Journal of Cloud Computing
a SpringerOpen Journal

RESEARCH Open Access

A financial brokerage model for cloud computing

Owen Rogers* and Dave Cliff

Abstract

One of the major benefits of cloud computing is the ability for users to access resources on a pay-as-you go basis, thereby potentially reducing their costs and enabling them to scale applications rapidly. However, this approach does not necessarily benefit the provider. Providers have the responsibility of ensuring that they have the physical infrastructure to meet their users' demand and that their performance meets agreed service level agreements. Without an accurate view of future demand, planning for variable costs such as staff, replacement servers or coolers, and electricity supplies, can all be very difficult, and optimising the distribution of virtual machines presents a major challenge.

Here, we explore an extension of an approach first proposed in a theoretical study by Wu, Zhang, & Huberman which we refer to as the WZH model. The WZH model utilises a third-party intermediary, *the Coordinator*, who uses a variety of cloud assets to deliver resources to clients at a reduced price, while making a profit and assisting the provider(s) in resource forecasting. The Coordinator acts as a broker.

Users purchase resources in advance from the broker using a form of financial derivative contract called an *option*. The broker uses the uptake of these options contracts to decide if it should invest in buying resource access for an extended period; the resources can then subsequently be provided to clients who demand it.

We implement an extension of the WZH model in an agent-based simulation, using asset classes and price-levels directly modelled on currently available real-world data from markets relevant to cloud computing, for both service-providers provisioning and customers' demand patterns. We show that the broker profits in all market conditions simulated, and can increase her profit by up to 36% by considering past performance when deciding to invest in reserved instances. Furthermore, we show that the broker can increase profits by up to 33% by investing in 36-month instances over 12-month. By considering past performance and investing in longer term reserved instances, the broker can increase her profit by up to 44% for the same market conditions.

Keywords: Utility computing, Brokerage, Market-orientated computing, Cloud federation, Financial derivatives, Options, Markets

Background

It is generally accepted that on-demand pricing for cloud computing resources offers benefits to consumers [1,2]. They have full operational control of costs by being able to start and stop resources on demand, and they do not have to engage in the capital expenditure of building their own infrastructure, hiring IT systems support staff, or investing in maintenance of physical machinery. Furthermore, if different providers of cloud computing resources could interoperate, a federated cloud would in principle allow units of cloud-computing resources to be traded as commodities on an open marketplace, thereby allowing for the price of resources to smoothly vary while the market mechanism enables matching of consumer demand to provider supply [3].

But would open on-demand trading of cloud-computing resources with variable pricing actually benefit a cloud service provider? Purchasing goods and services in advance of delivery allows the provider to plan and prepare for the future. How can the provider ensure they are maximising profit and reducing cost if they must provide resources without knowledge of future demand?

Such knowledge offers benefits to providers in multiple ways. The provider must ensure that there is a physical capability for demanded resources, but when consumers engage in on-demand pricing the providers must predict what usage is required, and ensure that the infrastructure is there.

* Correspondence: csorr@bristol.ac.uk
Department of Computer Science, University of Bristol, Merchant Venturers Building, Bristol, UK

When do they invest in new infrastructure? As manufacturing processes improve and economies of scale increase, the real cost of infrastructure decreases while its technological capability increases. So it is better for the provider to wait for as long as possible before investing in additional capability so they get the best value for money [4]. But how do they know when is the best time?

As for any business, the provider has variable costs, which are related to the output being generated. How can these be planned? Running a thousand servers instead of just one will require many more support staff and engineers: if too few are employed, failures can mount up and cause service outages or downtime. This downtime is not only costly in terms of SLA (Service Level Agreement) penalty fees and refunds, but also in terms of reputational risk; yet if too many are employed, expenditure is wasted on staff that are surplus to requirements [5].

Electrical power consumption is another key variable cost [6]. If the provider has a good view of future energy demand, they may gain a discount by forecasting their requirements and supplying this to their energy supplier.

It is also difficult to schedule customer instances to servers efficiently, without advance notification of usage [7,8]. Most cloud computing providers require no duration of execution to be stipulated when the instance is started. So efficiently scheduling instances such that the number of powered servers is reduced is a very tough challenge.

A simple method of capacity planning is simply to track trends in demand, and ensure there is enough capacity (with an additional margin) to meet the maximum previously experienced. However, enterprise cloud computing resources have only been available to consumers for a relatively short amount of time, and it is likely that demand information is too variable and incomplete to build a reliable forecast without the considering users own predictions.

Currently, most providers offer fixed price models for immediate delivery. Market-leader Amazon Web Services (AWS) also offer a spot price model, where the price varies depending on current supply and demand, and winning bidders gain access to the resource until the spot price moves above their maximum bid. These schemes do not aid in capacity planning.

There are a number of alternative schemes that could be used. Users could instead purchase *derivatives contracts*, which give the user contractual rights to a resource at some specified later date [9]. *Futures contracts* are a type of derivative that give buyers guaranteed access to the resource in advance of when it is delivered, usually in return for an upfront payment on signing the contract followed by a final settlement when the resource is delivered: the user is *obliged* to take ownership of the resource on the delivery date that the contract specified. Alternatively, *options contracts* give buyers the legal *right* (but not an *obligation*) to purchase a resource for an agreed strike-price on (or sometimes on-or-before) some later delivery date [10]. Derivatives such as futures and options are commonly used in a number of commodity markets for industrial inputs such as wheat, oil, natural gas, and metals, and various types of *financial* derivatives have also in recent decades become notoriously commonplace in the global financial markets for equities, currencies, and bonds.

In a limited sense, current commercial cloud-computing vendors are already offering both spot and forward contracts. Amazon Web Services (AWS) is a major supplier of cloud infrastructure services, and in the AWS terminology a remotely-hosted virtual machine and its associated software is known as an Amazon Machine Instance, often abbreviated to AMI or simply referred to as an *instance*. AWS offer two types of fixed price asset classes: *On-Demand Instances* and *Reserved Instances*. AWS on-demand instances are fixed-price resources that are delivered as soon as they are purchased, and the purchase price is set by AWS. AWS reserved instances allow users to pay a reduced price per unit time for a resource, by paying an upfront fee. This fee guarantees them a reduced on-demand charge for a specified period, either 12 or 36 months.

The key question that we explore in this paper is this: is it possible for a cloud-computing services broker to use these derivative contracts in combination to reliably provide cheaper resources to the consumer and also aid in predicting future usage?

Cloud brokerage

A broker makes a profit by matching buyer's demands with seller's supplies: the broker uses a variety of methods to achieve a best price between these parties, and typically makes a profit either by taking a commission fee from any completed deal, or by varying the broker's *spread*, or some combination of fees and spread. The spread is the difference between the price at which a broker buys from sellers and the price at which it sells to buyers.

In a *commodity* market where goods cannot be differentiated between suppliers on any basis other than price, all that matters to the buyers is that they pay a price they are comfortable with, and all that matters to the sellers is that they receive a price they are also comfortable with [11]; typically the broker's fee and/or spread are tolerated by the buyers and sellers because the broker acts as a provider of *liquidity*: the buyers and sellers do not have to spend time finding potential counterparties to their transactions and then negotiating

with them, and they do not have to worry about there ever being no counterparties to trade with.

The broker aims to satisfy these parties' requirements while ensuring that he or she makes a profit. In this work, we focus on a simple way of achieving this by purchasing advance rights or obligations on resources, via derivative contracts. When the derivatives contract matures and the corresponding resources are delivered, the broker can then sell the resources to clients who need to use them: in this context, the broker is using the derivatives as a mechanism for hedging risk in the uncertainty over future demand and supply.

The broker's skill lies in forecasting when to purchase the resources in advance and when to simply provide their clients with resources purchased directly from the spot market or from the broker's own private stock of resources.

A simple broker model consists of two stages. In the first stage, the broker plans what to buy. The broker will make a forecast of demand, determine what and when to buy and then will make any advanced purchases from the provider.

In the second stage, clients approach the broker for a resource. If the broker has previously purchased a resource, they can sell it to the client for a profit if the cost is less than the client wants to pay. If not, the broker must purchase an on-demand instance and provide it to the client.

But how can the broker effectively predict usage? Wu, Zhang, and Huberman suggested a two-period model (which we refer to hereafter as the WZH model) for resource reservation in which in the first period the user knows her probability of using the resource in the second period, and purchases a reservation whose price depends on that probability [12].

Consider N users who live for two discrete periods. Each user can purchase a unit of resource from a service provider to use in the second period, either at a discounted rate of 1 in Period 1, or at higher price C, where $C > 1$, in Period 2. In Period 1, each user only knows the probability that they will need the resource in Period 2–it is not known for certain until the next period.

A third agent, the Coordinator, is introduced who makes a profit by aggregating the users' probabilities and absorbing risk through a two period game described below:

1. Period 1: Each user i submits to the Coordinator a probability, q_i, which does not have to be the real probability, p_i, that they will require a unit of resource in Period 2.
2. Period 1: The Coordinator reserves $q_i n_i$ units of resource from the resource provider at the discount price for use in Period 2, where n_i is the number of units of resource required by each user. For simplicity in this simulation, $n_i = 1$ for all users.
3. Period 2: The Coordinator delivers the reserved resources to users who claim them. If the amount reserved by the Coordinator is not enough to cover the demand, the Coordinator purchases more from the resource provider at the higher unit price C.
4. Period 2: User i pays:
   $f(q_i)$ if resource is required
   $g(q_i)$ if resource is not required

The contract can be regarded as an option if $g(q_i)$ is paid in Period 1 (i.e. as a premium), and $f(q_i)$-$g(q_i)$ is paid in Period 2 (i.e. as a price) should the resource be required. In Period 1, the resource is reserved, but the user is not under any obligation to purchase.

Wu *et al.* showed that if the following conditions could be met, the Coordinator would make a profit:

- Condition A: Each user prefers to use the service provided by the Coordinator, rather than to deal with the resource provider (for example, by achieving cost-reduction)
- Condition B: The Coordinator can make a profit by providing the service.

The following truth-telling conditions are not completely necessary, but are useful, for conditions A and B to hold:

- Condition T1 (truth-telling): Each user submits his true probability in Period 1 so that he expects to pay the lowest amount later.
- Condition T2 (truth-telling): When a user does not need a resource in Period 2, it is reported to the Coordinator in the same period.

The following specific case was proved to meet these conditions, where k, a constant chosen to alter the price paid by the customer, is set to 1.5 and C is set to 2:

$$g(p_i) = \frac{kp_i^2}{2}$$

$$f(p_i) = 1 + \frac{k}{2} - kp_i + \frac{kp_i^2}{2}$$

In previous work [13], we validated Wu *et al.*'s claims though simulation experiments and showed that a simple evolutionary optimization process operating on an initially maximally dishonest pool of users results in the

pool of users becoming more honest over time when interacting with the WZH system.

The WZH model can be used to forecast demand as it encourages users to submit an honest estimate of future usage by rewarding them with a reduction in cost over time, compared to buying direct from the provider at the higher rate *C*.

In our model, we refer to the Coordinator as the broker, as the Coordinator performs a role traditionally performed by a financial broker. In the first period, the broker asks each user i to submit a probability p_i in advance that they will use a resource, and pays a premium $g(q_i)$.

The broker sums these probabilities, which is the forecast of how many resources will be required in the next period.

Once a forecast has been made, the broker must make a decision on whether to hedge the risk and invest in a reserved instance, or to wait until the next period and buy an on-demand instance. The broker does this by comparing the performance of a reserved instance over the past instance term and comparing it to the amount of capacity it currently has available through reserved instances over the same period in the future. It considers whether another purchased reserved instance will be suitably utilised such that it gives better returns than delaying the purchase and buying on-demand later. The broker does this by using a variable called the *threshold*–if the resource is likely to be used more than the threshold it invests; if not, it does not. In the rest of this paper we denote the threshold by θ.

The *margin resource utilisation* (i.e. the likely utilisation of an additional reserved instance) is calculated as follows:

Previous demand profile $\boldsymbol{A} = [d_{t-36}.....d_t]$

Future capacity profile $\boldsymbol{B} = [c_t.....c_{t+36}]$

Deficit profile $\mathbf{C} = \boldsymbol{A} - \boldsymbol{B}$

For each resource required, the

Marginal Resource Utilisation (MRU) is the ratio of items in $C > 0$, and the key decision is then:

If $MRU > \theta$: Buy reserved instance and therefore increase current capacity profile;

If $MRU <= \theta$: Do not invest, "wait and see".

In the second period, users choose if they should execute their right to use the resource. If they wish to, they pay $f(q_i)$-$g(q_i)$ to use the resource.

The broker gives the user access to one of their previously purchased reserved instances for the full calendar month. If the broker has not purchased enough, it buys an on-demand resource from the provider. Potentially, the broker could also offer fractions of months for a reduced price, but in our model here, we only offer full-month resources.

Simulation

We now present results from our computer simulation experiments that explore the performance of the WZH system. The simulation, which was programmed in Python [14], was developed to explore the dynamics of the WZH model in situations where supply and demand fluctuate in patterns directly inspired by historical records of economic activity in a number of distinct real-world market sectors. For each sector, simulations were implemented with a pool of 1000 user agents submitting probabilities, and each simulation was run 100 times with systematic changes in threshold θ, between 0 and 1, in increments of 0.01. These simulations were run for reserved instance contract lengths of both 12 and 36 months. Further details of the simulation are as follows:

Market demand data

We are not aware of any public-domain real-world datasets showing historical demand data for cloud computing resources, aggregated over a large number of users, over the kind of timescales that a broker such as the Coordinator in the WZH model would need to operate on. Nevertheless, intuitively, we can expect there to be periodic fluctuations in demand from any one user, and yet we can also expect that in some market sectors there will be significant correlations in demand from the population of users (or "user-base") within that sector. If demand across all sectors is sufficiently decorrelated, then aggregate demand would be a constant "white noise" and provisioning for that demand would be relatively straightforward.

However, although such an aggregate white-noise steady-state of demand would be desirable, it is unlikely to be achieved in practice for sustained periods. It is the presence of difficult-to-predict peaks and troughs in aggregate demand from real-world markets that makes the problem of adequately provisioning for that demand a deep and challenging research issue–the research issue that is addressed here. For this reason, we have used public domain data on real-world market activity that can plausibly be argued to serve as a good proxy for real-world demand for cloud computing in certain market sectors, and where there are known to be peaks and troughs. Our belief is that by demonstrating success on this real-world proxy data, we can plausibly claim that success of our system would *also* be likely for any other similarly fluctuating pattern of demand.

We assume that demand for computing resources by an e-commerce application will be related to sales being executed by the application (for example, an increase in on-line sales will result in an increase in

web server and credit card processing). Following this assumption, datasets were obtained from the UK National Statistics Office on the Non-Seasonally Adjusted Index of Sales at Current Prices from 1988 (earliest available) to 2011 for four different market sectors. These four sectors were chosen as they have a strong relationship to IT usage and they vary differently over the period, therefore allowing the new model to be evaluated in a number of market conditions. The duration over which these statistics were gathered represents a typical period of modern times where demand has changed frequently, with times of both recession and growth. As such, it is a plausible model of market variance. The demand patterns were normalised between 0, where none of the N users submit a resource request, and 1, where all N users submit a resource request. This allows the model to be assessed between the extremities of low and high demand.

All of the demand profiles showed regular annual patterns (such as large increases around Christmas), but to aid evaluation each profile is assigned a name appropriate to its market behaviour over the entire period:

Rapid growth market

(Figure 1)–Data on Non-Store Retailing: All Businesses–Little annual growth followed by rapid growth over smaller period.

Steady growth market

(Figure 2)–Data on Non-Store Retailing: Large Businesses–Steady annual growth throughout.

Recession and recovery market

(Figure 3)–Data on Non-Store Retailing: Small Businesses–Shrinkage in annual demand followed by period of recovery.

Steady market

(Figure 4)–Retail of Computer and Telecoms Equipment–Some peaks and troughs but fairly steady throughout.

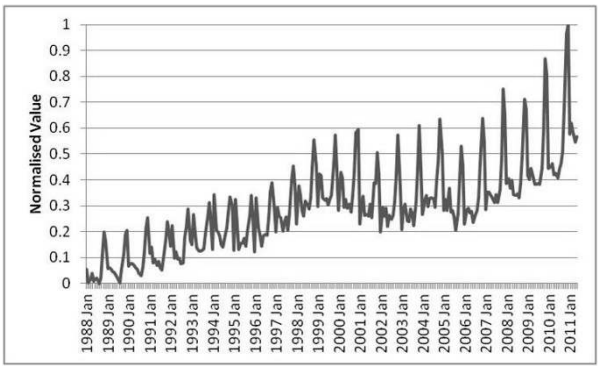

Figure 2 Non-store retailing: large business (steady growth) market profile.

User agents

Each user records when it has previously needed a resource during the simulation. When asked to submit a probability to the broker, the user will aggregate all the data from the same month in previous years to determine the probability it will need a resource again.

In the second period, the number of users who execute is determined from the market demand for that month. The users who execute their right to use a resource are chosen at random.

The approach means that users are submitting a true probability based on their previous performance, but may not execute at this probability due to changes in market demand.

Service provider pricing

The unit of resource being purchased is an Amazon Web Services EC2 Standard Small Instance (US East). At the date of simulation (July 2011), these were being advertised at a cost of D_h = \$0.085/hour (approximately \$60 for a whole month of usage) for an on-demand instance. For reserved instances, the same type instance

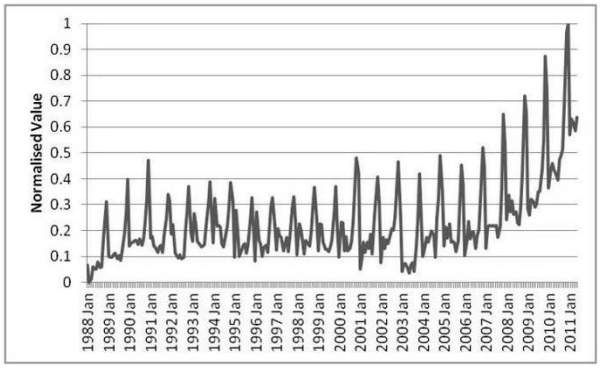

Figure 1 Non-store retailing: all business (rapid growth) market profile.

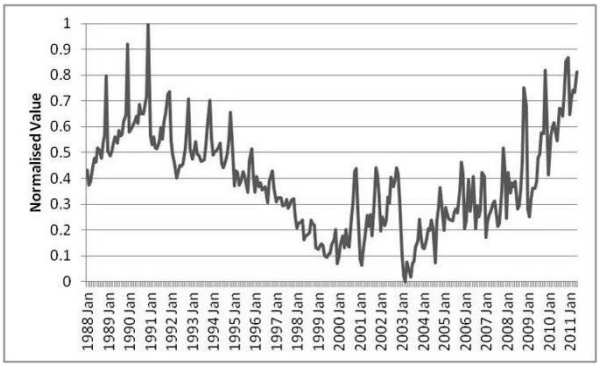

Figure 3 Non-store retailing: small business (recession&recovery) market profile.

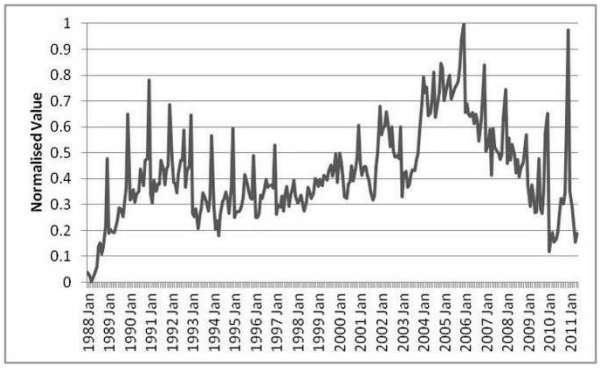

Figure 4 IT equipment (steady) market profile.

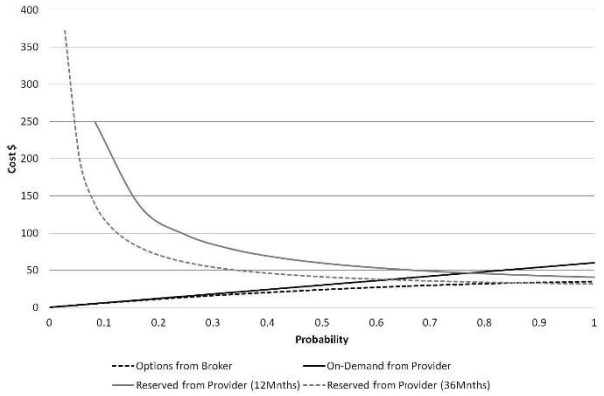

Figure 5 Resource pricing comparison.

for 12 months costs R = \$227.50 plus R_h = \$0.03/hour, and costs R = \$350 plus R_h = \$0.03/hour for a 36 month reserved instance.

Broker pricing

Users are charged a price to access the instance for a calendar month based on the values of $f(p_i)$ and $g(p_i)$ suggested by Wu *et al:*

$$g(p_i) = \frac{kp_i^2}{2} \text{ if required}$$

$$f(p_i) = 1 + \frac{k}{2} - kp_i + \frac{kp_i^2}{2} \text{ if not required}$$

To incentivize users to use the broker's service, it must save them money compared to going to the provider direct.

If the user is submitting honestly (which was proven by Wu *et al.* to benefit the user), they will expect to pay:

$$w(p_i) = p_i f(p_i) + (1 - p_i)g(p_i)$$

If they purchase resources directly from the provider, they will expect to pay: $w(p_i) = Cp_i$ where C is the on-demand cost of a resource.

Figure 5 shows the pricing model for the simulation. Wu et al.'s original pricing equations were based on an on-demand price of \$2. In our simulation, the cost of an on-demand instance for the entire month is \$60. Using a simple iterative algorithm, it was found that increasing Wu et al.'s original pricing equations by a factor of 35 maximises the brokers profit, while remaining cheaper for the user than using the providers service directly.

Although the broker in our model is only offering monthly options, in a practical implementation users would be free to purchase additional on-demand capability from the provider or broker for smaller units of time during periods where demand is above their current quota of options. Where the application has been built to rapidly scale automatically on a cloud infrastructure, this allows a user to make cost reductions through forecasting, without being confined to a limited amount of resources.

Results

Our approach determines the total profit made by the broker for different values of the threshold, for a number of market demand profiles and both 12- and 36-month contract lengths. Figures 6 and 7 show the total profits obtained for the various levels of thresholds tested, for 12- and 36-month contract terms respectively.

The broker is profitable

From Figures 6 and 7, it can be seen that the broker always profits, regardless of the market profile, contract length of the reserved instance, or the threshold (apart

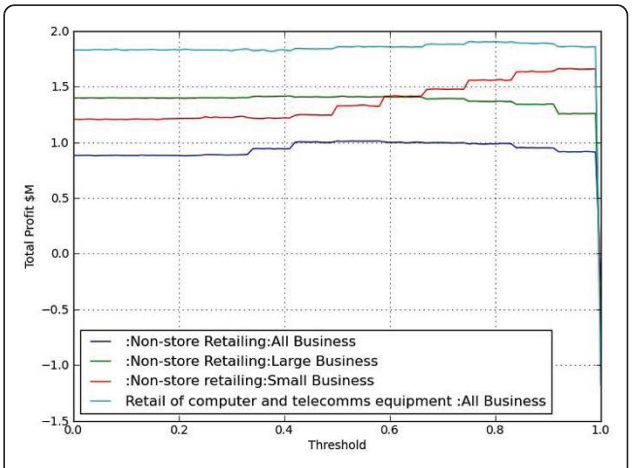

Figure 6 Total profit achieved for different thresholds using 12 month reserved instances.

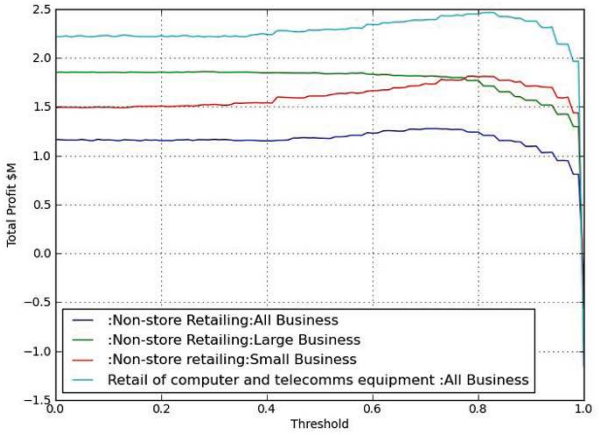

Figure 7 Total profit achieved for different thresholds using 36 month reserved instances.

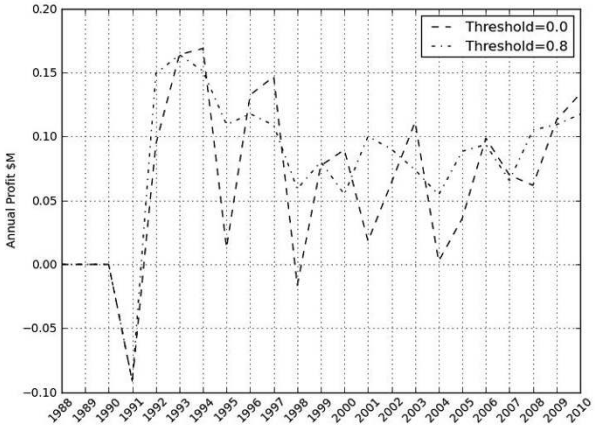

Figure 8 Annualised profit for broker in recession and recovery market profile, using 36 month reserved instances.

from the trivial case when the threshold is 1 and reserved instances are never purchased).

In our experiments, the worst performing case was when the broker purchases 12 month instances when the market grows rapidly and reserved instances were always purchased by broker. However, even in this scenario the broker still manages to make a total profit of $0.8 M.

By considering past performance through the use of the optimum threshold, and purchasing 36 month instances instead of 12-month ones, the broker can increase its profit by 44% in the same market conditions. If these conditions were to change to that of a steady market, this profit increases by 164%. This was the best performing case in our simulations.

Considering past performance benefits the broker

Generally, the broker benefits by setting a threshold so that reserved instances are purchased based on previous performance, instead of being purchased as standard practice. Setting the optimum threshold can increase profits by up to 21% when purchasing 36 month reserved instances, and 36% when purchasing 12 month reserved instances (Table 1); this is illustrated in Figure 8.

Using Figure 8 as an example, we can see why the broker profits when past performance is considered. When 36 month instances are always purchased, profits cycle every 36 months. The broker does not consider past performance, so large investments are made in reserved instances at these intervals which subsequently decreases profit. These reserved instances are subsequently not fully utilised by user demand, as shown in Figure 9.

When the threshold is set to the previously determined optimum threshold, we see that the cost of large investments made every 36 months decreases and the profit is smoothed. The broker now only buys resources when it believes they are required. Investments are made throughout the simulation, as and when are required, rather than at regular intervals. From Figure 9, it can be seen that over purchasing of resources is significantly reduced and more closely matched to the annual demand, hence the decrease in costs. The amount of investment made is proportional to market demand as shown in Figure 9 as users submit their requirements and the broker buys up capability.

A similar pattern is seen for 12 month instances, illustrated in Figure 10, where more reserved instances are available than required. By examining the broker's

Table 1 Profits (and increase) achieved for threshold of 0 and optimum threshold

(θ = Threshold)	36 Months				12 Months			
	Profit $M				Profit $M			
	$\theta = 0$	$\theta = \theta_{opt}$	Change	θ_{opt}	$\theta = 0$	$\theta = \theta_{opt}$	Change	θ_{opt}
Rapid Growth	1.15	1.27	10.4%	0.72	0.88	1.00	13.6%	0.50
Steady Growth	1.85	1.85	N/A	0.00	1.39	1.41	1.4%	0.38
Recession & Recovery	1.48	1.80	21.6%	0.80	1.21	1.65	36.4%	0.92
Steady	2.22	2.45	10.4%	0.82	1.82	1.89	3.8%	0.80

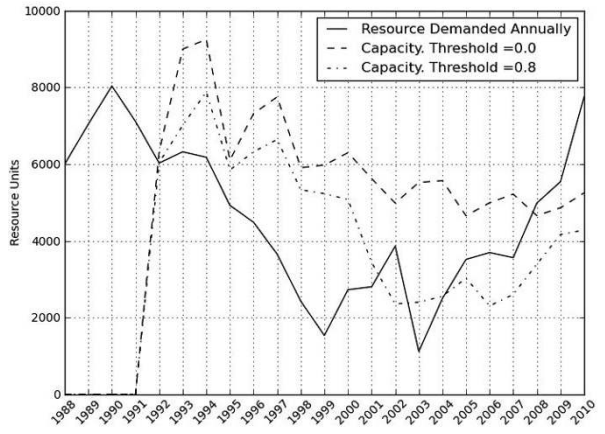

Figure 9 Annualised resource usage for broker in recession and recovery market profile, using 36 month reserved instances.

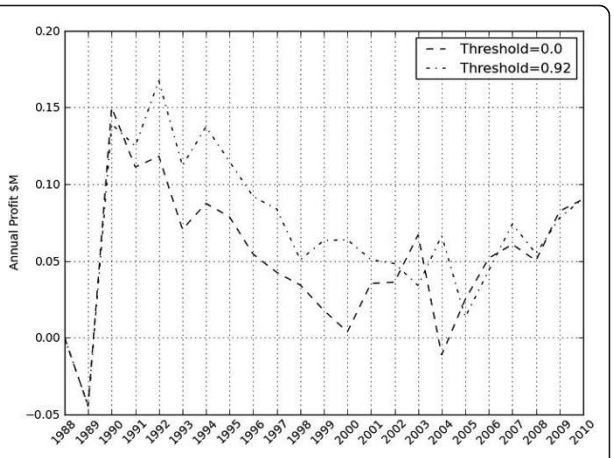

Figure 11 Annualised profit for broker in recession and recovery market profile, using 12 month reserved instances.

capacity and profit in Figures 10 and 11 when the threshold is set to be optimum, we see that more profit is achieved due to the predicted demand closely matching that available in reserved instances. This pattern is repeated for the other market profiles.

However, we can see that for the steady growth market where the broker uses 36 month reserved instances, the optimum threshold is at 0, meaning that the broker should always buy a reserved instance. Reserved instances purchased for sale in the steady growth market are likely to be fully utilised due to the increased demand experienced in subsequent periods. A similar pattern is seen when 12 month reserved instances are used.

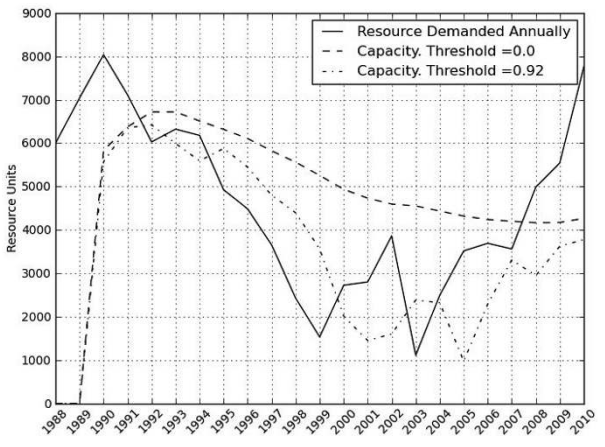

Figure 10 Annualised resource usage for broker in recession and recovery market profile, using 36 month reserved instances.

It is more profitable for the broker to purchase longer-term contracts

Table 2 show increases in profit by choosing 36 month contract terms over 12 months. It was found that it is always worth paying the higher upfront cost for 36 month reserved instance contract terms. When longer-term instances are purchased, total profit is increased by 22- 33% when the threshold is 0, depending on the market conditions. When the threshold is set to its optimum, this profit is increased by 9-31%. Although the upfront cost is larger, longer contracts allow the broker to maximise profit by being able to provider instances at a lower cost per unit.

Further market observations

In the rapid growth market, the optimum threshold is 0.5 for 12 month instances and 0.72 for 36 months. In this market, for most of the time, past performance will be a fair indicator of future performance due to steady annual demand, hence the moderate threshold level. However, when rapid growth hits, the reserved instances will be better utilised and become more profitable– when this happens, the 36 month instances reap the benefits by being available for times of boom, hence the higher threshold.

In the recession and recovery market, high optimum threshold values of 0.8 for 36 month reserved instances

Table 2 Increase in profits achieved by using 36-month reserved instances over 12-month reserved instances

(θ = Threshold)	$\theta = 0$	$\theta = \theta_{opt}$
Rapid Growth	30.7%	27.0%
Steady Growth	33.1%	31.2%
Recession & Recovery	22.3%	9.1%
Steady	22.0%	29.6%

and 0.92 for 12 month were identified. A higher threshold protects the broker from investing in a reserved instance that is subsequently not used when demand decreases due to recession. Investing in a 36 month reserved instance before a period of recession is likely to result in a loss, as demand for these will decrease and the upfront cost not repaid–this explains the high threshold.

In the steady market, again, it is most profitable to use a fairly high threshold. As there are few large changes in market demand, reserved instances will not always make a good return because they are not being regularly fully utilised by periods of growth. In this case, the broker must be cautious in investing.

Discussion

Benefits

The approach discussed in this paper has a number of benefits. The broker translates user's on-demand behaviour into reserved instance requirements by incentivising them to accurately predict their own usage. In turn, providers have a better view of future demand across the entire user-base rather than just the customers who are resource intensive enough to purchase a reserved instance of their own. This also means that in our system presented here, providers have a 12- or 36-month indication of market demand, rather than just the one month that was the case in Wu *et al.'s* original formulation of the WZH model, or no indication (as per the case of an on-demand instance). This information can be used to plan future staffing levels, to schedule workloads more effectively, or to reduce costs by purchasing swing options for electricity.

For the user, costs can be reduced by forecasting some, or all, of their resource requirements, and purchasing options to match these predictions. Options provide cost-saving potential to the user without locking them into higher prices or obligating them to purchase a resource.

Application

On-demand pricing is regarded as one of the key attributes and benefits of cloud computing. Does an options-market for cloud computing, which requires an element of forecasting by the consumer, negate this major benefit?

We believe that options contracts are one of a potential armoury of financial instruments in cloud computing that can benefit the consumer. Options could be used in conjunction with pay-as-you-go pricing, and other instruments to obtain best value for the consumer.

An application could be designed such that it takes advantage of long term options contracts should it believe that they will be adequately utilised. If a surge in demand was experienced by an application and it was likely an additional resource would be required for the whole month, an option could be executed automatically to provide whole-month access. Should further increases in demand be seen, additional on-demand resources could be started and integrated with the already operational option resource. These on-demand resources could start and stop to suit smaller periods of additional demand over the capacity provided by the option. Cloud computing is vital to the operation of the model, as it allows resources from a number of providers, purchased through a number of instruments and lasting for a variety of time periods to be aggregated together to provide rapid and automatic scalability, while aiding in capacity planning and reducing costs for the consumer. Figure 12 shows a fictional example of resource requirements for an application over a four year period by month.

By considering performance of the application for each month of the three year period, the consumer can save money in the fourth year. In January, a minimum of two resources were used by the application for each year. If we consider previous performance, the likelihood these resources will be used is very probable, so we can purchase two resource options with $p = 1$. This is the essentially a futures contract as the consumer will definitely require the resource in the next period. In the same month, four additional resources were required for two of the three years. As such, these four resources are purchased as options with $p = 0.66$. In the final year, two more resources were required, so two options where $p = 0.33$ are purchased.

If the actual demand experienced in the fourth year is 9 units, the consumer will execute their right to buy the resource for all eight contracts. The consumer will also purchase another on-demand resource at the going rate. The total cost for their monthly resource usage is $375. If these were purchased on-demand, this would cost $540 for a whole month's utilisation. The use of options in this case saves the consumer 30%. We believe many

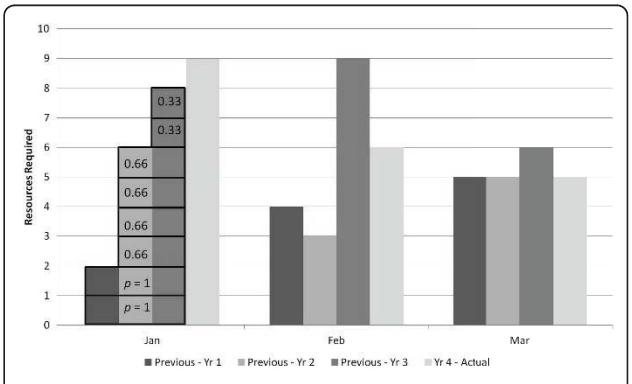

Figure 12 Example web application resource requirements.

websites and web applications would show at least some predictability in resource requirements. For example, in the run up to Christmas, it is likely many e-commerce websites will experience a surge in traffic and that the probable minimum size of this increase can be predicted using historical data. Retailers can purchase the resources they are likely to utilise thereby reducing costs, but retain the benefit of being able to take extra on-demand resources should they need to scale quickly due to increased traffic. These extra resources could be purchased for the hour or longer depending on requirements. The period of the options can be changed depending on the application. For some websites, a minimum number of users might be accessing the site at all times of day in a particular month. The website can purchase an option which entitles them to use the minimum number of resources for that day. Should this be exceeded at any point through the course of their options contract they can purchase on-demand instances for those hours that are busier.

Offering American options

Wu *et al.* suggested that the probability-based options they originally proposed could be combined to allow users to pay for the right to use a resource at any point in an agreed period. We propose that the broker could offer these combined contracts as a separate pricing model in a manner similar to an American Option.

The options contracts that we have discussed thus far are known as European options, which allow delivery of the resource at a specific date in the future. An alternative contract is an American option. This gives the holder the right to use a resource at any time until the expiry of the contract.

At the beginning of a year, a customer may know they will need to use a resource for an entire month during that year. However, it may not know which month it will need it. The broker can offer the customer a contract that allows them to execute their right during any point in the year, but saves them money compared to purchasing on-demand instances. In a period of one year, all other things being equal, there is a 1/12 chance that the holder of the contract will need to use their resource in a particular month. So the cost of the contract w is the cost of utilising a resource with that probability in one month, plus the cost of reserving the resource for the remaining eleven months.

$$w(p_i) = Cp_i$$

This comes to around \$59, \$1 cheaper than if on-demand instances were used. This also aids the broker, as the probability of 1/12 for each month contributes to the forecast calculation.

If the consumer has an application with a high probability of execution over 12- or 36-month period, they may choose to buy a reserved instance themselves and use it as and when is required–this can be save the consumer money (see Figure 5). However, for smaller probabilities the user is better off using options. The consumer could, of course, aggregate reserved instances, on-demand instance and options together to achieve best value.

These American options would most suit industries where demand over larger periods is fairly predictable, but where exact details of when demand will peak are not known. This could be where the contract holder's customer demand is based on externalities which are unpredictable or otherwise out of the holder's direct control.

The contract details could vary depending on the nature of the customer's requirement. For example, a website supplying garden equipment might know at the start of the British summer that, at some point, they will have a large surge in demand for barbeques and outdoor recreational equipment. This demand usually is related to high temperatures and sunshine. The owners of the website know this period usually lasts for two months (although the British public may be optimistic of longer!). However, because the weather is unpredictable they do not know exactly when it will happen. This consumer can reduce costs by purchasing an American option with an expiry date six months in advance, which entitles them to use the resource for two months. The cost of this is:

$$w = 2g\left(\frac{2}{6}\right) + 4f\left(\frac{2}{6}\right)$$

This comes to \$105, \$17 cheaper than if they used on-demand resources.

A news and travel website is another example. It may typically experience a number of periods of high demand during a year–but when this demand will happen is difficult to predict as it might be related to acts-of-god, traffic accidents or spikes in social media activity.

Mobile-phone operators may have surges in visits to their websites when new handsets are released. It may be simple to predict how many new handsets the mobile operator wishes to offer to customers. However, it might be more difficult to determine when manufacturers will make these available to the operator.

These American options could also be used to provide redundancy for an IT infrastructure. If an IT Director knows that an IT or telecommunications system is built for 99.9% availability with a Mean Time to Repair (MTTR) of four hours, she can estimate the amount of

time in a particular period that the system will be down. In this case, the system will be down for around 8 hours a year. As the MTTR is 4 hours, it is likely two of these incidents will occur annually. The director can buy two options, each which entitles her to use four hours of resource at any point during the year. This will cost very little but will provide access to a resource should the infrastructure fail for less cost than that of an on-demand instance.

Conclusions

An extension of the WZH mechanism was proposed and implemented in an agent-based simulation using real-world economic demand data, using current costs of an Amazon Web Service cloud instance, and where users submit probabilities based on previous demand. It was found that the broker profits in such a situation in a number of market conditions thereby demonstrating that a stable commercial implementation is feasible. It was also found that the broker can increase profits by considering past performance and purchasing longer-term contracts.

In the worst case scenario discussed in this paper the broker still makes nontrivial profit. Small changes to the broker's operating procedures such as purchasing longer term reserved instance contracts can improve profits by over 30%. Considering past performance can also reward the broker with increased profits, by up to 36% in one experiment discussed in this paper. In ideal market conditions, where longer term contract terms are used and an optimum threshold set, profit was seen to increase by 165%.

The WZH mechanism provides a useful theoretical foundation for an options-market in computing resource. However, the service provider would have to provide specific pricing to support the Coordinator, and this might not always be profitable for the service provider. Our extension to this model does not require new pricing to be agreed, but contract restrictions on reselling may be a barrier to commercial implementation.

Our work shows that financial brokering in computing capability has potential as a viable commercial proposition, and that all parties can potentially benefit as a result of such a system. The advantage of this approach is that a forecast of future usage requirements is obtained, which can be subsequently used to plan future capacity requirements and so that targets on performance as detailed in a Service Level Agreement can be met.

The optimum threshold is the value at which market demand is fully anticipated by the broker and which is fully provisioned through reserved instances. Determining this threshold mathematically is likely to be challenging due to difficultly in determining market dynamics over a very long period. However, an empirical simulation using actual market data could produce such a threshold for commercial implementation. It may also be possible for the broker to track performance, and update its threshold in real-time.

Further work should be carried out to translate this forecast information into tangible benefits for the provider, which might include automatic purchasing of electricity swing options, staff scheduling, or more efficient distribution of workloads to meet SLA's.

We believe that the results we have presented here clearly demonstrate that options and other financial derivatives contracts, when appropriately applied and developed for cloud computing, offer benefits to both consumers and service-providers, and (crucially) the brokerage function can also be a profitable business too. Combining the forecasting benefits of both American and European options and other derivatives with the convenience of existing spot and on-demand pricing is likely to benefit all participants in cloud-computing markets: providers, users, and intermediaries.

By taking the results from this paper and extending them with future research into the performance of the extended WZH model under different conditions and in different segments, a commercial offering that is profitable to the broker, beneficial to the user, and with a calculated level of risk looks likely to be readily achievable.

Acknowledgements

We thank the UK Engineering and Physical Sciences Research Council (EPSRC) for funding the Large-Scale Complex IT Systems Initiative (http://www.lscits.org) as well as HP Labs Adaptive Infrastructure Lab for providing additional financial support.

Authors' contributions

OR designed and implemented the simulations, analysed the results, and wrote the first complete draft of this paper. The novels aspects of the cloud brokerage model described in this paper are entirely OR's contribution. DC provided direction and advice, proposed research areas and reviewed and edited earlier drafts of this paper. Both authors read and approved the final manuscript.

Authors' information

Owen Rogers is a doctoral researcher at the University of Bristol. Owen graduated from Cardiff University with a MEng in Computer Systems Engineering in 2005. Following this, he joined telecommunications firm Cable & Wireless, first as a Graduate Engineer, and then as Product Development Manager for Managed Security Services. In 2009, Owen joined managed services provider Claranet as Product Portfolio Manager, before returning to academia in 2010. He is currently researching financial mechanisms for improving cloud computing's potential as a tradeable commodity. Owen is a Chartered Engineer through the British Computer Society.

Dave Cliff is a professor of computer science at the University of Bristol. He has previously held faculty positions at the universities of Sussex and Southampton in the UK, and at the MIT Artificial Intelligence Lab, USA. He also spent seven years working in industrial technology research, first for Hewlett-Packard Labs and then for Deutsche Bank's Foreign Exchange Complex Risk Group. Since 2005, Dave has been Director of the UK Large-Scale Complex IT Systems Initiative (http://www.lscits.org). He is a Fellow of the British Computer Society.

Competing interests
Owen Rogers' PhD studentship is part-funded by Hewlett-Packard Research Labs. Primary funding for both Rogers' and Cliff's work on this project was provided by a research grant from the UK Engineering and Physical Sciences Research Council. The authors list these sources of funding for completeness, and as a mark of acknowledgement and grateful thanks. Neither author considers these sources of funding to constitute competing or conflicting interests in the work.

Received: 21 November 2011 Accepted: 12 April 2012
Published: 12 April 2012

References

1. Weinman J (2011) Time is Money: The Value of On-Demand. http://joeweinman.com/Resources/Joe_Weinman_Time_Is_Money.pdf. Accessed 21st November 2011
2. Armbrust M, Fox A, Grittith R., *et al* (2010) Above the clouds: a Berkeley view of cloud computing cloud computing: an old idea whose time has (Finally) come. Commun ACM 53(4):50–58. doi:10.1145/1721654.1721672.
3. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Generation Comput Syst 25:599–616. doi:10.1016/j.future.2008.12.001.
4. Shapiro, Farrell, Varian (2004) The Economics of Information Technology–An Introduction. Cambridge
5. Patel CD, Shah AJ (2005) Cost model for planning, development and operation of a data center. Development 107:1–36
6. Barroso LA, Hoyle U (2007) The case for energy-proportional computing. Computer 40(12):33–37
7. Sandholm T, Lai K, Andrade J, Odeberg J (2006) Market-based resource allocation using price prediction in a high performance computing grid for scientific applications, HPDC'06. Proceedings of the 15th International Symposium on High Performance Computing 1:132–43
8. Stage A, Setzer T (2009) Network-aware migration control and scheduling of differentiated virtual machine workloads. Proceedings of the ICSE Workshop on Software Engineering Challenges of Cloud Computing 1:9–14
9. Hull JC (2007) Fundamentals of Futures and Options Markets. Pearson Education, 6
10. Clearwater SH, Huberman B (2005) Swing options: a mechanism for pricing IT peak demand. International Conference on Computing in Economics
11. Papazoglou M (2007) Service oriented architectures: approaches, technologies and research issues. VLDB J 16(3):389–415. doi:10.1007/s00778-007-0044-3.
12. Wu F, Zhang L, Huberman B (2005) Truth-telling reservations. Lect Notes Comput Sci 3828:80–91. doi:10.1007/11600930_9.
13. Rogers O, Cliff D (2010) The effects of truthfulness on a computing resource options market. Proceedings of the 2nd Annual International Conference on Advances in Distributed and Parallel Computing 2:330–335
14. Hetland M (2008) Beginning Python: from novice to professional. APRESS

doi:10.1186/2192-113X-1-2
Cite this article as: Rogers and Cliff: **A financial brokerage model for cloud computing.** *Journal of Cloud Computing: Advances, Systems and Applications* 2012 **1**:2.