Journal of Cloud Computing
a SpringerOpen Journal

**RESEARCH** **Open Access**

# Correcting a financial brokerage model for cloud computing: closing the window of opportunity for commercialisation

John Cartlidge[1*] and Philip Clamp[1,2]

**Abstract**

In April 2012, Rogers and Cliff (R&C) demonstrated a theoretical financial brokerage model for cloud computing that is profitable for the broker, offers reduced costs for cloud users, and generates more predictable demand flow for cloud providers. Relatively cheap, long-term reserved instances (RIs) are bulk-purchased by the broker, and then re-packaged and re-sold as monthly options contracts at a price lower than a user can purchase "on-demand" from the provider. Thus, the broker risks exposure on purchase for margin on sales. R&C's result has generated significant interest in the cloud computing community and is currently the fifth most accessed research paper of all time in the *Journal of Cloud Computing: Advances, Systems and Applications*.

Here, we perform an independent replication of R&C's brokerage model using CReST, a discrete event simulation platform for cloud computing developed at the University of Bristol. We identify two implementation problems in R&C's original work: firstly, the broker buys fewer RIs than the model suggests; and secondly, the broker is undercharged for RIs used. We correct R&C's results accordingly: while broker's profits are not as high as R&C suggest, the model still supports the theoretical possibility of a profitable brokerage.

However, aggressive competition between cloud providers has reduced the cost of cloud services to users and led to the introduction of new secondary markets where users can buy and sell RIs between themselves. This has squeezed the opportunity for an intermediary brokerage. By recalibrating R&C's model to fit current market conditions, we conclude that the commercial viability of R&C's brokerage model has been eradicated. The window of opportunity has now closed.

**Keywords:** Cloud computing; Cloud brokerage; Replication; Simulation; CReST

## Introduction

In April 2012, Rogers and Cliff (R&C) demonstrated a theoretical financial brokerage model for cloud computing that is profitable for the broker, offers reduced prices for cloud users, and generates more predictable demand flow for cloud providers [1]. The broker achieves this by acting as an intermediary between cloud providers and cloud users. Firstly, the broker bulk-purchases relatively cheap, long-term 12-months or 36-months reserved instances (RIs) from the provider. Then, the broker re-packages as monthly units and re-sells to cloud users at a price *lower* than users can otherwise purchase directly "on-demand" from the provider. The broker thus risks exposure to loss on large up-front purchase costs that may never be recovered, for the potential reward of profit margin generated on each unit sold—i.e., the difference between the purchase cost and sale price. As a consequence, cloud providers benefit from more predictable demand (achieved by bulk-selling long-term RIs); and cloud users benefit from access to cheaper prices. Thus, by providing liquidity, the broker improves market efficiency. Having potential for profitable commercial exploitation, this theoretical result has generated significant interest in the cloud computing community: since its publication 18 months ago, R&C's model has become the fifth most accessed research paper of all time in the *Journal of Cloud Computing: Advances, Systems and Applications* [2].

*Correspondence: john@john-cartlidge.co.uk
[1]Department of Computer Science, University of Bristol, Woodland Road, Bristol, BS8 1UB, UK
Full list of author information is available at the end of the article

However, the cornerstone of science is the principle of replication. New findings should only be provisionally accepted—and considered with scepticism—until verified by independent replication (cf. the infamously high-profile discovery, and subsequent refutation, of neutrinos having velocity greater than the speed of light in a vacuum, e.g., [3]). Here, we perform an independent replication of R&C's brokerage model using CReST, the Cloud Research Simulation Platform (freely available for open-source download [4]). CReST was developed at the University of Bristol to address the need for a robust simulation modelling tool for research and teaching of data centre (DC) management and cloud provision; and has successfully been used to refute, revise and extend findings in the cloud computing literature [5].

Our replication study identifies two implementation problems in R&C's original work. Firstly, the broker buys fewer RIs than the model suggests; we call this the *"reservations bug"*. Secondly, the broker underpays for RIs used; we call this the *"payment bug"*. We demonstrate the effects of these bugs on the brokerage model and correct R&C's published results: while broker's profits are not as high as R&C suggest, the model still supports the *theoretical possibility* of a profitable brokerage.

However, since the publication of R&C's model, aggressive price competition between cloud infrastructure providers has dramatically reduced the cost of cloud services to end users, both from the provider directly (the primary market) and through newly introduced secondary markets where users buy and sell RIs second-hand. To reflect this, we recalibrate R&C's model using current prices charged by Amazon Web Services (AWS). Results demonstrate that competition between providers has increased market efficiency and squeezed the opportunity for a profitable intermediary brokerage. At today's prices (September, 2013), and with updated constraints to reflect the introduction of secondary markets, R&C's model is *no longer theoretically profitable*. We conclude that increased competition between vendors has eradicated the commercial viability of R&C's financial brokerage model for cloud computing. As such, the window of opportunity for commercial exploitation has closed.

The rest of this paper is organised as follows. In Section 'Background' we introduce R&C's brokerage model and describe our CReST implementation in Section 'Experimental method'. In Section 'Experiment 1: Model verification' we validate our model replication by generating results almost identical to those presented by R&C. The effects of the reservations bug (Section 'Experiment 2: the reservations bug') and payment bug (Section 'Experiment 3: the payment bug') identified in R&C's implementation are then demonstrated, before a set of "corrected" results for R&C's model are presented (Section 'Experiment 4: corrected replication of R&C's brokerage model'). Finally, in Section 'Experiment 5: Recalibrating R&C's model to reflect current market conditions' we recalibrate R&C's model to reflect the current market for cloud provision and observe the impact this has on theoretical profits. The paper finishes in Section 'Conclusions' with our primary conclusion that R&C's brokerage model is *no longer commercially viable.*

## Background

### Pricing the cloud

The *on-demand* delivery model for cloud computing resources offers a variety of benefits for business consumers: no up-front capital expenditure on (often underutilised) compute infrastructure needed to cover peak business demand; flexibility and scale-out opportunities from the ability to start and stop VM instances at will; and reduced operational costs from outsourcing maintenance and support [6]. However, the on-demand model is not necessarily ideal for cloud providers, as they attempt to adhere to strict Service Level Agreements in the face of fluctuating demand. If providers could accurately forecast future resource demand, then they would have the opportunity to reduce costs by optimising electricity purchases, engineering staff, and hardware utilisation, etc. [7]. At present, most providers offer a choice between an *on-demand* tariff ("on-demand" instances, billed per hour of usage with no upfront purchase cost [8]) and a *contract* tariff ("reserved" instances (RIs), leased for a fixed time period and offering significant hourly usage discounts [9]). Some providers, e.g., AWS, offer an alternative *spot price* tariff that varies in real-time based on current supply and demand [10]. However, of these methods, only long-term RI contracts (12-months or 36-months) have the potential to aid the provider in capacity planning.

Several alternative pricing models have been proposed in academic research, most notably involving *derivatives contracts*, such as (European) *options* [11]. Options contracts involve the payment of an up-front fee that gives the buyer the legal right, but not the obligation, to purchase a resource for an agreed strike-price on some later delivery date [12]. These types of financial instruments are commonly used in financial markets, with underlying assets ranging from commodities such as wheat and oil, to a suite of complex financial products. In 2012, R&C demonstrated that it was possible for a cloud computing brokerage to utilise options contracts to provide cheaper resources to consumers while simultaneously aiding providers in predicting future demand [1]. Significantly, R&C also demonstrated that the broker could profit from providing this service.

R&C's result has the potential to positively disrupt the delivery and pricing of cloud services. As the market in cloud resources matures and becomes more standardised, the promise of a *federated cloud*—where cloud users can migrate between providers seamlessly—will enable resources to be traded as a commodity; eradicating existing concerns of vendor lock-in. In turn, this will open opportunities for brokers to enter the market, acting as intermediary *market makers* between users and providers. In such a scenario, R&C's result could have commercial as well as academic significance.

## R&C's options model for cloud computing

Typically, the role of a broker is to facilitate the matching of supply and demand in a market. Brokerage services primarily generate profit by charging commission fees, and/or *making the spread* by buying at a lower price and selling at a higher price. In the cloud brokerage model of R&C [1], the broker aims to make a profit by purchasing long-term advanced obligations on resources (12-months or 36-months RIs), and repackaging them as one-month options contracts to users.

The brokerage model of R&C consists of two stages: (1) each month, the broker takes orders from clients for future resource needs by selling *options*, and determines how many RIs to purchase; (2) in the following month, clients can request instances from the broker by *exercising* their options. If the broker has capacity available from previously purchased RIs, these are sold to the users at a profit. Otherwise, the broker must purchase additional (more expensive) on-demand instances from the provider to fulfil the obligation to the client.

R&C's brokerage model extends a pricing structure that was initially developed at HP Labs by Wu, Zhang, and Huberman (WZH) [11]. The WZH model financially rewards clients that reveal the *true likelihood* that they will utilise a resource in the future. Every month, each client, $i$, estimates the probability, $p_i$, of using a resource in the following month. Clients then submit their estimation, $p_i$, to the broker in order to purchase a resource contract. In the following month, the client is charged $Used(p_i)$ if the resource is used and $Unused(p_i)$ if the resource is not used, such that:

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

and

$$Unused(p_i) = \frac{kp_i^2}{2}. \tag{2}$$

If users choose instead to purchase resources directly from the provider, they will expect to pay $p_iP_D$, where $P_D$ is the on-demand cost of a one-month instance (in WZH's model, $P_D = 2$ [11]). It has been proven that this pricing model encourages users to truthfully submit their honest estimate of resource usage, $p_i$, when $k = 1.5$ [11].

We can consider WZH's contract as an options model if the broker charges clients $Unused(p_i)$ to purchase the option contract and then a further charge of $Used(p_i) - Unused(p_i)$ in the following month if the option is exercised (i.e., if the resource is used). The model can then be calibrated to real-world prices by multiplying $Used(p_i)$ and $Unused(p_i)$ by a *cost factor*, $C$. Setting $k = 1.5$, to ensure truthful submission of usage probability, this gives us the final options pricing model used by R&C [1]:

$$OptionPrice(p_i) = C\left(\frac{3p_i^2}{4}\right) \tag{3}$$

and

$$ExercisePrice(p_i) = C\left(\frac{7}{4} - \frac{3p_i}{2}\right) \tag{4}$$

where (3) is the *price of an option contract*, and (4) is the *price of exercising the contract*. Hence, the full cost of purchasing a one-month RI from the broker is:

$$TotalPrice(p_i) = OptionPrice(p_i) + ExercisePrice(p_i). \tag{5}$$

Figure 1 shows the price of options as a function of probability of usage, $p_i$. Prices assume a cost factor of $C = 1$ and a monthly on-demand price of $P_D = 2$ (as in WZH's model). We see that $OptionPrice(p_i)$ increases non-linearly from 0 when $p_i = 0$, to a maximum value of 0.75 when $p_i = 1$ (blue line); and $ExercisePrice(p_i)$ falls linearly, from a maximum value of 1.75 when $p_i = 0$, to a minimum value of 0.25 when $p_i = 1$ (red line). Therefore, the $TotalPrice(p_i)$ of purchasing and exercising an option (green line) falls non-linearly from a maximum value of 1.75 when $p_i = 0$ to a minimum value of 1 when $p_i = 1$. As a result, the probabilistic cost of an option ($p_i\ TotalPrice(p_i) + (1-p_i)\ OptionPrice(p_i)$; orange dots) is *always lower* than the probabilistic cost of purchasing an on-demand instance directly from the provider ($p_i.P_D$; purple dots). This is the maximum constraint on the model and ensures that the user will always benefit from purchasing options from the broker, rather than buying from the provider directly.

## R&C's brokerage implementation

R&C implement their brokerage framework using a simulation model consisting of: a population of user agents that have demand for cloud resources (VM instances); a cloud provider offering VM instances either on-demand or as long-term RIs; and a single broker with a market monopoly, acting as intermediary between the population of agents and the cloud service provider. We introduce each, below.

### *User agents*

Each month, agents $a_i \in A$ submit their probability, $p_i$, of demanding a resource in the following month. Agents calculate $p_i$ as the mean number of resources they have consumed during the same month in previous years. Since

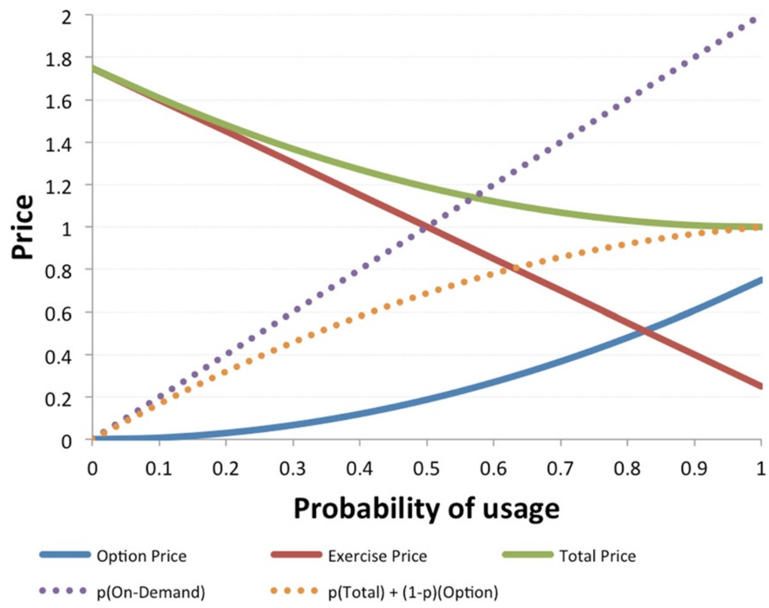

**Figure 1 Option prices.** With a cost factor $C = 1$ and on-demand price $P_D = 2$, the cost of options from the broker (orange dotted line) is always cheaper than purchasing an on-demand instance directly from the provider (purple dotted line).

each agent, $a_i$, has a maximum demand of one resource per month, then: $0 \leq p_i \leq 1$; and the predicted demand for the full population is: $\sum_{\forall i} p_i \leq |A|$, where $|A|$ is the size of the agent population.

### *Cloud provider*

The cloud provider offers VM services for sale: either on-demand, at price $P_D$ per month; or reserved for $R$ months with up front cost $P_{Ru}$, plus an additional cost $P_{Rm}$ per month of use. Since the provider offers RIs at a discounted rate, it is always true that:

$$P_D > P_{Rm} + \frac{P_{Ru}}{R}. \tag{6}$$

However, there exists a value $\hat{R} < R$ such that:

$$P_D = P_{Rm} + \frac{P_{Ru}}{\hat{R}}. \tag{7}$$

Thus, if a RI is purchased and utilised for *fewer* than $\hat{R}$ months, then there is no longer a cost saving; in fact, the RI will be *more expensive* than the on-demand price, $P_D$. For this reason, to guarantee savings on a RI purchase, one must be certain that it will be utilised for at least $\hat{R}$ months of the full $R$-months term.

### *Broker*

The broker aims to make a profit by taking the risk of purchasing long-term RIs ($R = 12$, or $R = 36$ months) from the provider and re-selling to users as one-month options contracts. Each month, users purchase options from the broker by submitting their probability of demand, $p_i$. The broker then uses Algorithm 1 to determine how many (additional) RIs to purchase. The broker first checks whether there is sufficient capacity available to cover the following month's demand, i.e., whether $f_{t+1} \geq \sum p_i$. If capacity covers demand, then no further RIs are purchased; else, additional RIs are incrementally purchased until either: broker capacity covers demand (Algorithm 1, line 3); or the expected *Marginal Resource Utilisation (MRU)* of an additional purchase falls below a minimum threshold, $\theta$ (Algorithm 1, line 5).

**Algorithm 1** Broker's algorithm used for purchasing RIs each month.

```
1:  function PURCHASERESERVEDINSTANCES
2:      $continue \leftarrow$ True
3:      while ($\Sigma p_i > f_{t+1}$ and $continue$) do
4:          $MRU \leftarrow \frac{1}{R}\sum_{d \in \mathbf{D}}[d > 0]$, where $[x] = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise} \end{cases}$    ▷ calculate MRU
5:          if $MRU \geq \theta$ then
6:              for $i \leftarrow 1$ to $R$ do
7:                  $f_{t+i} \leftarrow f_{t+i} + 1$    ▷ purchase new RI; update future capacity
8:              end for
9:          else
10:             $continue \leftarrow$ False    ▷ stop purchasing RIs
11:         end if
12:     end while
13: end function
```

The broker calculates expected *MRU* (Algorithm 1, line 4) by comparing historical demand, $\boldsymbol{H} = [h_{t+1-R}, h_{t+2-R}, \ldots, h_t]$, over the previous $R$ months, against the future resource capacity, $\boldsymbol{F} = [f_{t+1}, f_{t+2}, \ldots, f_{t+R}]$, (the number of RIs owned) over the forthcoming $R$ months. Using a simple forecasting mechanism that assumes future demand will equal previous demand lagged $R$ months, the broker then calculates an *expected deficit profile*, $\boldsymbol{D} = \boldsymbol{F} - \boldsymbol{H}$, for each forthcoming month by subtracting historical demand from future capacity, such that: $\boldsymbol{D} = [f_{t+1} - h_{t+1-R}, f_{t+2} - h_{t+2-R}, \ldots, f_{t+R} - h_t]$. Then, the expected *MRU* of an additional RI purchase is the proportion of months in $\boldsymbol{D} > 0$. More formally:

$$MRU = \frac{1}{R}\sum_{d \in \mathbf{D}} [d > 0]\,, \text{where}[x] = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

The *MRU* estimates the fraction of life ($months/R$) an additional reserved instance is likely to be utilised over the next $R$ months, based on historical demand. A minimum threshold, $\theta$, is then used to determine whether or not to make the purchase (Algorithm 1, line 5). If $MRU \geq \theta$ then the broker buys a new RI, estimating that it will be utilised enough to make a profit. Conversely, if $MRU < \theta$, the broker does not purchase a new RI, estimating that it will be under-utilised. For each new purchase, the broker's future capacity, $\boldsymbol{F}$, is incremented (Algorithm 1, line 7) and the while loop restarts. However, if $MRU < \theta$, then no more RIs are purchased and the while loop is exited (Algorithm 1, line 10).

In the following month, the broker delivers one-month instances to users that exercise their options. If the broker does not have the capacity to fulfil client demand, then additional on-demand instances are purchased directly from the provider to supply to the users. The broker is able to vary exposure to risk by altering the value of $\theta$ over the range $0 \leq \theta \leq 1$. When $\theta$ is low, the broker purchases more RIs per unit of demand than when $\theta$ is high.

## Experimental method

In this section, we describe our replication of R&C's brokerage model. To ensure that the replication is entirely independent, we have implemented R&C's model on a different platform (CReST) written in a different programming language (Java).

### CReST, the cloud research simulation toolkit

The Cloud Research Simulation Toolkit (CReST) was developed at the University of Bristol to address the need for a robust simulation modelling tool for research and teaching of data centre (DC) management and cloud provision. CReST is a discrete-event simulation platform written in Java, and is freely available open source under a GNU General Public License version 3.0 [4].

Although a variety of cloud simulation platforms exist (e.g., [13-16]), CReST has a unique feature set (see [5] for a detailed comparison) that enables simulation at multiple abstraction levels: from physical hardware, energy usage and thermal flows within a DC; to networked infrastructure and the virtualisation layer of application services supporting dynamic user demand.

CReST is designed as a set of *coupled modules* that can be independently switched on or off depending on the level of abstraction required. To ensure extensibility and module independence, CReST has a Model-View-Controller (MVC) architecture. Following Figure 2, each module has a *ModuleRunner* that views the time-ordered *EventQueue* model using Java's *Observer-Observable* interface. Modules observe each time-stamped *Event* popped from the *EventQueue* and decide whether to ignore the event or take appropriate action; which may include generating new *Events* that are subsequently pushed onto the *EventQueue* with a post-dated timestamp. Modules are therefore independent *Observers* and only interact *via* the *EventQueue*, ensuring strict delineation between modules and making it possible to switch modules on and off, delete modules, and add new modules with relative ease (for more detail on the architecture of CReST, refer to [5]).

For all experiments reported in this paper, we use CReST as the cloud simulation platform. To optimise simulation performance, we disabled several of the lower-level physical infrastructure modules, such as the *Thermal*

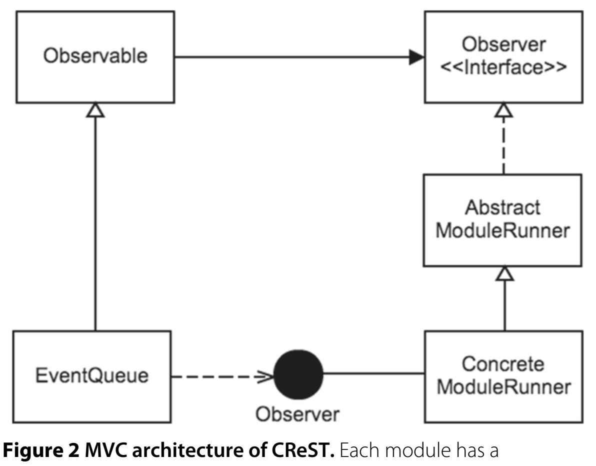

**Figure 2 MVC architecture of CReST.** Each module has a *ConcreteModuleRunner* object that extends the *AbstractModuleRunner* class. The *AbstractModuleRunner* implements the *Observer* interface and uses the method *Observer.update()* to view each new timestamped *Event* popped from the time-sorted *EventQueue*. The *ConcreteModuleRunner* then chooses to ignore or react to the *Event*, which may cause future *Events* to be generated and added to the *EventQueue*.

module that tracks heat-flow in the data centre. The active modules used in all of the brokerage simulations that we perform include: *Brokerage*, *Pricing*, *Events*, *Services* and *Simulation*. This enables us to efficiently run experiments that simulate decades of time, without compromising on the abstraction level needed. All CReST code used to run the experiments performed here, and associated Python scripts used for data analysis and visualisation, are available to download in version 0.4.0 of CReST [4].

### Parameter configuration settings

The configuration settings used for all experiments, unless otherwise stated, are detailed below:

- **Demand profiles:** Following R&C, to simulate *realistic* demand for cloud resources, we consider four demand profiles generated using real demand data over the period 1988-2011 for a variety of IT-related industries. Owen Rogers, using the UK Office for National Statistics' database of Non-Seasonally Adjusted Index of Sales, collated these data [1]. Figures 3, 4, 5 and 6 displays the four demand profiles that we label using R&C's terminology: *Rapid Growth* (Figure 3), *Steady Growth* (Figure 4), *Recession & Recovery* (Figure 5) and *Steady* (Figure 6). These data were kindly supplied to us by Owen Rogers to enable us to perform a strict replication of R&C's experiments. As Rogers and Cliff explain, since historical demand data for cloud resources over long periods (necessary for the brokerage model) are not available (due to the relatively short history of cloud computing), public domain data were selected for markets that can be considered *complementary* to cloud computing—for example e-commerce applications, where demand for computing resources is positively related to sales executed. Furthermore, to enable the brokerage model to be evaluated across a range of market conditions, markets exhibiting qualitative variation (e.g., growth versus recession) were preferentially selected. As one may expect, all demand profiles exhibit strong seasonality, including significant spikes around Christmas.
- **Running time:** Each simulation lasts 276 simulated months. This time period is determined by the available demand data utilised by R&C (refer to Figures 3, 4, 5 and 6). Results statistics are all calculated up to and including 2010.
- **Number of user agents:** Following R&C, we set the number of user agents in the population to $|A| = 1000$.
- **RI Period and learning period length:** Following R&C, we explore RIs with both $R = 12$ and $R = 36$ months. At the start of each simulation, there is a *learning* period of $R$ months such that the broker observes the market and updates historical demand, but does not trade.
- **Pricing:** Prices for cloud computing resources in the real world continue to fall as providers face increasing competition. For the R&C replication experiments, we follow the same pricing scheme as [1], shown in Table 1. We take these values directly from the source code used by R&C for their brokerage model; kindly made available to us by Owen Rogers. The values in Table 1 differ slightly from those described in [1]; which states that the monthly cost of an on-demand instance is *"approximately $60 for a whole month of usage"* page 5, [1], and does not give absolute values for the monthly usage cost of reserved instances. In the final experiment (Section 'Experiment 5: Recalibrating R&C's model to reflect current market

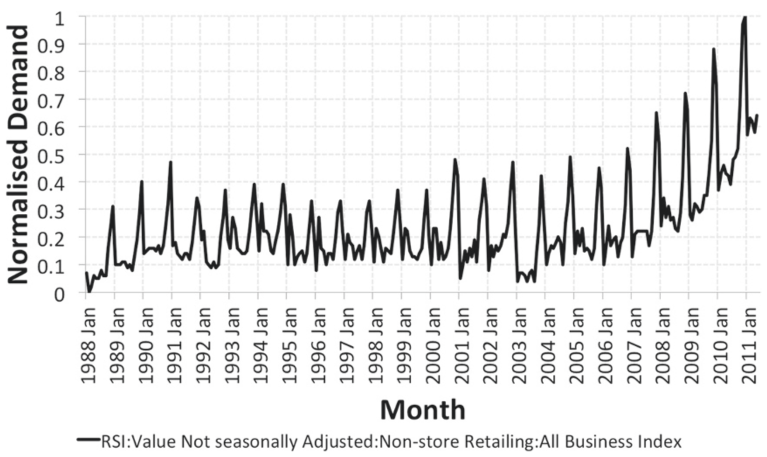

**Figure 3 Normalised demand profiles for the period 1988-2011, labelled: *Rapid Growth*.** RSI: Not seasonally adjusted, non-store retailing, all business index. (Data identical to Figure one, [1]).

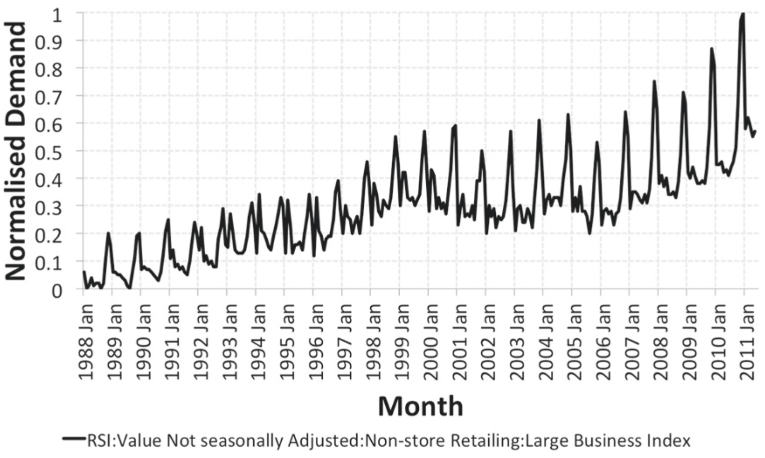

**Figure 4 Normalised demand profiles for the period 1988–2011, labelled: *Steady Growth*.** RSI: Not seasonally adjusted, non-store retailing, large business index. (Data identical to [1], Figure two).

conditions'), we recalibrate the model using current AWS prices (September 2013) and observe the effect of these changes on the brokerage model.

- **Cost Factor, *C*:** The WZH charging model [11] is based on instances with a cost of 1 or 2. The model must therefore be scaled by a cost factor, *C*, in order to simulate real-world pricing (refer to Equations (3) and (4)). To replicate R&C we use a cost factor of $C = 35$. However, in Section 'Experiment 5: Recalibrating R&C's model to reflect current market conditions', when AWS prices are updated to represent current prices, we vary the value of *C* to ensure the model still benefits users.
- **Minimum MRU threshold, $\theta$:** Following R&C, we explore a range of thresholds, $\theta$, to determine the optimal (most profitable) value, $\theta_{opt}$. In each experiment, $\theta$ is varied between 0 and 1 using increments of 0.01.

In this paper, we repeat each experimental condition only once. Since the broker follows a deterministic procedure (see Algorithm 1), and since demand is fixed during each run (see Figures 3, 4, 5 and 6), there is very little stochastic variation in the model; generated only by the random ordering of user agents selected to *purchase* and *exercise* options each month. Previous studies that performed 30 repeated trials of each condition to enable statistical hypothesis testing of the results identified that 95% confidence intervals were negligible (usually narrower than the thickness of the line presented in the graph of results; see, for instance Figure two, [17]). Hence, the brokerage model of R&C is largely deterministic and

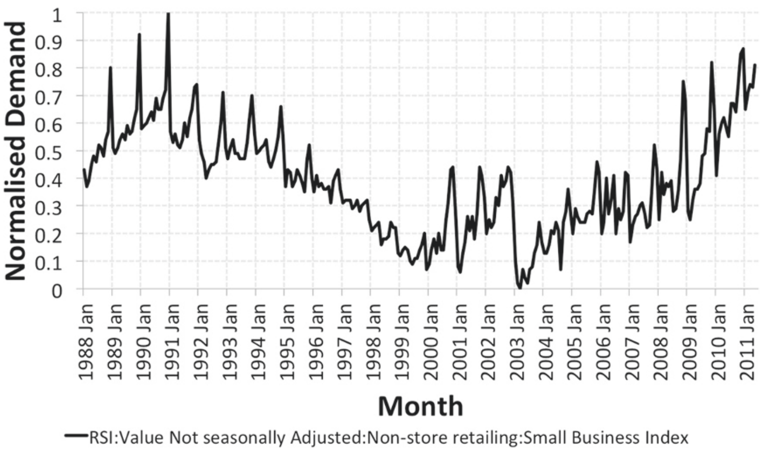

**Figure 5 Normalised demand profiles for the period 1988–2011, labelled: *Recession & Recovery*.** RSI: Not seasonally adjusted, non-store retailing, small business index. (Data identical to [1], Figure three).

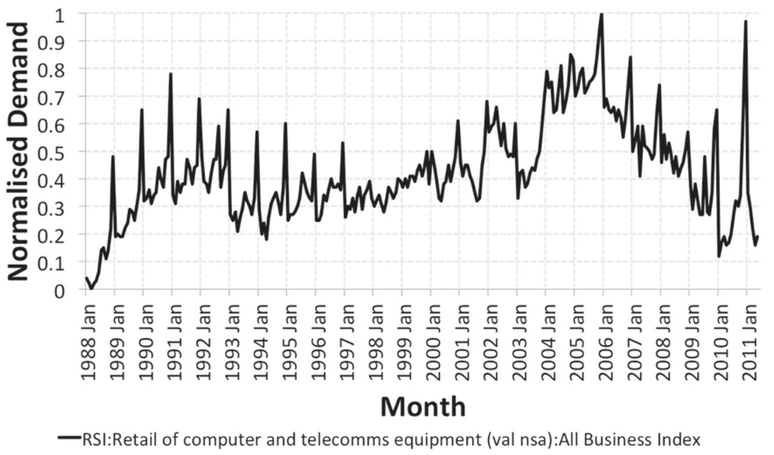

**Figure 6 Normalised demand profiles for the period 1988-2011, labelled: *Steady*.** RSI: Not seasonally adjusted, retail of computer and telecoms equipment, all business index. (Data identical to [1] Figure four).

we expect to replicate the results presented in [1] very closely.

## Experimental results

In this section, we present a series of experimental results using our CReST replication of R&C's brokerage model. In Experiment 1 (Section 'Experiment 1: Model verification'), we present results that closely match those presented by R&C [1]. However, to generate these results, it is necessary to replicate two problems identified in R&C's implementation: the *reservations bug*, rectified in Experiment 2 (Section 'Experiment 2: the reservations bug'); and the *payment bug*, rectified in Experiment 3 (Section 'Experiment 3: the payment bug'). In Experiment 4 (Section 'Experiment 4: corrected replication of R&C's brokerage model'), we present a "corrected" version of R&C's results, without the reservations and payment bugs. Results are compared with those published in [1]. Finally, in Experiment 5 (Section 'Experiment 5: Recalibrating R&C's model to reflect current market conditions'), we recalibrate the model using current AWS prices and updated constraints to reflect the introduction of a secondary marketplace for users to buy and sell RIs between themselves. Results demonstrate that the brokerage model is no longer profitable.

**Table 1 Instance prices**

| Linux/Unix Standard RI | | 12-months | | 36-months | |
|---|---|---|---|---|---|
| Reserved up-front | $P_{Ru}$ | \$227.50 | | \$350.00 | |
| | | Hourly | Monthly | Hourly | Monthly |
| Reserved usage | $P_{Rm}$ | \$0.03 | \$21.88 | \$0.02 | \$21.88 |
| On-demand usage | $P_D$ | \$0.085 | \$62.00 | \$0.085 | \$62.00 |

AWS prices used for R&C replication studies (AWS prices in June 2011).

### Experiment 1: Model verification

Here, we present results that closely replicate those presented by R&C [1]. Figures 7 and 8 show total broker profits as a function of $\theta$ for 12-months and 36-months RIs, respectively. These graphs are directly comparable to Figure six, [1] for 12-months RIs and [1], Figure seven for 36-months RIs. To the naked eye, these results are *qualitatively identical* to those presented by R&C; and also *quantitatively very similar*, with broker profits across all markets falling within 2% (the majority within 1%) of R&C's published values [1], Table one.

These results offer compelling evidence that our CReST implementation is an *accurate replication* of R&C's brokerage implementation. However, it was *not possible* to achieve these results by strictly following the brokerage model. Our failure to replicate R&C's results led us to scrutinise R&C's source code, to which we were kindly given access by Owen Rogers. During this process, we noticed two inaccuracies in R&C's implementation—the reservations bug and the payment bug. *Only* by replicating these bugs in our CReST model were we able to produce the results (presented in Figures 7 and 8) that closely match those of R&C. This verifies that the reservation and payment bugs were present in R&C's model implementation and that the headline results presented by R&C are *inaccurate*. Given the significant interest that R&C's results have generated within the cloud computing community, this is an important discovery. In the following sections, we demonstrate the impact that the reservations and payment bugs have on R&C's results.

### Experiment 2: the reservations bug

In R&C's implementation of the brokerage model, there is a bug in the software code that causes the broker to purchase fewer reservations than the model suggests. This

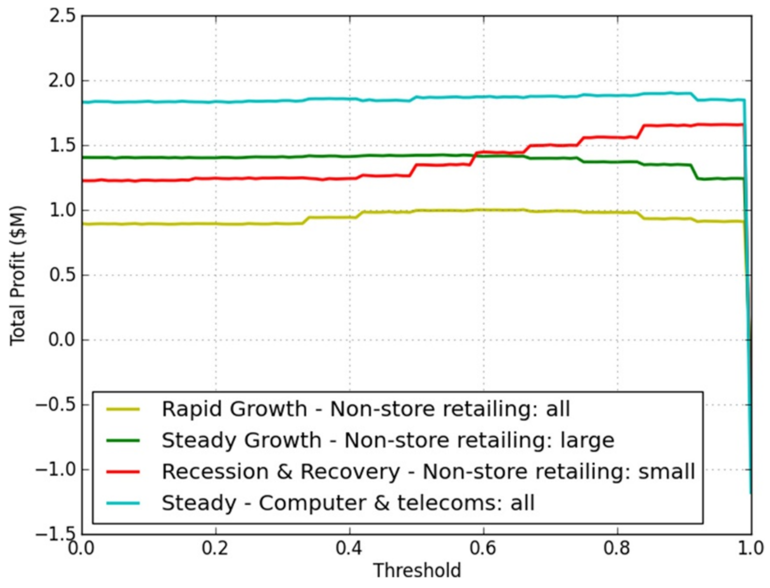

**Figure 7 Replication.** Total profits plotted against $\theta$ using 12-months reserved instances (compare Figure six, [1]).

occurs during the while loop in the *PurchaseReservedInstances* function detailed in Algorithm 1 (see Section 'Broker'). Each time the broker decides to purchase a new reserved instance, i.e., when "*if MRU* $\geq \theta$" is *True* (line 5), the broker then allocates this new purchase to the *next* unit of demand. In essence, there is an additional line in Algorithm 1, between lines 8 and 9, which reads:

$$\texttt{line 8a:} \qquad \sum p_i \leftarrow \sum p_i - 1 \qquad (9)$$

*This is the reservations bug*—each new RI purchase is effectively allocated to *two* units of demand. As a result, the broker purchases fewer RIs than the brokerage model dictates. (For further discussion of the reservations bug, refer to pp. 26-33, [18]).

We can see the effect of removing the reservations bug from our CReST implementation in Figures 9 and 10. For 12-months RIs, the effect is most clear (Figure 9). We see that for all markets, the total profit for brokers is much lower when $\theta < 0.5$. In particular, in the *Recession*

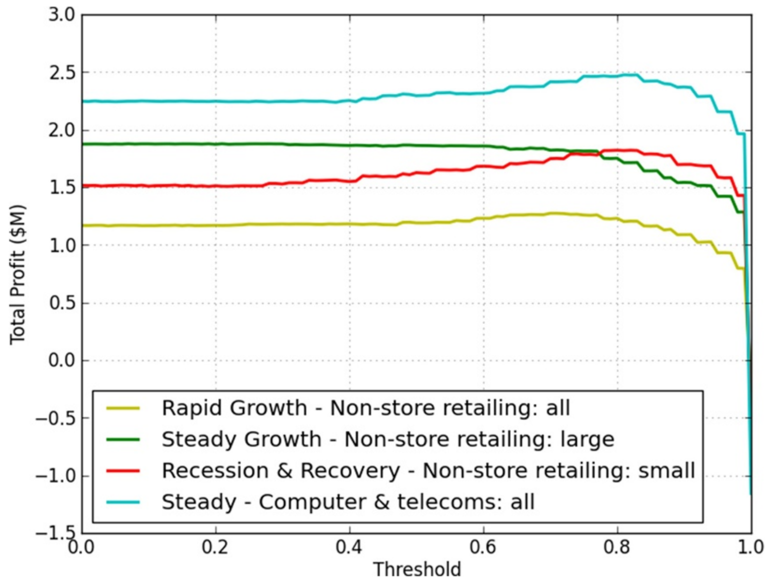

**Figure 8 Replication.** Total profits plotted against $\theta$ using 36-months reserved instances (compare Figure seven, [1]).

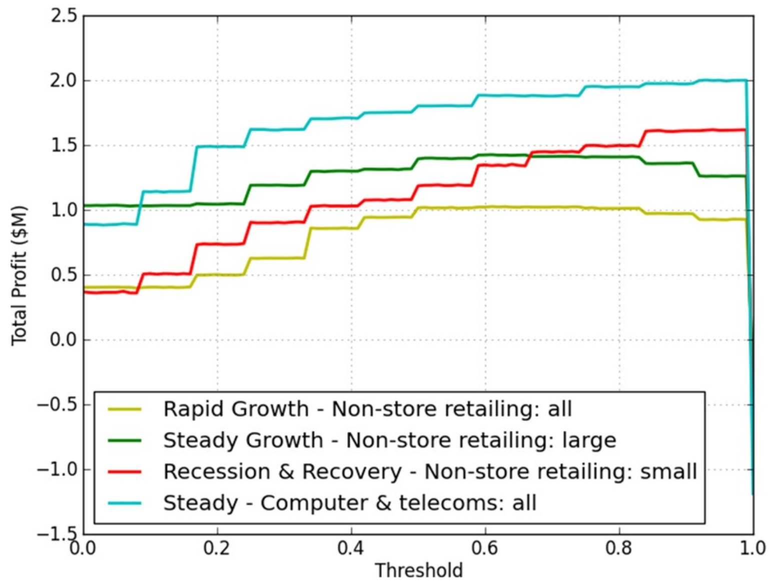

**Figure 9 No reservations bug.** Total profits plotted against $\theta$ using 12-months reserved instances (compare Figure 7).

*& Recovery* market, when $\theta = 0$, broker's profits are 70% lower than R&C report on 12-months RIs. A similar, but smaller effect is also observed for 36-months RIs (Figure 10).

This result can be explained as follows. As the value of $\theta$ falls, the statement "*if MRU*$\geq \theta$" (line 5) is more likely to evaluate *True*. Therefore, smaller values of $\theta$ are more affected by the reservations bug than larger values of $\theta$. When the reservations bug is *not* present, the broker purchases *more* RIs per unit of expected demand. However, since under-utilised RIs reduce the broker's profits (and potentially generate a *loss*), we see profits most affected when $\theta$ is small (i.e., when the broker takes on most risk from purchasing RIs).

We thus conclude that the reservations bug has artificially inflated the brokerage profits presented by *R&C*. This effect increases as the value of $\theta$ falls and is greatest when $\theta < 0.5$.

## Experiment 3: the payment bug

Algorithm 2 describes the process used to calculate the monthly usage fees a broker must pay to satisfy user

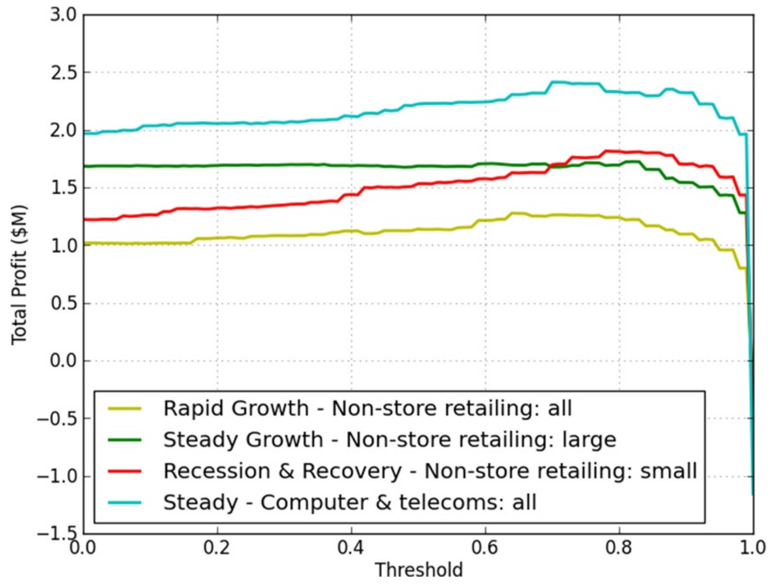

**Figure 10 No reservations bug.** Total profits plotted against $\theta$ using 36-months reserved instances (compare Figure 8).

demand. Each month, the number of options exercised by users is the *demand* that the broker must satisfy. The broker compares demand against the number of RIs owned—the current *capacity* (Algorithm 2, line 3). If capacity covers demand, then the broker simply pays the provider the monthly fee to utilise enough RIs to fulfil demand (Algorithm 2, line 4). Else, if capacity is less than demand, the broker first pays to utilise all RIs owned (Algorithm 2, line 6), before covering excess demand by purchasing additional on-demand instances from the provider (Algorithm 2, lines 7-8).

**Algorithm 2** The payment bug—line 6 is missing from R&C's implementation.

```
1:  function CalculateUsageFees(capacity, demand)
2:      fees ← 0
3:      if capacity ≥ demand then          ▷ if broker owns enough reserved instances to cover demand
4:          fees ← demand × $P_{Rm}$          ▷ pay monthly fee for reserved instances used
5:      else
6:          fees ← capacity × $P_{Rm}$        ▷ pay monthly fee for reserved instances used
7:          excessDemand ← demand − capacity
8:          fees ← fees + excessDemand × $P_D$   ▷ purchase on-demand instances to cover excess
9:      end if
10:     return fees
11: end function
```

From an inspection of R&C's code, it is apparent that R&C have erroneously implemented the procedure described in Algorithm 2 to calculate monthly utilisation fees that the broker must pay the provider. In particular, R&C's code has *accidentally omitted line 6. This is the payment bug*—during months that demand is greater than capacity, the broker *does not pay* the monthly utilisation charge for RIs. Although the payment bug only manifests when there is excess demand for broker resources (and hence is more likely to occur when $\theta$ has a high value), it has a significant effect on results. (For further discussion of the payment bug, refer to [18]).

Figures 11 and 12 present results from our CReST replication of R&C's model with the payment bug corrected (to ensure a controlled comparison, the reservations bug remains). We see that the broker's profits are much lower across all thresholds, $\theta$, and in all markets. Comparing Figure 11 with Figure 7, we see that, for 12-months RIs, maximum profits ($\theta = \theta_{opt}$) fall by more than 50% in all markets (and more than 75% in the *Steady* market). Likewise, comparing Figure 12 with Figure 8, we see that, for 36-months RIs, maximum profits fall by more than 33% in all markets (and more than 50% in the *Steady* market). Clearly, the payment bug is responsible for a large inflation in the brokerage profits reported by R&C.

### Experiment 4: corrected replication of R&C's brokerage model

Here, we perform a fully "corrected" replication of R&C's model with *both bugs removed*. Figures 13 and 14 once again show total profits against $\theta$. It can be seen that total profits are *significantly lower* than those presented by R&C. However, the broker remains profitable under all conditions other than when investing in 12-months RIs in a *Recession* & *Recovery* market using a low *MRU* threshold, $\theta \leq 0.07$. Under these conditions, the broker invests too much capital in purchasing RIs that are then under-utilised by users.

Table 2 presents a summary of results. This table is intended to replace the results table presented by R&C Table one, [1]. In general, the results in Table 2 largely support the conclusions drawn by R&C. Apart from the particular loss-making conditions described above, the broker profits under all other conditions and will increase profits: by considering past performance (i.e., by selecting a value of $\theta > 0$); and by investing in 36-months RIs rather than 12-months RIs. Therefore, although the reservations and payment bugs have a significant impact on *absolute profits* (the corrected model produces between 31% and 68% *less* maximum profit than the original model; see Table 3), the headline results presented by R&C largely hold true in the corrected model.

However, in the following section, we recalibrate R&C's brokerage model using current market conditions (September 2013) to test whether the conclusions are still valid.

### Experiment 5: Recalibrating R&C's model to reflect current market conditions

Since the publication of R&C's brokerage model in April 2012 [1], the market for cloud computing provision has become increasingly competitive. Over the last 18 months, AWS' prices have fallen rapidly (compare Table 4 with Table 1); and new RI price tiers have been introduced (*"Light"*, *"Medium"* and *"Heavy"* Utilisation), enabling users more flexibility to offset upfront payment, $P_{Ru}$, against the monthly utilisation rate, $P_{Rm}$.

Perhaps more significantly, AWS have also introduced a *secondary* marketplace venue for cloud users to buy and sell RIs—the *Amazon EC2 Reserved Instance Marketplace* (ARIM) [19,20]. ARIM enables cloud users that own RIs to re-sell the time remaining (in whole months). For instance, if a user with 9 months and 2 weeks remaining on a 12-months RI decides that the RI is surplus to requirements, the instance can be resold on ARIM with a reservation period of 9 months. Sellers are able to offer

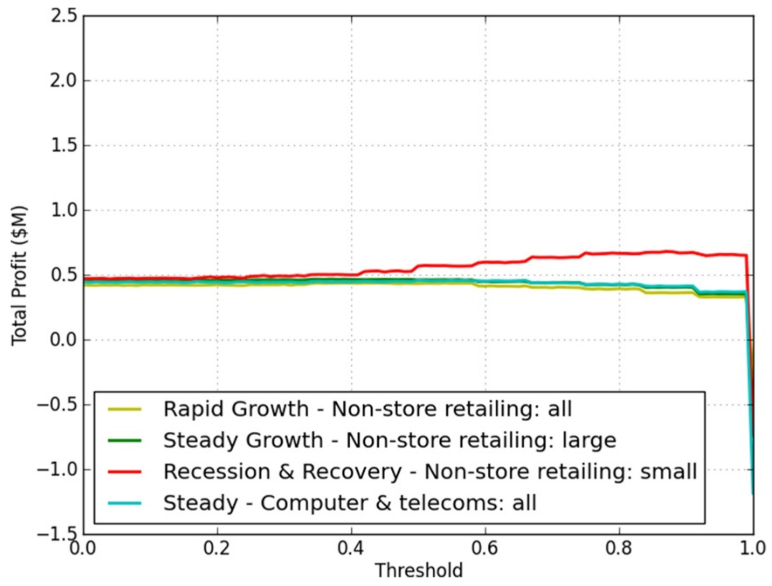

**Figure 11 No payment bug.** Total profits plotted against $\theta$ using 12-months reserved instances (compare Figure 7). For values of $\theta \leq 0.4$, profits are similar (non-identical) for all markets. This is purely co-incidental.

their RIs for sale at *any* price, $P_A$, but are encouraged by default to use a *"linear price drop"*, such that:

$$P_A = \left(\frac{r}{R}\right) P_{Ru}, \quad (10)$$

where $r$ is the time in whole months remaining on the instance, $R$ is the RI's full term (in months), and $P_{Ru}$ is the price of the instance when bought directly from the provider [19].

ARIM has the potential to radically disrupt the market for RIs. Firstly, the introduction of a re-sale venue means that investment in RI purchases on the *primary* market—i.e., directly from the provider—are less risky, as unwanted time can be traded away. Secondly, the availability of RIs on the secondary market with contract periods shorter than those available on the primary market (since $r \leq R$) produces a further reduction in the purchase risk of RIs. Thus, by enabling the trade of risk, ARIM encourages users to switch from the (more expensive, but less risky)

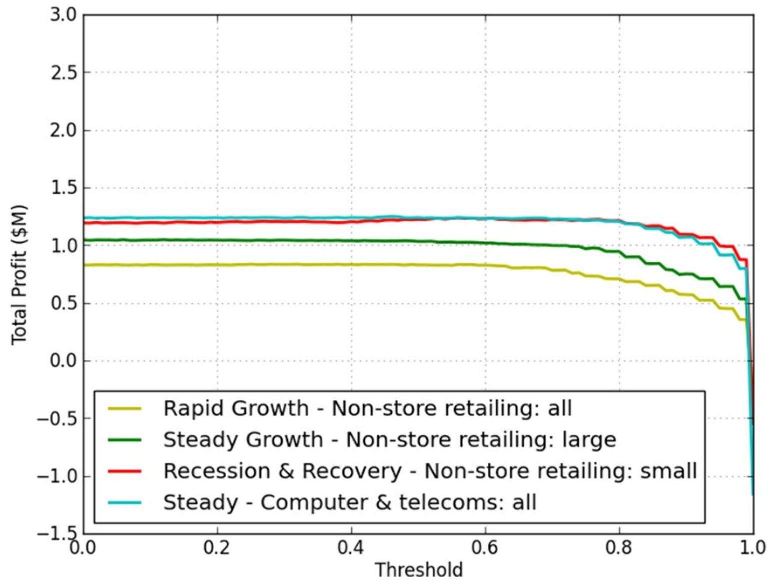

**Figure 12 No payment bug.** Total profits plotted against $\theta$ using 36-months reserved instances (compare Figure 8).

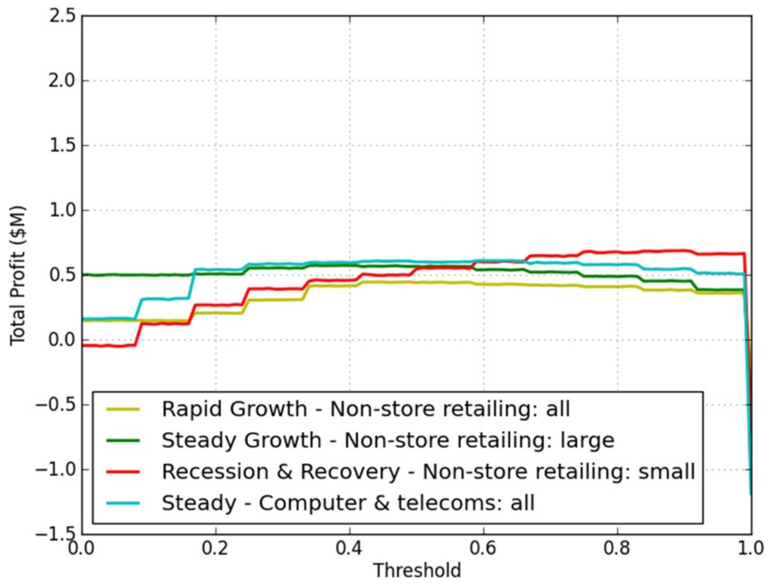

**Figure 13 Corrected model.** Total profits plotted against $\theta$ using 12-months reserved instances (compare Figure 7).

on-demand purchase model, to the RI purchase model. This benefits AWS, as an increase in the sale of RIs on the primary market generates more predictability (in revenue and demand) for the provider and encourages user loyalty.

In light of these changes, it is necessary to *recalibrate* the brokerage model of R&C and address assumptions of the model that may no longer hold.

### *Current AWS pricing (September 2013)*

Table 4 presents current AWS prices for a *Medium Utilisation* RI (prices assume 730.5 hours/month; 365.25 days/year). The new tiered pricing of RIs by AWS means that users can select to purchase RIs based on their likely utilisation. Higher utilisation RIs have a higher up-front cost, $P_{Ru}$, and lower usage cost, $P_{Rm}$, than lower utilisation RIs. Here, we select Medium Utilisation RIs as a simple proxy for *average* prices, however, the conclusions drawn are largely unaffected by this choice. We can see from Table 4 that all prices have fallen roughly 30% since April 2012 (compare Table 1). In this section, we use these new AWS prices to test the profitability of the brokerage model in the current market. However, when calibrating

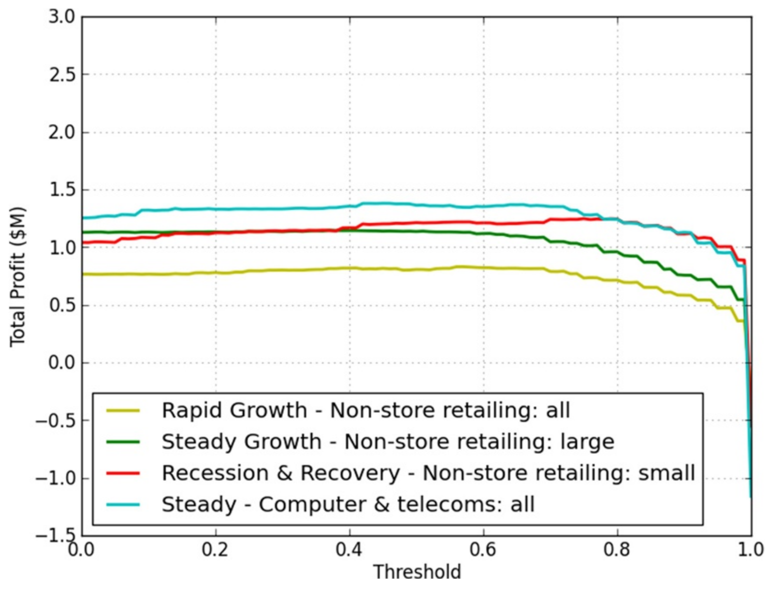

**Figure 14 Corrected model.** Total profits plotted against $\theta$ using 36-months reserved instances (compare Figure 8).

**Table 2 Profits (and increase) achieved for $\theta = 0$ and $\theta = \theta_{opt}$ (compare [1], Table one)**

| | 12-months | | | | 36-months | | | |
|---|---|---|---|---|---|---|---|---|
| | Profit $M | | | | Profit $M | | | |
| | $\theta = 0$ | $\theta = \theta_{opt}$ | Change | $\theta_{opt}$ | $\theta = 0$ | $\theta = \theta_o pt$ | Change | $\theta_{opt}$ |
| Rapid growth | 0.15 | 0.44 | 202% | 0.42-0.49 | 0.76 | 0.83 | 8% | 0.56-0.58 |
| Steady growth | 0.50 | 0.57 | 15% | 0.34-0.41 | 1.13 | 1.15 | 2% | 0.37-0.41 |
| R & R | −0.05 | 0.69 | −1565% | 0.84-0.91 | 1.0 | 1.25 | 19.8% | 0.71-0.80 |
| Steady | 0.16 | 0.61 | 279% | 0.59-0.64 | 1.3 | 1.4 | 10% | 0.42-0.46 |

the model with real-world prices, it is necessary to adjust the cost factor, $C$, to ensure that the options prices that the broker offers remain low enough to benefit users (and hence ensure that each user, $i$, estimates their probability of exercising an option, $p_i$, truthfully). In the following section we describe how we derive $C$, given the current market landscape.

### *Cost factor*

The pricing structure of the brokerage model is defined as:

$$ModelPrice(p_i) = p_i\ TotalPrice(p_i) + (1 - p_i)\ OptionPrice(p_i), \tag{11}$$

where $TotalPrice(p_i)$ and $OptionPrice(p_i)$ are given by equations (5) and (3), respectively. For the model to offer value to users (a requirement necessary to remain within the WZH bounds for truth-telling conditions [11]), $ModelPrice(p_i)$ must be lower than (is bounded by) the lowest available price on the market, $P_M$, such that:

$$ModelPrice(p_i) < p_i P_M, \quad \forall p_i \in [0, 1]\,. \tag{12}$$

Then, $C_{max}$ is the value of $C$ that maximises broker's profits, i.e., the highest value of $C$ that satisfies inequality (12).

In [1], R&C assume that $P_M$ is the price of an on-demand instance purchased directly from the provider, $P_D$. Using current AWS prices (Table 4), we have:

$$P_D = \$43.83. \tag{13}$$

When using R&C bounds, the maximum cost factor $C_{max} = 25$ (see Figure 15; where the blue-dashed line is the upper-bounds on the model, $p_i P_D$, and the solid blue line represents $ModelPrice(p_i)$ when $C = 25$). Thus, when $C = 25$, the user is incentivised to purchase options rather than purchasing on-demand.

However, since the publication of [1], the introduction of ARIM means that users can now purchase instances on the secondary market (at price $P_S$). We can be sure that one month RIs offered for sale on ARIM will have price $P_S < P_D$, since RIs inherently offer less flexibility than on-demand instances. Therefore, the assumption that $P_D$ is the best available price is no longer reasonable—we must now consider $P_S$ as the best available price.

We can estimate the likely value of $P_S$ using simple game-theoretic reasoning. Sellers on ARIM want to maximise price subject to the constraint $P_S < P_D$. However, sellers must offer a *competitive* price to ensure a sale. If $P_S > P_C$, where $P_C$ is the cost price of a RI to the seller, then there exists an opportunity for a professional re-seller to enter the market at price $P_B$ and profit from simply buying long-term RIs from the provider, slicing into monthly units, and re-selling at a price $P_C < P_B < P_S$. In the limit, therefore, we expect ARIM to tend to an equilibrium price such that: $\lim_{t\to\infty} P_S = P_C$. At prices $P_S < P_C$, the seller makes a loss on each sale. This is clearly not a profitable strategy to pursue, and hence professional re-sellers on ARIM will not price at these values. However, there may be individual firms that attempt to dump surplus RI volume at a price $P_S < P_C$ to ensure a sale. But, assuming that ARIM is a *liquid* market—i.e., demand on ARIM is high enough to ensure that a competitive sale price will always result in a sale—then there is no need for a seller to

**Table 3 Overall deviations of maximum profits between R&C's model and corrected model**

| | 12-months | | | 36-months | | |
|---|---|---|---|---|---|---|
| | Maximum profit $M | | | Maximum profit $M | | |
| | R&C | Corrected | Difference | R&C | Corrected | Difference |
| Rapid growth | 1.00 | 0.44 | −56% | 1.27 | 0.83 | −35% |
| Steady growth | 1.41 | 0.57 | −59% | 1.85 | 1.15 | −38% |
| R & R | 1.65 | 0.69 | −58% | 1.80 | 1.25 | −31% |
| Steady | 1.89 | 0.61 | −68% | 2.45 | 1.38 | −44% |

**Table 4 Current AWS prices for a Medium Utilisation RI (September 2013)**

| Medium Utilization RI (m1.small) | | 12-months | | 36-months | |
|---|---|---|---|---|---|
| Reserved up-front | $P_{Ru}$ | $139.00 | | $215.00 | |
| | | Hourly | Monthly | Hourly | Monthly |
| Reserved usage | $P_{Rm}$ | $0.021 | $15.34 | $0.017 | $12.42 |
| On-demand usage | $P_D$ | $0.060 | $43.83 | $0.060 | $43.83 |

reduce prices below $P_C$ unless the volume to be sold forms a significantly large proportion of the market (at which point there will be an adverse "price impact", such that $P_S$ must be lowered in order to *discover* enough demand to fulfil the volume [21]). For simplicity, we assume that ARIM is a liquid market and ignore commission costs (ARIM charges a 12% commission fee on all sales [19]). Then, we expect:

$$P_S = P_C = \frac{P_{Ru}}{R} + P_{Rm} = \$18.39. \tag{14}$$

We consider $P_S$ to be the new *strong bounds* on the brokerage model (see Figure 15, green dashed line). When $C_{max} = 10$, broker's profits are maximised (Figure 15, green solid line), while ensuring that users are incentivised to purchase options from the broker rather than purchase RIs on ARIM.

Finally, we also consider a *weak bounds* on the model, such that $P_M = P_W$, where:

$$P_W = \frac{P_D + P_S}{2} = \$31.11. \tag{15}$$

$P_W$ assumes that RIs are available to users on ARIM only half of the time. We introduce this as an approximation of ARIM being an *illiquid* market, such that supply cannot always cover demand. Hence, under these conditions, users cannot always be certain that a RI will be available for purchase on ARIM. Under this condition, $C_{max} = 18$ (Figure 15, red line).

Although we expect the strong bounds to be a more realistic expectation of the long-term equilibrium price on ARIM, we include the weak bounds as a *transitional* model of ARIM, since it is likely to take some time before ARIM gains enough traction to become a liquid market trading at equilibrium.

### *Empirical evidence of instance prices from ARIM data*

At the time of writing (March 2014), ARIM has been operational for 18 months. However, since Amazon AWS do not currently publish historical ARIM trade data it is impossible to perform a full analysis of the behaviour of the market. Therefore, to determine the level of RI prices available on ARIM, a snapshot of current price data was collected across all instance types and all availability zones in US-East on 14th March 2014, 13:14 GMT. To normalise the data across instance types, the effective hourly rate (EHR) for each instance, defined as:

$$\textit{Effective hourly rate (EHR)} = \frac{\textit{upfront cost}}{\textit{term in hours}} + \textit{hourly rate}, \tag{16}$$

was indexed against the effective hourly rate of the original full-term instance when purchased from AWS, such that:

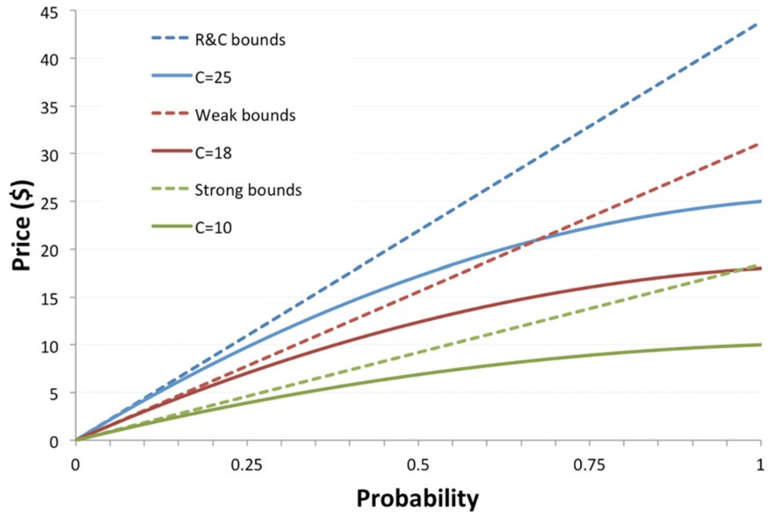

**Figure 15 Cost factor bounds.** Maximum cost factor, *C*, given upper bounds on options pricing.

$$EHR\ index = \frac{EHR\ of\ instance\ listed\ on\ ARIM}{EHR\ of\ instance\ purchased\ from\ AWS}. \tag{17}$$

Therefore, when EHR index = 1, the ARIM list price has the same effective hourly rate as the same instance purchased directly from the provider—this is also equivalent to the linear drop price of the instance, defined by equation (10). Index values below 1 indicate that ARIM prices are effectively lower than provider prices, and values above 1 are effectively higher than provider prices.

Figure 16 represents a snapshot of all EHR index prices across all instance types against the proportion of time remaining on the RI term. Each bubble plots an individual quote price, with bubble area proportional to volume (number of instances) listed at that price. It can clearly be seen that the majority of instances are quoted for sale at, or just below, EHR index = 1. Therefore, the argument presented previously in Section 'Cost factor'—that prices on ARIM are likely to be available at the linear drop price—appears valid. This supports the strong assumption, $P_S$ (14), which implies a cost factor $C_{max} = 10$.

However, a recent blog article presents analysis concerning the liquidity of ARIM [22]. The author monitors ARIM over a 45-day period between 28th November 2013 and 9th January 2014, taking hourly snapshots of the marketplace. After cleaning the data, the author uses survival function estimates to calculate that the median time-to-sell (for instances that eventually do sell) is 15.5 days and the censoring ratio is 30% (the proportion of instances unsold at the end of the observed interval) [22]. This analysis suggests that ARIM is still an illiquid and fragmented market, thus supporting the weak assumption, $P_W$ (15), which implies a cost factor $C_{max} = 18$.

The empirical evidence presented here supports the argument proposed in Section 'Cost factor'. Specifically, while the strong bounds offer a more realistic expectation of the long-term equilibrium of ARIM, the weak bounds are likely to be a more accurate transitional model of the market's current illiquidity. This argument is further supported by evidence generated from a multi-agent simulation of ARIM [23], using a simple model containing a population of ZIP trading agents buying and selling RIs on a simulated exchange (see [24] for ZIP's design and [25-27] for a review of ZIP). Results from the model showed that simulated trade prices tend to equilibrate at the linear drop price when the market is liquid; and equilibrate below the level required for a broker to profit when there is an excess of sellers dumping unwanted capacity into the market [23].

### Results

When running the brokerage model using current AWS prices (Table 4) and a cost factor $C = 25$, the brokerage model remains profitable (at a level similar to that

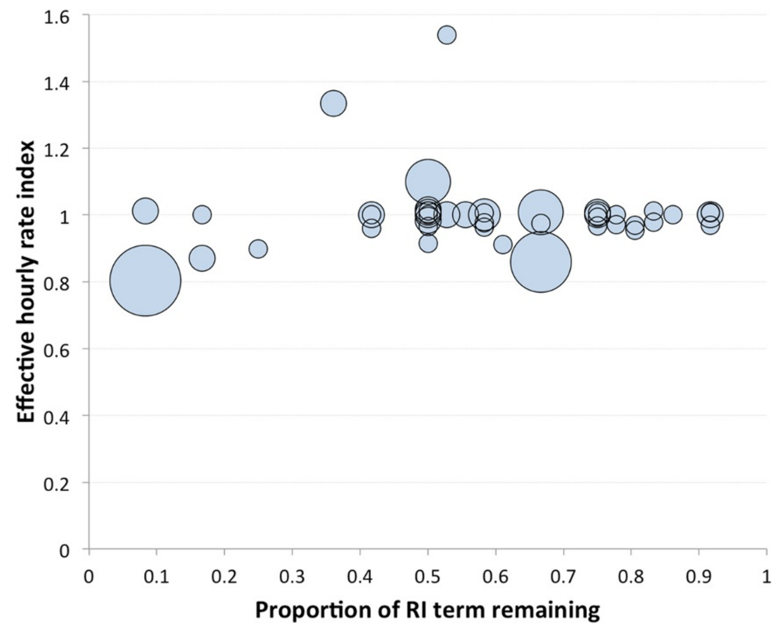

**Figure 16 Snapshot of US-East ARIM quote prices for all instance types, 14th March 2014 13:14 GMT, with bubble area proportional to liquidity (volume) at a specific price.** An effective hourly rate index of 1 indicates that an instance is listed at the linear drop price. Values below (above) 1 indicate instance is listed at a price below (above) the linear drop price. Clearly, the majority of instances are listed at or below the linear drop price (index $\leq$ 1).

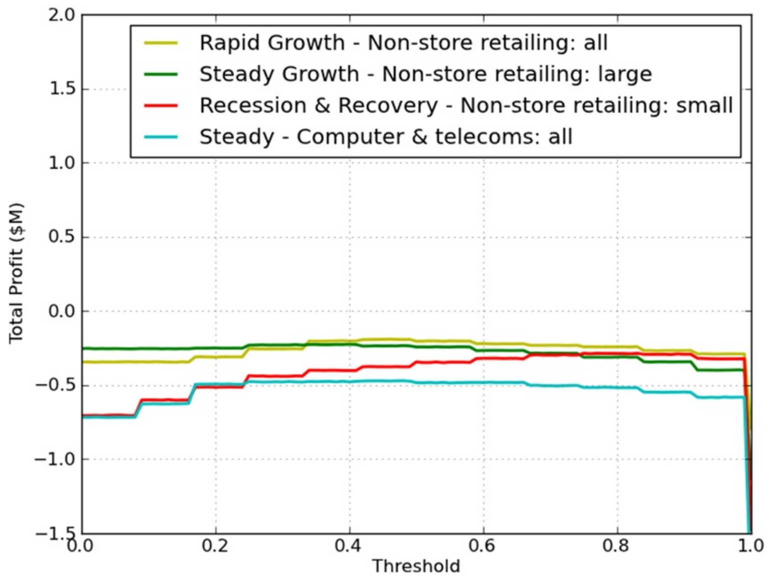

**Figure 17 September 2013 calibration (C=18).** Total profits plotted against $\theta$ using 12-months reserved instances.

presented in Figures 13 and 14). However, $C = 25$ is only acceptable if we assume R&C's upper bounds (Figure 15). Given the introduction of ARIM, we reject this result and re-test the brokerage model using cost factors of $C = 18$ (weak assumption) and $C = 10$ (strong assumption).

Figure 17 shows total broker profits for 12-months RIs when $C = 18$. For all market conditions, the broker makes a loss. This demonstrates that, under the weak assumption, the brokerage model is no longer profitable for 12-months RIs. However, the model *does* remain profitable for 36-months RIs, but at a much reduced rate (maximum profit is less than \$0.4M). Hence, as ARIM grows in popularity and becomes more liquid, we will expect the brokerage model to be much less profitable.

Finally, when running the model using $C = 10$, 36-months RIs are *no longer profitable* (see Figure 18). Thus, under the strong assumption, the brokerage model is no longer profitable *under any condition*. This effectively

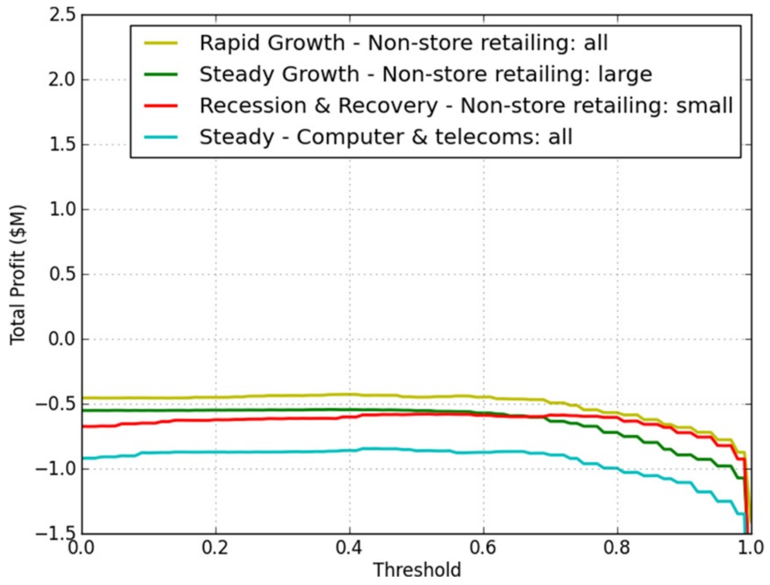

**Figure 18 September 2013 calibration (C=10).** Total profits plotted against $\theta$ using 36-months reserved instances.

demonstrates that, once ARIM matures into a liquid market, there will be no profit available for a broker using R&C's model. Thus, we conclude that (if it hasn't already) the window of opportunity to commercially exploit the brokerage model of R&C will soon close.

### *Discussion: the effect of broker competition*

Throughout this paper, and following directly from R&C's brokerage model, it has been assumed that the broker has a monopoly position. Indeed, the WZH reservation model, which R&C's brokerage model extends, is described by the authors as a mechanism that *"extracts revenue similar to that of a monopoly provider practicing temporal pricing discrimination with a user population whose preference distribution is known in advance"* [11]). Thus, the brokerage model relies on the broker being somehow able to position itself as a monopoly provider of options; perhaps through an exclusive licensing agreement with the provider.

We have demonstrated in Section 'Results' that R&C's corrected brokerage model is profitable under current AWS prices when acting as a monopoly provider of options. However, we have also shown that this profit is eroded through the availability of cheaper instance prices on ARIM. What we have not considered is the impact that competition between multiple brokers may have on profit.

It has been shown that when competition is introduced into a market dominated by a monopolistic supplier performing differential pricing—similar to R&C's brokerage model—there will be an increase in intra-firm price dispersion [28]. That is, the difference between the highest and lowest prices charged by each individual firm will increase: *"as the market becomes more competitive, prices become more dispersed, a pattern documented in the airline industry"* [28]. In the airline industry for example, one can observe that while competition between airlines has lowered the price of the cheapest economy seats (booked off-season, non-refundable, non-transferable, booked in advance), the most expensive seats (first or business class, flexible dates, refundable) have experienced rapid price inflation.

Therefore, in a market containing multiple brokers using R&C's brokerage model we would expect competition between brokers to similarly increase intra-firm price dispersion. Therefore, either: (i) the minimum price (when the probability of an individual consumer exercising the option $p_i = 1$) will fall; or (ii) the maximum price ($p_i = 0$) will rise; or (iii) both the minimum price will fall and the maximum price will rise together. However, since the maximum price is constrained by the cost factor, $C$, it is impossible for the maximum price to increase beyond the bounds determined by $C_{max}$ (otherwise the brokerage model will no longer encourage consumers to act truthfully, thus destroying the model). Hence, for competition to occur, either the minimum price ($p_i = 1$) must fall, or the cost factor must fall. Either way, since the maximum price ($p_i = 0$) cannot simultaneously rise, then the brokers' margins (and ultimately, the brokers' profits) must fall. In reality, the dynamics of a competitive brokerage market, resulting from the coevolutionary interaction of multiple adapting entities, are likely to be complex (see, for example [29,30]). However, we fundamentally expect a downward trend in brokerage profits over the long-term.

In summary, therefore, we reach the intuitive conclusion that competition between brokers will reduce brokers' profits. Furthermore, this effect will be exacerbated by the availability of "cheap" instances on the secondary market, ARIM. Thus, unless the cloud provider acts as a monopoly brokerage supplying instance options, R&C's brokerage model looks unsustainable in the face of potential competition. That is not to say that profitable opportunities do not exist via alternative differential pricing models using temporal demand forecasts (for example [31,32]), or alternative cloud brokerage models.

## Conclusions

The financial brokerage model for cloud computing presented by Rogers & Cliff (R&C) demonstrated that there is an opportunity to profit from acting as an intermediary between cloud providers and cloud users. This result has received significant attention within the cloud computing community and the publication is currently the fifth most accessed article of all time in the *Journal of Cloud Computing: Advances, Systems and Applications* [2].

Here, we have presented results from our independent replication of R&C's brokerage model using CReST, the Cloud Research Simulation Toolkit. In doing so, we have identified two problems with R&C's implementation: the *reservations bug*, which forces the broker to be more risk-averse than the brokerage model suggests; and the more influential *payment bug*, which incorrectly calculates the payment a broker makes to the provider, thereby inflating the broker's profits.

After demonstrating the impact of the reservations and payment bugs, we then presented a set of "corrected" results for the brokerage model, intended to replace those presented by R&C. While the corrected results demonstrate that the brokerage model is less profitable in absolute terms than previously thought, the conclusions drawn by R&C still largely hold.

However, since R&C's model was published, the landscape for cloud computing provision has radically altered. In particular, increased competition between providers has driven down prices and encouraged innovation, such as the introduction of a secondary marketplace—Amazon's Reserved Instance Marketplace (ARIM)—for users to buy and sell reserved instances. To reflect these changes, we updated the assumptions of R&C's model, by altering the maximum allowable cost factor.

In R&C's original work, the model was bounded to ensure that options from the broker are always cheaper for users than an on-demand instance purchased directly from the provider. However, the introduction of ARIM means that users can now purchase instances on the secondary market for prices cheaper than the on-demand price. Therefore, we assume that the price of an option must always be cheaper than the expected price of a reserved instance on ARIM. Under this assumption, we show that the broker's profits will reduce as ARIM grows in popularity and at the expected equilibrium the brokerage model is no longer profitable.

We therefore conclude that the introduction of a new secondary market venue for users to buy and sell RIs has closed the window of opportunity to commercially exploit the brokerage model of R&C. This is particularly compelling, since R&C's brokerage model assumes that the broker has a monopoly position. Indeed, the WZH reservation model, which R&C's brokerage model extends, is described by the authors as a mechanism that *"extracts revenue similar to that of a monopoly provider practicing temporal pricing discrimination with a user population whose preference distribution is known in advance"* [11]). Thus, the brokerage model relies on the broker being somehow able to position itself as a monopoly provider of options; perhaps through an exclusive licensing agreement with the provider. However, every user now has the ability to trade RIs on ARIM. Thus, the assumption that the broker has a monopoly position is no longer tenable. For this reason, the brokerage model must be able to compete with prices expected on the secondary market. As we have shown, this is not possible.

## Appendix

## Rogers & Cliff: right of reply

Here, in the interest of fairness and clarity, we present Owen Rogers' response to the camera-ready version of this paper (20/03/2014).

### Replication

*"We're disappointed that errors crept into the code, but are pleased that our main finding that the consumer and broker can both benefit using the scheme still holds. JC and PC's work demonstrates the importance of confirming results using independently developed code, a point we raised in our original paper".*

### Calibration

*"The ARIM marketplace, rather than weakening the commercial viability of the mechanism, could provide more opportunities for the broker to hedge capacity. We agree that based on the current market, the brokerage model is less commercially viable than when previously investigated, but only when the broker purchases standard RIs directly from AWS.*

*"The broker could potentially buy reserved instances from the ARIM too, giving it further opportunity to reduce its costs. It could also choose to offer options of smaller durations not supported by the current ARIM, such as options for a week or fortnight. Furthermore, the ARIM gives the broker the opportunity to purchase RIs for a wide range of time periods, rather than just the 12 and 36-months offered as standard and investigated in the original paper - this could enable a wide range of thresholds to be set for each length of reserved instance to maximise utilization of the RIs, forcing costs down further. The mechanism in these scenarios essentially remains the same - the configuration of the options sold, and RIs bought is different.*

*"The original mechanism was simulated in a monopoly market. It would be interesting to assess the mechanism in a competitive market, where each broker differentiates, essentially, through its assessment of the risk in purchasing resources. Competing and differentiated financial broker-dealers currently exist in other commodity markets. Could the accuracy of thresholds (which may vary over time, based on market prices in the ARIM) provide opportunities for multiple brokers to co-exist, taking profits dependent on different market trends? By using the ARIM marketplace as a source of cheaper resources, this could potentially provide some interesting results on market dynamics."*

### Abbreviations

ARIM: Amazon EC2 reserved instance marketplace; AWS: amazon web services; CReST: cloud research simulation toolkit; DC: data centre; MRU: marginal resource utilisation; MVC: model-view-controller; OS: operating system; R&C: Rogers and Cliff (2012) [1]; RI: Reserved instance; VM: Virtual machine; WZH: Wu, Zhang, and Huberman (2008) [11].

### Competing interests

The authors declare that they have no competing interests.

### Authors' contributions

JC conceived and coordinated the study and wrote the full manuscript. JC architected and managed the development of CReST and was the primary code developer. All experiments and analysis presented here were performed by JC. PC was primarily responsible for developing the CReST brokerage code. PC also performed preliminary experiments and statistical analysis under the supervision of JC, which led to the discovery of the "bugs" in R&C's model. Both authors read and approved the final manuscript.

### Authors' information

JC is a Research Associate in Cloud Computing at the University of Bristol. He has a BSc in AI & Mathematics (Leeds, 2000) and a PhD in Computer Science (Leeds, 2004). Since 2008, John has held academic research positions, first at the University of Central Lancashire and then at the University of Bristol. Previously, he spent four years working in industrial technology research for Hewlett-Packard Labs, the London Stock Exchange and several technology start-ups. John is Director of Electric Lamb Ltd, a software development company and IT consultancy specialising in financial systems, cloud computing and data analytics.
PC is an Analyst Developer at Goldman Sachs. He graduated from the University of Bristol with a MEng in Computer Science in 2013. His final year

research project, entitled "Pricing the cloud: an investigation into financial brokerage for cloud computing", led to many of the discoveries presented in this work.

**Acknowledgements**
The authors would like to thank Owen Rogers for supplying his demand data and original source code that enabled us to perform a strict replication of R&C's brokerage model. We also acknowledge the contributions of all the LSCITS summer interns that contributed to the development of the CReST simulation platform in 2011 and 2012: Luke Drury, Tomas Gražys, Sarah Haswell, James Laverack, Callum Muir, and Alex Sheppard. CReST was developed at the University of Bristol, UK, using financial support from the UK's Engineering and Physical Sciences Research Council (EPSRC) grant EP/H042644/1 entitled "Cloud Computing for Large-Scale Complex IT Systems". Primary financial support for JC comes from EPSRC grant number EP/H042644/1. LSCITS and the LSCITS internship program is funded by EPSRC grant number EP/F001096/1.

**Author details**
[1]Department of Computer Science, University of Bristol, Woodland Road, Bristol, BS8 1UB, UK. [2]GSS Technology, Goldman Sachs International, Peterborough Court, 133 Fleet Street, London, EC4A 2BB, UK.

**Received: 28 November 2013 Accepted: 21 March 2014**
**Published: 8 April 2014**

**References**

1. Rogers O, Cliff D (2012) A financial brokerage model for cloud computing. J Cloud Comput: Adv, Syst Appl 1(2): 1–12
2. Most Viewed Articles. http://www.journalofcloudcomputing.com/mostviewed/alltime. Accessed 02/09/13
3. Antonello M, Baibussino B, Benetti P, Boffelli F, Calligaric E, Canci N, Centro S, Cesana A, Cieslik K, Cline DB, Cocco AG, Dabrowska A, Dequal D, Dermenev A, Dolfini R, Farnese C, Fava A, Ferrari A, Fiorillo G, Gibin D, Gninenko S, Guglielmi A, Haranczyk M, Holeczek J, Ivashkin A, Kisiel J, Kochanek I, Lagoda J, Mania S, Menegolli A, et al. (2012) Precision measurement of the neutrino velocity with the ICARUS, detector in the CNGS beam. J High Energy Phys 49: 1–21
4. CReST - The Cloud Research Simulation Toolkit. https://sourceforge.net/projects/cloudresearch/. Accessed 17/09/13
5. Cartlidge J, Cliff D (2013) Comparison of cloud middleware protocols and subscription network topologies using CReST, the cloud research simulation toolkit. In: Desprez F, Ferguson D, Hadar E, Leymann F, Jarke M, Helfert M (eds) 3rd Int. Conf. Cloud Computing & Services Science (CLOSER-2013). SciTePress, Aachen, pp 58–68
6. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M (2009) Above the clouds: a berkeley view of cloud computing. Tech. Rep. EECS-2009-28, University of California, Berkeley
7. Rogers O, Cliff D (2013) Contributory provision point contracts—a risk-free mechanism for hedging cloud energy costs. J Cloud Comput: Adv, Syst Appl 2(10): 1–18
8. Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/. Accessed 22/09/13
9. Amazon EC2 Reserved Instances. http://aws.amazon.com/ec2/reserved-instances/. Accessed 22/09/13
10. Amazon EC2 Spot Instances. http://aws.amazon.com/ec2/spot-instances/. Accessed 22/09/13
11. Wu F, Zhang L, Huberman BA (2008) Truth-telling reservations. Algorithmica 52(1): 65–79
12. Clearwater SH, Huberman BA (2005) Swing options: a mechanism for pricing IT peak demand. In: 11th Int. Conf. on Computing in Economics & Finance CEF-2005, Washington, DC. http://www.hpl.hp.com/research/idl/papers/swings
13. Fujitsu Laboratories Fujitsu laboratories develops world's first datacenter simulator for promptly predicting the total energy consumption and evaluating energy-saving control of datacenters, 13/10/2011. http://www.fujitsu.com/global/news/pr/archives/month/2011/20111013-01.html. Accessed 18/03/14
14. CoolSim4. http://www.coolsimsoftware.com/. Accessed 24/09/13
15. Calheiros RN, Ranjan R, Beloglazov A, Rose CAFD, Buyya R (2011) Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software Pract Ex 4(1): 23–50
16. Casanova H, Legrand A, Quinson M (2008) Simgrid: a generic framework for large-scale distributed experiments. In: Proceedings of the tenth international conference on computer modeling and simulation. *UKSIM '08*. IEEE Computer Society, Washington, pp 126–131
17. Clamp P, Cartlidge J (2013) Pricing the cloud: an adaptive brokerage for cloud computing. In: Bauer M, Lorenz P (eds) 5th Int. Conf. on Advances in System Simulation (SIMUL-2013). IARIA XPS Press, Venice, pp 113–121
18. Clamp PJ (2013) Pricing the cloud: an investigation into financial brokerage for cloud computing. Master's thesis, Dep. Comp. Sci., Univ. Bristol, UK
19. Amazon EC2 Reserved Instance Marketplace - Beta. http://aws.amazon.com/ec2/reserved-instances/marketplace/. Accessed 26/09/13
20. Ricknäs M Amazon Web Services Lets Users Sell Reserved Instances, 12/09/2012. http://www.computerworld.com/s/article/9231204/Amazon_Web_Services_lets_users_sell_reserved_instances. Accessed 22/09/13
21. Bouchard JP, Farmer JD, Lillo F (2009) How markets slowly digest changes in supply and demand. In: Hens T, Schenk-Hoppé KR (eds.) Handbook of financial markets: dynamics and evolution. North-Holland, Elsevier, Inc., Amsterdam, pp 57–160
22. Bradic A Caveat Emptor: Amazon EC2 Reserved Instance Marketplace, 12/01/2014. http://blog.supplyframe.com/2014/01/12/caveat-emptor-amazon-ec2-reserved-instance-marketplace/. Accessed 17/03/14
23. Cartlidge J (2004) Trading experiments using financial agents in a simulated cloud computing commodity market. In: Duval B, van den Herik J, Loiseau S, Filipe J (eds) 6th Int. Conf. on Agents & Artificial Intelligence, Vol. 2 - Agents (ICAART-2014). SciTePress, Angers, pp 311–317
24. Cliff D, Bruten J (1997) Minimal-intelligence agents for bargaining behaviors in market-based environments. Tech. Rep. HPL-97-91, Hewlett-Packard Labs (August 1997). http://www.hpl.hp.com/techreports/97/HPL-97-91.pdf
25. De Luca M, Szostek C, Cartlidge J, Cliff D (2011) Studies of interactions between human traders and algorithmic trading systems. Foresight Driver Review - DR13. In: The future of computer trading in financial markets. Government Office for Science, Crown Copyright, London, pp 1–60. http://webarchive.nationalarchives.gov.uk/20121212135622/http://www.bis.gov.uk/assets/foresight/docs/computer-trading/11-1232-dr13-studies-of-interactions-between-human-traders-and-algorithmic-trading-systems.pdf
26. Stotter S, Cartlidge J, Cliff D (2013) Exploring assignment-adaptive (ASAD) trading agents in financial market experiments. In: Filipe J, Fred ALN (eds) 5th Int. Conf. on Agents & Artificial Intelligence, Vol. 1 - Agents (ICAART-2013). SciTePress, Barcelona, pp 77–88
27. Stotter S, Cartlidge J, Cliff D (2014) Behavioural investigations of financial trading agents using Exchange Portal (ExPo). LNCS Trans Comput Collect Intell (TCCI) (in press)
28. Dana JD Jr (1999) Equilibrium price dispersion under demand uncertainty: the roles of costly capacity and market structure. RAND J Econ 30(4): 632–660
29. Cartlidge J, Bullock S (2003) Caring versus sharing: how to maintain engagement and diversity in coevolving populations. In: Banzhaf W, Christaller T, Dittrich P, Kim JT, Ziegler J (eds) 7th Eur. Conf. Artif. Life (ECAL-03). Springer Verlag, Dortmund, pp 299–308
30. Cartlidge J, Bullock S (2004) Unpicking tartan CIAO plots: understanding irregular coevolutionary cycling. Adaptive Behav 12(2): 69–92
31. Cartlidge J, Phelps S (2010) Estimating consumer demand from high-frequency data. In: Kumar K (ed) Ann. Int. Academic Conf. Business Intelligence & Data Warehousing (BIDW-2010). Global Science and Technology Forum (GSTF), Singapore, pp 132–138
32. Cartlidge J, Phelps S (2011) Estimating demand for dynamic pricing in electronic markets. GSTF J Comput (JoC) 1(2): 128–133

doi:10.1186/2192-113X-3-2
**Cite this article as:** Cartlidge and Clamp: **Correcting a financial brokerage model for cloud computing: closing the window of opportunity for commercialisation.** *Journal of Cloud Computing: Advances, Systems and Applications* 2014 **3**:2.