

[4] S. D. Gribble, G. S. Manku, D. S. Roselli, E. A. Brewer, T. J. Gibson, and E. L. Miller, "Self-Similarity in File Systems", *Measurement and Modeling of Computer Systems*, 1998, pp. 141-150.

[5] B. A. Huberman and R. Lukose, Social Dilemmas and Internet Congestion, *Science*, Vol. 277, 535-537, (1997).

[6] Patrick Jaillet, Ehud I. Ronn, and Stathis Tompaidis "Valuation of Commodity-Based Swing Options" Management Science" Vol. 50, No. 7, pp.909-921 (2004).

[7] Jussi Keppo "Pricing of Electricity Swing Options" Journal of Derivatives, Vol. 11, pp.26-43 (2004).

[8] Ali Lari-Lavassani, Mohamadreza Simchi, and Antony Ware "A Discrete Valuation of Swing Options" Canadian Applied Mathematical Quarterly, Vol. 9, pp.35-73 (2001).

[9] G. A. Paleologo "Price-at-Risk: A methodology for pricing utility computing services" IBM Systems Journal, Vol. 43, No. 1, pp.20-31 (2004).

[10] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling", *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, 1995, pp. 226-244.

[11] Jerry Rolia, Xiaoyun Zhu, Martin Arlitt, and Artur Andrzejak "Statistical Service Assurances for Applications in Utility Grid Environments" Hewlett-Packard technical report HPL-2002-155 (2002).

[12] J. Voldman, B.B. Mandelbrot, L.W. Hoevel, J. Knight, and P. Rosenfeld, "Fractal Nature of Software-Cache Interaction", *IBM Journal of Research and Development*, vol. 27, no.6, Nov. 1981, pp. 164-170.

[13] Z. Zhang, D. Towsley, and J. Kurose “Statistical analysis of generalized processor sharing scheduling discipline” IEEE Journal on Selected Areas in Communication, Vol. 13, No. 6, pp. 1071-1080 (1995).