STRATEGIC DECISION SUPPORT FOR SMART-LEASING INFRASTRUCTURE-AS-A-SERVICE

Completed Research Paper

Christian Bodenstein
University of Freiburg
Platz der Alten Synagoge
79085 Freiburg, Germany
christian.bodenstein@is.uni-freiburg.de

Markus Hedwig
University of Freiburg
Platz der Alten Synagoge
79085 Freiburg, Germany
markus.hedwig@is.uni-freiburg.de

Dirk Neumann
University of Freiburg
Platz der Alten Synagoge
79085 Freiburg, Germany
dirk.neumann@is.uni-freiburg.de

Abstract

In this work we formulate strategic decision models describing when and how many reserved instances should be bought when outsourcing a workload to an Infrastructure-as-a-Service (IaaS) provider. Current IaaS providers offer various pricing options for leasing computing resources. When decision makers are faced with the choice and most importantly with uneven workloads, the decision at which time and with which type of computing resource to work is no longer trivial. We present case studies taken from the online services industry and present solution models to solve the various use case problems and compare them. Following a thorough numerical analysis using both real, as well as augmented workload traces in simulations, we found that it is cost efficient to (1) have a balanced portfolio of resource options and (2) avoid commitments in the form of upfront payments when faced with uncertainty. Compared to a simple IaaS benchmark, this allows cutting costs by 20%.

Keywords: Cloud computing, E-business, Outsourcing, Strategic information systems, Web services

Introduction

For every online web service, a stable computing infrastructure is essential. While in a start-up the requirements might still be manageable, once the service runs successfully you will rapidly need to expand your infrastructure into a large scale datacenter. Where a few servers can be stored in basement-like rooms, once the infrastructure reaches a certain magnitude, entire datacenter architectures need to be set up – and with it, the expertise required to run them.

The costs of owning computing infrastructures, on a large scale to harbor such online services, have continually increased and if the web service is successful, this bill will only increase. Start-up's aside, even established businesses might face a large upfront investment to accommodate for a sudden increase in computing infrastructure. Due to spatial imitations, investment capital shortage, or any other limiting factors, CIO's are often unable to harbor an increased computing demand by expanding the in-house computing architectures.

With technological innovations in the service science industry, cloud computing in particular, the management of IT infrastructures can be outsourced in the form of a leasing contract much like the leasing contracts available in the car industry. Instead of purchasing a car, a leasing contract allows a car to be rented at a monthly fee, paying only for the wear and tear of everyday use. The main advantage is that if the car is no longer needed, it can be given back at no extra cost. The IT counterparts to car-rentals are Infrastructure-as-a-Service (IaaS) providers. For example, Amazon EC2 allows users to select and configure the IaaS hardware instances at a per-hour rate. These per-hour rates vary depending on how often IaaS users require the underlying computing infrastructure. Users can either purchase computing power at a per-hour rate without long-term commitments, or, following a fixed annual fee, purchase the same computing power at a reduced per-hour rate. For applications which require 100% uptime, choosing reserved instances over on-demand instances potentially cuts the overall costs in half (Amazon 2011).

The question is in what proportion do we purchase these instance options? Simply running with only on-demand instances can become inefficient. On the other hand, operating the entire workload with only reserved instances could also be inefficient if the workload was only required for a short period of time. Regardless of whether the decision-maker is an entrepreneur new to the industry, or a well established IT-Manager, when configuring his infrastructure to meet future capacity requirements, a manager is faced with the question, *what should the cost minimal computing portfolio of IaaS instances look like*? The question, and therefore the general research question pursued in this work, is *in what proportion cloud infrastructures should be leased in order to effectively provide an online web service at minimal cost?*

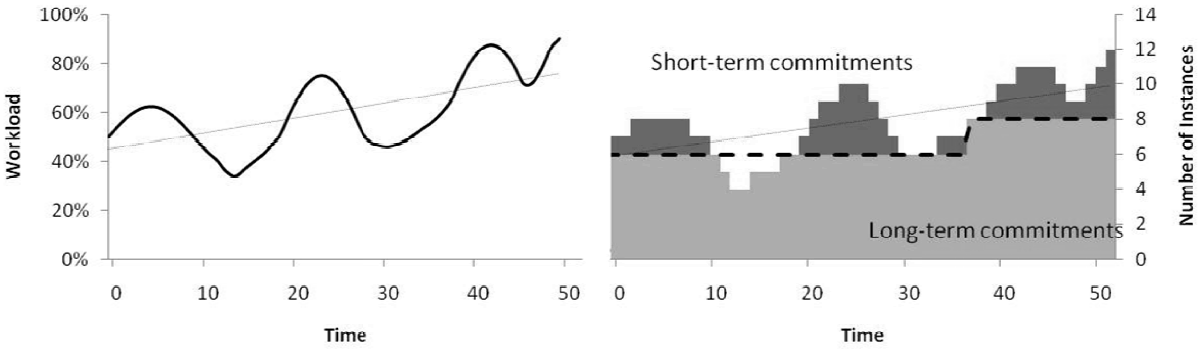

Figure 1. Optimal Resource Assignment Intuition

Figure 1 portrays the idea we pursue in this work. Assume we are given a workload which varies as time progresses. Further assume that this workload can be plotted as a function between 0 and 1, where 0 represents the state where no workload is generated and 1 is the highest recorded workload. Figure 1 (left) shows such a typical workload scenario with volatility in the short run and a gradual long-run workload increase (shown by the linear trend added to the workload trace). The diagram on the right shows how we intend to handle the workload by making use of both reserved instance resources and on-demand resources (for simplicity of the example we handle the number of instances discretely). By cleverly combining the portfolio of long-term and short-term resource commitments we are able to create a

resource portfolio capable of handling any workload as cheap as possible. When operated over a longer period of time, the reserved instances become cheaper per-hour, since they generally include some installment fee. On-demand resources on the other hand have fixed (generally hourly) charges and are pre-destined to be used for short peaks. This optimum is portrayed by the black dotted line.

While the pricing decision for IaaS providers has been covered in research (Harmon et al. 2008), little work has been done to help the end consumers' challenge of selecting the right portfolio of such IaaS resources for their specific needs. This work explicitly addresses this challenge and provides a novel resource management model to support decision makers in selecting their IaaS portfolios. We introduce a theoretic model utilizing state-of-the-art concepts from the field of service science. In previous work we evaluated the use of IaaS as a supplement to in-house server architecture (Hedwig, Bodenstein, et al. 2010), however in this work we intend to present a decision model based solely on using IaaS. More concretely, this work presents a Strategic IaaS Investment Support System, a framework to assess and interpret the cost involved in the operation of a pure IaaS system. This may be used as both a decision support system for investment and a tool for examining the cost impact of policy changes (i.e. "*what happens if we increase our share of instance type X?*"). Along with an intuitive mathematical formalization, it gives an overview of IaaS providers including their perks and drawbacks. In its role for finding an optimal investment strategy for IT equipment, this work can be generalized as a decision support system able to inform decision makers on how to invest in cloud infrastructures. Therefore, this work is more than just a standard buy-or-lease model since we also cover strategies that function dynamically and in real-time, presenting practically feasible solutions.

This work is set up as follows. In the next section we review related literature on IT outsourcing, IaaS business models and their applications, and define the problem set covered in this work. We review the current scope of IaaS services available, as well as cover the characteristic workloads of online services in use today. Subsequently, we formulate a continuous model to show how the per-hour price develops as we change the share of various IaaS products in the third section. To be viable as a decision model, however, we require a discrete decision model, which is presented and evaluated in the fourth section. Both the continuous and discrete models cover an ex-post view. As a strategic tool, we require a portfolio of strategy suggestions for decision makers. This is covered in section five along with an evaluation of the strategies. In section six we conclude this work and present a summary as well as an outlook toward further research avenues.

Related Work

The general idea of IT outsourcing is to decouple tasks to a 3rd party provider who is able to operate reliably and more efficiently. IT Outsourcing, especially in the context of IT vendors, has become widespread with firms outsourcing anything from repetitive tasks to entire business processes (Fichman and Kemerer 2005; Greco 2001). Defined as "the transfer of a part or all of IT services to an external service provider" (Bahli and Rivard 2005), IT-outsourcing originated with the financial and operational services sector in the 1960s. Since its inception however, IT outsourcing has evolved from initial software development to the hosting of IT infrastructure and applications (Lacity et al. 2009; Lacity and Hirschheim 1993). According to (Progent Research 2002), over half of all IT services of North American vendors were outsourced to a third party in 2000.

The question which remains is whether outsourcing is always profitable? In related literature, outsourcing is often viewed as a means to increase efficiency and to reduce the total operation costs by reaping economies of scale benefits as well as allowing vendors to access specialized resources and know-how (Ketler et al. 1993). Some researchers, however, view outsourcings more harshly and coin it as a loser's game. (Strassmann 2004) analyzes the payroll data of a diverse and random sample of 324 companies listed in Standard & Poor's financial reports and his observations contradict the often claimed theory that suggests that outsourcing improves profitability. (Bettis et al. 1992), points out that outsourcing can lead to the depreciation of a firm's capabilities. Further, in the case of customer-facing technologies it could result in disruptions in the firm-customer relationship. For example, (Weigelt 2009) points out that when outsourcing services such as internet applications, a firms customers directly interact with the outsourcing partner. According to (Johnson 2003) this might reduce the firm's ability to reap benefits. Nevertheless, outsourcing has continued to grow in popularity for the IT industry, and the debates have shifted from whether or not it is good to outsource IT, to how much IT to outsource. Research has

included whether outsourcing should be total or selective, involve services or assets, be long or short term, and involve single or multiple vendors (Lacity et al. 2009). (Khajeh-Hosseini et al. 2010; Motahari-Nezhad et al. 2009) show in their case study that outsourcing computing tasks instead of operating with in-house infrastructure is a feasible operation strategy.

With advances in virtualization technology easing the transfer of data, the scope and intensity of outsourcing changed. Several models have been developed to assess the feasibility and implications of IT outsourcing (Gilley and Rasheed 2000). As a hardware abstraction layer virtualization provides and enables fast deployment and migration of encapsulated software systems. Furthermore, it enables the fine-grained management of server resources (Padala et al. 2007). Service-oriented architectures constitute the first feasible software design concept to efficiently distribute applications over several servers with enough abstraction from the hardware to dynamically add to and remove resources from the system (Channabasavaiah et al. 2003). This process of adding and removing hardware can be automated. The idea of automatically scaling a system by including IaaS has been implemented using an automated load balancer in (Ragusa et al. 2008). Their load balancer continuously monitors the server and activates remote resources upon detecting a potential demand overload, showing the technical practicability of using IaaS. In this work, we are less dependent on the accuracy of future resource demand predictions; if we need more, we simply lease more using either the automated vendor specific load balancers, or balancers similar to those presented in (Ragusa et al. 2008).

Infrastructure as a Service (IaaS) is a business model for the cloud computing industry to put up remote resources for lease. Specifically, IaaS supplies hardware and equipment to deliver software applications at a resource-based pricing (Foster et al. 2008). Unique to IaaS is the ability to scale the hardware dynamically, based on the application's resource needs using the vendor-specific controller. Focusing on the investment of IT infrastructures, in (Jagannathan et al. 2003) the authors derived the optimal infrastructure size based on the analysis of their customer contracts and past user behavior evaluations. As a result, they were able to forecast their system utilization and derive cost optimal investment decisions for their system architectures. However, the authors still optimized their investment decisions to satisfy peak demand. If the forecast is inaccurate, then the investment is not sufficient.

The idea of leasing computing resources has already been well established since the 1960s. Oddly, (Gray 2003) found computing resources have not readily established themselves as standard products, resulting in the economy of scale effect being virtually non-existent. From this point-of-view, outsourcing IT systems and using IaaS results in generally economically infeasible options, and would only be implemented when computing requirements are beyond the financial capabilities of single enterprises, as proposed by (Lee et al. 2003). In (Risch and Altmann 2008) this assumption is empirically proven by examining the charging models of Grid and Cloud providers as well as determining the operation costs of owned infrastructure. Their comparison shows that established enterprises can usually provide their computing demand themselves at lower costs; however the authors did not look at the total costs involved in operating the hardware, but rather used a simple and static binary decision of buying or renting the computing hardware. This result does not hold when looking at a broader cost structure and with recent developments in cloud technology, we have shown in related work (Hedwig, Malkowski, et al. 2010) that it is cheaper to operate in the cloud as an Infrastructure as a Service (IaaS).

With the abundant offerings from the IaaS industry, given the uncertainty about future computing requirements and fluctuating levels of computing workloads, the choice of how to configure the optimal portfolio of IaaS hardware is a tough decision. We found it a commonplace for IaaS providers to offer some form of price discrimination in their pricing schemes, two of which are of interest for this work. The most prominent two classes of price discrimination are the standard per-hour pricing and the reduced per-hour pricing following an upfront payment. Each of these options is predestined for certain types of IaaS services. While the standard per-hour rates are intended for short-term computing requirements, the reduced per-hour rates are only profitable for long-term computing requirements because of the high up-front payment. Typical examples of IaaS providers are Amazon or Rackspace, who offer computing services at an hourly charge. In this work we use the elastic computing services provided by Amazon EC2 as a discussion and sample base, as it is a good market representative.

The IaaS Instance Purchasing Model

Since 2002, Amazon EC2 allows users to freely select and configure the size of the IaaS instances by supplying different purchasing options offering the same service. Relevant to this work are *On-Demand Instances* and *Reserved Instances*. *On-Demand Instances* are highly demand oriented (computing power is adapted as required by the application) without long-term commitments, where users pay for instances by the hour. Amazon recommends these instances be used for short-term workloads with volatile CPU and Memory usage. *Reserved Instances* are a purchase option with long-term commitment in mind. Following a minor fee, instances are reserved for the duration of 1-3 years. These instances then have a reduced per-hour price and are, most importantly, guaranteed to be available. For applications which require 100% uptime, choosing reserved instances over on-demand instances potentially cuts the overall costs in half. (Amazon 2011)

Further examples of IaaS Suppliers, ranked in the Top 10 IaaS Providers of 2010 (SearchCloudComputing.com 2010), are GoGrid's "HybridHosting" (see (GoGrid 2011)), Rackspace (see (Rackspace 2011)), Microsoft Windows Azure (see (Azure 2011)), Joyent (See (Joyent 2011)) or Verizon (See (Verizon 2011)). Common to most of the above providers is their pricing set-up defined by two distinct groups: Instances without commitment at a high hourly rate and instances with commitment at a lower hourly rate following some upfront payment. The commitment periods range between six months and three years, although a yearly commitment is most commonly sold. In this work we will use the jargon and pricing set-up from the Amazon Online Services as a base example to work with, since Amazon is the leading *IaaS* provider and has the most transparent pricing scheme. Regardless, this model is not unique to Amazon's Services. Other providers could just as well be chosen.

Looking at the demand side, end users' resource utilization is generally highly fluctuant. The most prominent fluctuation factors are periodic effects in user behavior. Usually this can depend on the time of day or day of the week. In the context of our web service entrepreneur, an example could be a hotel reservations site, where more page accesses are registered during the holiday season and fewer in off-peak seasons. Another example could be an auction site, where more information is transferred during an auction. In the case of more established sites, fluctuations can be more repetitive, as shown in Figure 2, the page accesses on Wikipedia. (Mituzas 2011)

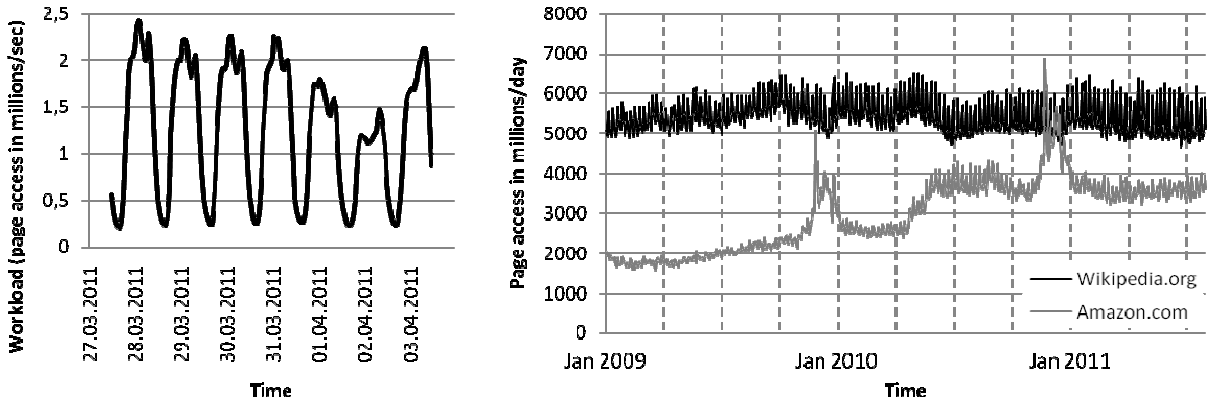

Figure 2. Wikipedia.org and Amazon.com Page Access Workloads

A sample extract from the workload trace of the last week in March 2011 is shown in Figure 2 (left) showing the workload in page accesses *per second*. This particular week was arbitrarily chosen and found representative of the typical workload distribution. While the workload peaks off at around noon, the lowest utilization levels are realized at around 04:00 in the morning. Over the course of the week, the workload on Saturdays (02.04.2011) is generally significantly lower than on Mondays (28.03.2011). Looking at longer time periods, Figure 2 (right) shows the *daily* page views 'crawled' by Alexa.com over the past two and a half years (which coincidently is also hosted on an IaaS platform). However, the data is to be handled cautiously since it is an independent and one-sided view mined from Alexa users. Viewed on its own, it does not allow for rigorous analysis. Compared to other sources it can be used as an indicator of a *de-facto* trend for the industry (assuming that the Alexa users represent the industry). For

comparison, we added a very seasonal page access trace of amazon.com (the lower grey plot). Most noticeable is the page view surge at the end of the year. This surge is most likely related to Christmas and end of the year budget purchases, where most of Amazon.com sales are recorded. While Wikipedia.org page accesses can be seen to be relatively constant with daily fluctuations, the trace of Amazon is a good example of the presence of seasonal effects in a workload sample.

From the above examples we can conclude, if service provider's provide for peak resource requirements using in-house architecture, for a scenario with fluctuant workloads, it would result in some of the resources running idle for most of the time – and with it high and unnecessary costs. This is not the case if the workloads are processed in the cloud infrastructures proposed above, since IaaS users are only charged when resources are actually used. As an IaaS example, Amazon set up their computing instances as packaged environments in the form of machine images. Through a management console users are able to scale their instances seamlessly during spikes to maintain performance using a management console CloudWatch™. This scaling can also be handled automatically using the so-called auto-scaling option (Amazon 2011). In this work we will make use of this technology to promote future online services, moving from in-house applications to perfectly scalable cloud-based services.

With the technical foundation for scalable online services set, we now focus on the economic question of how much it costs. Common ground for cloud infrastructures is the option to purchase computing resources with commitment of some form or another at a discount price following some upfront payment or to purchase resources as you need them at a standard price. Analogue to Amazon EC2, we refer to these purchase options as *reserved* and *on-demand* respectively. Therefore, to minimize the total costs of operation, we must monitor and adapt the amount of reserved instances. The idea is to provide for some basis of operation using reserved instances and to service the less frequent peaks with on-demand purchases while fulfilling the target workload at all times. Since prices are given by the IaaS providers the costs for providing a service based on the IaaS infrastructure can be calculated. Of interest is in what combination reserved and on-demand instances should be purchased for cost minimal operation.

Continuous Model

Since online services are known to have volatile demand patterns, the optimal choice of purchase options would be to purchase a set level of reserved instances to serve the base workload at a reduced per-hour rate and to serve the remaining peaks with on-demand. If w_t describes the level of workload over time and with hindsight $max(w_t)$ is known, then $b = i(w_t) \in [0\%; 100\%]$ determines the share of server instances required to satisfy the demand w_t. To uphold the model continuity, we assume that the resource demand is completely divisible.

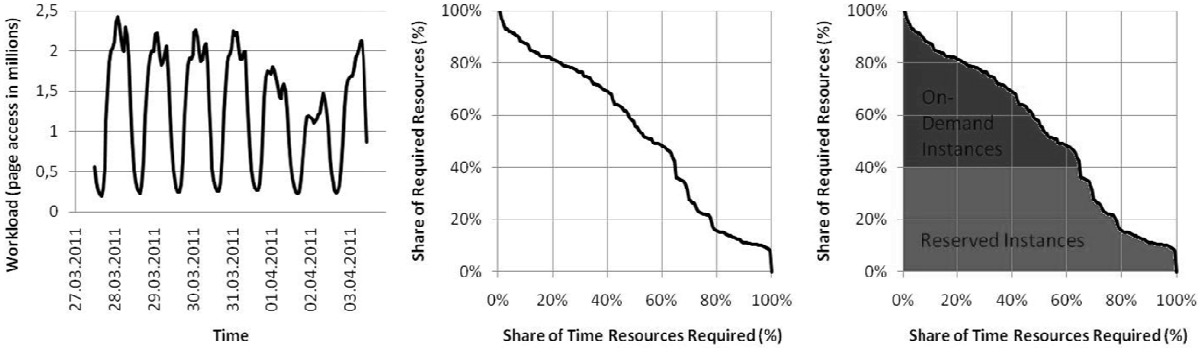

Figure 3. Acquiring the Share of Instances from the Workload Trace

Figure 3 (left to right) depicts the process of transforming the raw workload to a breakdown of how many instances are required for what length of time. The left diagram shows the raw page access workload as it occurred in the week between 20.03.2011 and 03.04.2011. The center diagram shows the same workload in a distribution graph showing what share of resources was required for how long. We now turn to defining the workload demand percentage divided into a percentage served by reserved instances and on-demand instances shown in Figure 3 (right). Let x_r be the percentage of reserved instances. Hence $(1 - x_r)$ is the percentage of *IaaS* resources purchased on-demand. For example, if we assume $x_r = 40\%$, then the

resulting relationship can be illustrated as shown in Figure 3 (right). Up to a workload of *40%* of the peak demand, the reserved instances are able to cope. These 40% are sufficient to service up to 62% of the investigated time interval. For the remainder, the reserved infrastructure must be supplemented by on-demand infrastructures to fulfill the service level objectives (in this case 100%). Note that the marginal utilization function with a set x_r divides the figure into four quadrants. Each quadrant is represented by a cost function per resource unit. The on-demand quadrant is the simplest, as the cost function for on-demand resources, $c_{o,}$ is equivalent to the hourly charge given exogenously by the *IaaS* provider. The reserved instances are a little trickier since idle reserved instances have a sunk-cost effect, $c_{i,}$ which must be factored into the total cost of operation function in addition to the utilized states and the advance payment of the reserved instance costs, c_r. The residual quadrant describes the idle reserved and on-demand resources. As per definition these incur no costs, thus $c_{res}(t) = 0$.

Deriving the Optimal Share of Reserved Instances

Consider a two-year scenario with a resource utilization function, where the peak demand remains constant over the course of these two years. Since the prices for our IaaS instances are given exogenously:

Proposition 1: The optimal number of reserved instances solely depends on the shape and magnitude of the marginal workload distribution and the workload time horizon. (Proof moved to the Appendix. See Proof Sketch to Proposition 1)

Graphically, this relationship can be displayed in an average cost graph plotted against the share of reserved instances for various lengths of work time spans. Figure 4 shows the average price development using the Wikipedia workload trace and the current market prices for extra large standardized instances operating MS Windows from Amazon EC2. An on-demand instance costs €0.96/h. The Reserved instances were purchased for a year respectively and cost €0.48/h following an upfront payment of €1820.00. If the workload is constant, the break-even point for the decision whether to use only reserved instances or on demand instances is solely dependent on the workload duration (i.e. how long do you require the instances). For the above pricing sample the break-even point lies at 3791 2/3 hours. In other words, given a constant workload, and a duration of only 3791 hours (157,95 days), the service should be operated using only on-demand instances. For a duration of more than 3792 hours (or 158 days), it is cheaper to operate the web service using only reserved instances.

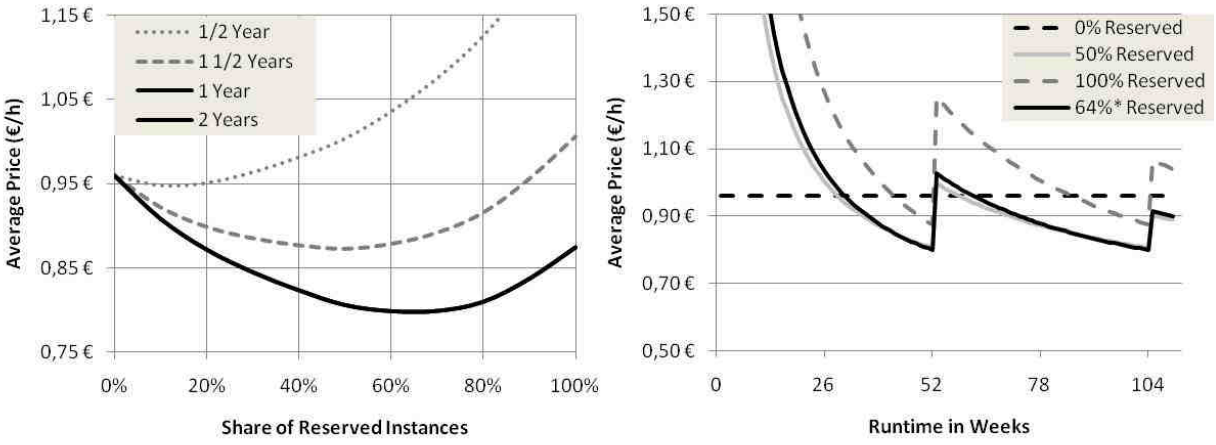

Figure 4. Average Costs of Operation per Share of Reserved Instances

Naturally, the choice of instances is not limited to an either-or configuration. Figure 4 (left) shows how the average price paid develops, as the share of reserved instances increases for various duration horizons. The average price for the infrastructure is lowest when the workload duration time horizon reaches a full year. The chart shows four chosen timeframes, namely ½, 1, 1½ and 2 years. The grey plots show the ½ and 1½ year duration plots respectively. The plots for one and two years overlap and in their price per-hour are consistently cheaper than the plots for the half years. This is largely due to the high initial payment needed for purchasing reserved instances. Since it takes some time for the reserved instance investment to pay off, the closer to a full year the infrastructure is used the lower the per-hour costs to

operate. For the Wikipedia sample workload trace shown in Figure 3 the optimal share of reserved instances relative to the peak workload is 64%.

Figure 4 (right) shows the average price per-hour as the runtime duration increases for various configurations of reserved instance shares. It was calculated using the various reserved instance share configurations of 0%, 50%, 100% and the optimal 64% (found in analysis for Figure 4 (left)) and averaged for the various runtime in weeks. The dotted horizontal price line is the average cost of using only on-demand instances. It does not change based on the runtime in weeks. The also dotted '100% Reserved' plot forms the exact opposite strategy to only using on-demand instances. As a strategy to serve the Wikipedia workload trace, on-demand instances is the preferred choice for timeframes of up to 42 weeks. For longer time periods, the reserved instances were the most cost-efficient choice. In other words, if the duration of future workloads were subject to a uniform distribution, using only on-demand instances would be the right choice 81% of the time, while reserved instances would only be preferred in 19% of the cases. Note that all strategies using a share of reserved instances experience a sudden increase in average prices at week 53. This is due to the contract renewals for reserved instances. In week 53, the costs for reserved instances are the workload processed multiplied by the hourly costs plus twice the installment fee, since the reserved instances are reserved for a second year. As the second year progresses the costs again decrease.

While both the time- and resource continuous model can generally be used to explain the theoretical behavior of the optimal reserved instance investment and derive optimal ratios of instances, a solution "use 6.7 reserved instances in t1, 5.6 in t2,…" is of limited use to decision makers. Managers need discrete answers for the reserved instances and the number of hours to be purchased. While the continuous analysis was able to give us some theoretical and formal insight into the solution to our decision problem, to act as a decision model, we must formulate the model discretely. This is done in the next section.

Discrete Model

Both time and resources are formulated discretely. Since the minimal interval for pricing is set exogenously by the IaaS providers, namely per-hour, we set the time interval to discrete hourly intervals (though the model is not limited to an hourly interval). Also, the amounts of reserved instances purchased are defined by discrete integers. For this reason we must convert our continuous workload demand function into a discrete step function. Figure 5 visually shows how this can be done (for a more formal introduction see (Hedwig, Bodenstein, et al. 2010)).

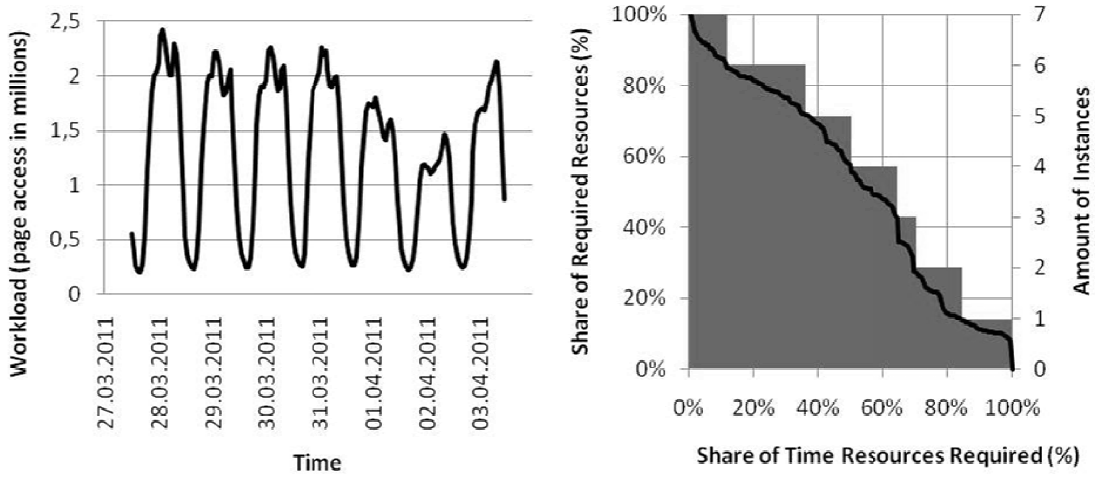

Figure 5. Discrete Instance Share Breakdown

Assume for the above example that the share of resources can be served using seven instances. The right diagram shows this distribution in terms of discrete server instances required to process the workload. Acting as a form of a marginal resource utilization curve, this transformation is necessary to derive the optimal amount of reserved instances. Similar to the example in Figure 5 we can also break up the share of resources into reserved and on-demand instances. The only difference is that for the discrete models

the instances are no longer infinitely divisible. Using the 40% rule from the continuous example above, we can read off Figure 5 (right) that the discrete model would require three reserved instances, while the rest would be served with on-demand instances.

With both time and resources no longer continuous, we can formulate the *Instance Purchasing Problem* (short *IPP*) as a discrete mathematical problem. Suppose you are an entrepreneur wishing to build up a online web service based purely on using third party IaaS Services. Assuming your income is exogenous, as a web service provider your goal is to minimize your total costs on all future operations of your online service, while maintaining a good service level standard. Unfortunately, you are unable to predict exactly how many resources you will require at any given time in the future. The only information you do have is historic information from other online services. Referring to the Amazon EC2 purchase options, the *IPP* describes the amount and relation of reserved and on-demand to be purchased to service the online application.

In its discrete form, a very similar problem can be found in related literature - the *BahnCard Problem*. The *BahnCard Problem,* an extension of the Ski Rental problem, originates from the ticket purchasing models offered by the German "Deutsche Bundesbahn" that offer travelers the opportunity to purchase a "*BahnCard*". Subsequently, the card entitles the traveler to a discount on their trips. This discount entitlement is valid for one year. The problem is at which point to buy a BahnCard, given an unforeseeable future. A traveler, who only travels with the train once, would be ill-advised to purchase a BahnCard, while a frequent traveler would be better off with one. (Fleischer 2001)

In both the IPP and the BahnCard Problem, this underlying uncertainty and whether the purchase of a reserved instance or a BahnCard is profitable in the future, is what makes it such an intriguing problem. It has also been generalized for dynamic TCP acknowledgement scenarios (Karlin et al. 2001) and capital investment scenarios (Azar et al. 1999).

Let $t = [0;T]$, be the set of phases and $x_r(t)$ the number of reserved instances active in phase t and $x_o(t)$ the amount of on-demand instances active in phase t, the IPP can be formalized as follows:

$$\min_{x_r} K := \sum_{j:t\le t_j}^{t+T_{res}} \left(y_r(t)c_r^{fix} + x_r(t)\cdot c_r^{var}\right) + \sum_{t=1}^{T}(x_o(t)\cdot c_o) \qquad \text{(Eq. 1)}$$

Subject to:

$$w(t) = \lambda_j(t)\cdot x_r(t) + x_o(t) \qquad \forall\, t \in T \qquad \text{(Eq. 2)}$$

$$\lambda_j(t) = \begin{cases} 1, & if\ j: t \le t_j \le t + T_{res} \\ 0, & else \end{cases} \qquad \text{(Eq. 3)}$$

$$y_r(t) = x_r(t) + x_r^{idle}(t) \qquad \forall\, t \in T \qquad \text{(Eq. 4)}$$

$$y_r(t) \ge y_r(t-1) \qquad \forall\, t \in T \qquad \text{(Eq. 5)}$$

(Eq. 1) shows the total cost function of the IPP problem to be minimized. It sums up all payments due for each hour (t) in which the reserved and on-demand instances are operated, plus the upfront payment of the installment fee, c_r^{fix}. c_r^{var} is the price per-hour of a reserved instance and c_o is the price of an on-demand instance. (Eq. 2) is the workload constraint which ensures that the workload $w(t)$ is met with a combination of reserved $x_r(t)$ and on-demand instances $x_o(t)$. We introduce a further binary variable $\lambda_j(t)$ which in combination with (Eq. 3) ensures that reserved instances are only active for a time period of T_{res} after initial purchase. For this model $T_{res} = 8760$ hours, since the reserved instance purchases are valid for exactly one year. (Eq. 4) introduces a further variable $y_r(t)$ which, together with (Eq. 5), ensures that already purchased reserved instances are carried over to the next period. Therefore, if $y_r(t) > x_r(t)$, this does not cause the stock of $x_r(t)$ to decrease. $x_r^{idle}(t)$ acts as a slack variable for the inequality to show in which t a reserved instance remained idle.

The IPP is the mathematically formal integer formulation of the decision problem, how many reserved instances to buy to minimize the total costs. As such, it optimally solves the problem. However, often optimality comes at the cost of computational complexity, as solving integer problems can often be

intractable. For this reason it is necessary to evaluate the complexity of the model. This leads us to the next proposition:

Proposition 2: *The discrete Integer Purchasing Problem (IPP) is solvable in polynomial time. (Proof moved to the Appendix. See Proof Sketch to Proposition 2)*

Since the IPP can be solved in polynomial runtime, we can proceed to evaluate the model for its decision support feasibility.

Numerical Evaluations

In the following section we present our evaluation results of the discrete IPP, but before we begin, we first introduce our methods used for data acquisition and simulation generation. Generally, two possible approaches can be found in literature when working with test instances. First, there are the practical use cases, which have high practical relevance in use, yet they do not follow any systematic structure required for rigorous analysis. As a result, an algorithm that performs well on one specific practical instance is not guaranteed to perform equally well on other instances. Second, there are artificial numeric simulations, generated randomly given predefined specifications. Their strength lies in the fact that fitting them to certain requirements such as given probability distributions poses no problem. They may however reflect situations with little or no resemblance to problem settings of practical interest. Hence, an algorithm performing well on several such artificial instances may or may not perform satisfactorily in practice.

Data Origin and Generation

In this work, we attempt to work with the best of both worlds using artificially enhanced practical use cases. More specifically, we use the Wikipedia trace to simulate the change in daily and weekly frequencies common for online services, and replicate the seasonal (annual) and long-term growth effects of demand on workloads using the Facebook and Amazon workload traces. The resulting synthetic workload process constitutes a time series composed out of seasonal components, each of which can be controlled individually. Additionally, for the function controlling the daily and weekly frequencies, we added a random uniformly distributed factor to create multiple versions for rigorous evaluation. Since we only have one data set available, this allows us to generate similar workload traces to better evaluate our discrete models.

Table 1. Workload Generator Parameters

Parameter	Range	Step
Growth	[-1.5;+1.5]	0.5
Frequency	[1;4]	0.5
Time Horizon	3 years	1 hour

Table 1 shows the various data generation parameters we used to enhance the available workload trace. The growth parameter is used to control the overall increasing or decreasing trend on workload frequencies. Looking at various page request sources from Alexa.com, an increase or decrease of 150% conveys a legitimate test basis. The frequency controls the volatility of the workload (i.e. the difference between the lowest and highest recorded workloads) with a frequency set at 1 representing the original trace. Therefore a frequency of four represents a workload with volatility four times as high. Figure 6 shows a sample of such a generated workload excerpt for one year. When viewed on a weekly basis, the workload sample is similar to the Wikipedia trace shown in Figure 6 (left).

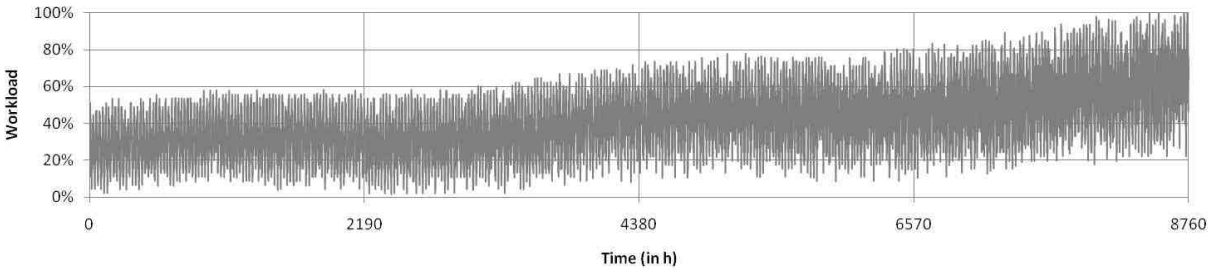

Figure 6. Synthetic Workload Sample

In this particular scenario, we generated a sample showing a gradual shape increase after the first quarter, followed by a further workload surge in the fourth quarter. Overall, the peak workload doubled (growth +1), while the volatility increased by 50% (frequency 1.5).

Solution Analysis

Using the data generation parameters described in Table 1, a total of 6x8 different Workload scenarios were created and evaluated for a timeframe of three years on a 24-hour step width. In other words, a workload was generated and evaluated with a 24-hour timeframe, again with a 48-hour timeframe, 72-hours and so on, up to three years (or 26,280 hours). For every scenario ten workload samples were generated to increase rigor and the average solutions of these ten "cases" were taken. The simulations were run with GAMS, using the CPLEX solver.

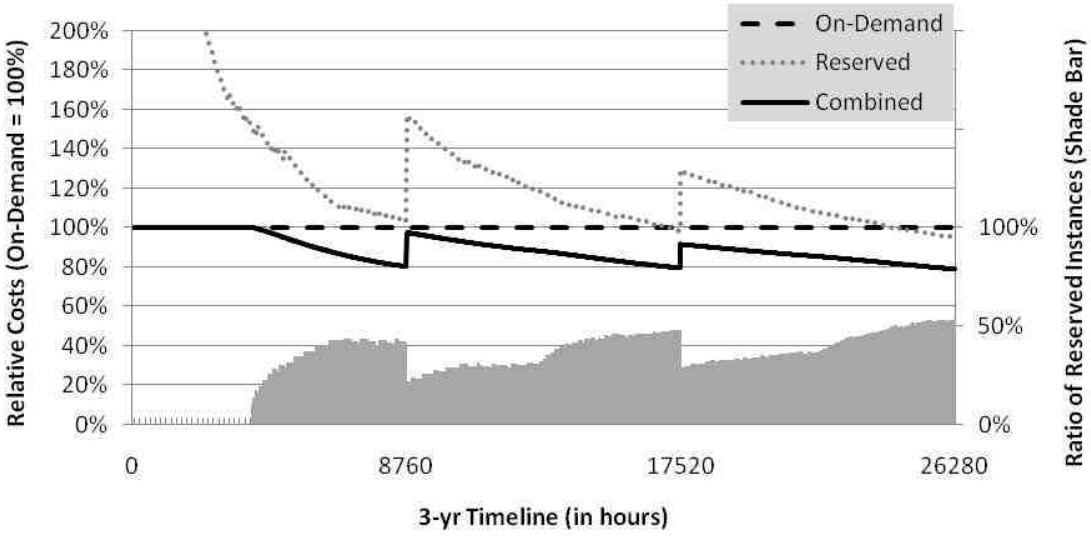

Figure 7. Operation Evaluation

Figure 7 shows the average cost evaluation results of the naïve operations as well as the optimal combined solution of all of our 48 scenarios. The dotted line shows the 24-hour average cost function for servicing the simulated workload using only reserved instances relative to the dashed function showing the average costs of servicing the workload using just on-demand instances. The solid line shows the cost behavior of the optimal combined portfolio using reserved and on-demand instances. The shaded bars at the bottom of Figure 7, scaled against the right vertical axis shows the share of reserved instances between [0;1] used to service the combined optimal operation of the simulated workloads. Again, a surge in costs is evident after the duration of one year, as the contracts are renewed in this period. Decisions to purchase reserved instances only become profitable after 164 days of operation on average as a startup. This does not mean that the first half year should be operated with on-demand instances. It only shows that if the operation horizon is less than half a year, reserved instances are more costly. Similar to the surge shown in the continuous model, the discrete model shows the same surge pattern after a year of operation. Also note that the optimal ratio of reserved instances to on-demand instances seems to increase gradually,

exceeding the 50% level after three years. As before, operating the workload using only reserved instances is inefficient for the first two years. As the simulations suggest though, after three or more years it becomes a viable option. Therefore, for longer time horizons the use of reserved instances should increase.

The above solutions are an ex-post analysis. It is necessary for insights into optimal behavior. As a strategic tool however, its uses are limited.

"Real-time" Operation Strategies

In the previous sections, we presented the continuous and discrete decision models and evaluations from an ex-post perspective. While knowing the optimal solution is all fine and well for an *ex-post* analysis of solutions, making choices on a daily basis makes using optimal *ex-post* algorithms infeasible. In reality when reacting to or anticipating a workload spike, a web service provider using purely IaaS services has one of two options to increase his computing infrastructure: either he purchases a further *reserved instance*, or a further *on-demand* instance. For daily provisioning, the choice will fall to on-demand instances, as they are only billed by the hour. Therefore, if decision makers purchase an additional instance at the wrong point in time, their mistake is short-lived. Long-term increases in workload, where the workload has increased on average for time horizons larger than one year are more destined to be served by reserved instances, as over the course of a year their operation is cheaper.

In this section we elaborate various operation strategies which can be applied in real-time for online decision-making as a direct result from our numerical evaluations. We define the operation strategy function *S(…)* which defines the configuration of reserved versus on-demand instances for the immediate future. In this work we look at three different strategies: the first involves a naïve strategy, where only one of the two pricing options is taken; the second strategy is reactive in nature where the future pricing decision is based on what was optimal in the past; the third strategy is forward looking in that it utilizes a workload forecasting model to decide what configuration is most likely to be the best in the future.

Naïve Operation

Naïve strategies include strategies based on the "one-size-fits-all" principle. The naïve strategies are therefore selected based on the expected time horizon for the future workloads. Based on our findings above:

Result 1: For time horizons less than nine months, the naïve strategy would be to pick only on-demand instances. If the time horizon is expected to be between nine months and a year, the strategy should be to use only reserved instances.

These strategies are additive for the number of years the time horizon is expected to be. In other words, if the time horizon is expected to be 18 months, on-demand instances should be chosen, while a time horizon of 21 months would require reserved instances. The decision set is therefore discontinuous as the value of $x_r(t)$ must be either $x_r(t) = 0$, or $x_r(t) = 1$ between two particular positive bounds. Therefore, based on result 1 we formulate our naïve operation strategy as follows:

Naïve Strategy: If $E(\tau)$ is the expected operation time horizon in months, k is the cyclic change observed in $E(\tau)$ and l and u are the lower and upper bounds where reserved instances are profitable (for the above sample $l = 9$ and $u = 12$), the naïve strategy can formally be described as:

$$S^{naive}(x_r, t, \tau) = \begin{cases} l + 12k \leq E(\tau) \leq u + 12k & x_r(t) = 1, \quad \forall\, t \in T \\ else & x_r(t) = 0, \quad \forall\, t \in T \end{cases} \qquad \text{(Eq. 6)}$$

Therefore, if the expected operation time horizon falls within the bounds, the naïve strategy is to use reserved instances, if not to use only on-demand instances.

Workload Forecast-based Operation

A very common approach to decision making is to base the choice of action in a reactive manner. The idea behind reactive strategies is that past observations will continue to occur in the future. When we look at the workload examples presented above, certain trends tend to repeat themselves. As we have learned

from the continuous model, the optimal purchase decision for reserved instances is highly dependent on the recurrent workload fluctuations.

Since, for the moment, we only have two purchase options available, the decision of how many reserved instances to purchase can also be seen as how many reserved instances do we need in relation to the amount of workload serviced by on-demand instances. Similar to the continuous model above for the percentage of reserved instances used, let this ratio be p. Following a training period, the optimal p^* can be calculated by viewing the past training period as an ex-post model and applying the discrete model. If x_r^* describes the optimal amount of reserved instances used in the training period, $p^*(t) = i(x_r(t))$ determines the percentage of reserved instances which should minimize the overall costs, while satisfying the service objective at all times.

Reactive Forecasting Strategy

The reactive operation strategy is purely observation-based. The intuition behind this strategy is: "*What was good for the past, is good for the future*". For each period *t,* the quantity decision $x_r(t)$ is made by looking back at a fixed interval τ, and basing the quantity decision on what would have optimally satisfied that interval. Formally, this reactive strategy can be defined as follows:

$$S^{reactive}(x_r, t, \tau) = \begin{cases} x_r(t) \leq avg\big(x_r(t-\tau)\big) & x_r(t) = avg\big(x_r(t-\tau)\big) \\ else & \emptyset \end{cases} \quad \text{(Eq. 7)}$$

The configuration of reserved instances will be increased if the current configuration $p(t)$ is not optimal for the training period τ. For our purposes as a strategic tool, the reactive strategy lacks the capability to include future events brought forward by other positive actions. For example, the reactive strategy cannot include information such as an increase in marketing expenses, or other events which might increase the workload in the future. For this reason we must enhance our reactive strategy with a workload prediction mechanism of some form or another.

Predictive Forecasting Strategy

While reactive operations are based on the past, predictive operations use forecasts to make a decision. Predictive operations are more concerned about future development based on forecasting models. Therefore purchasing reserved instances can be profitable if a considerable surge in workload frequency is expected for a lengthy period of time. Moreover, they allow adding certain shock parameters to be included in the model following certain management decisions. For example, if management decides more should be invested into marketing and advertisement, the resulting workload of the service is expected to increase. Likewise, some actions might negatively impact the workload.

The predictive operation strategy is based on current and future workload expectations. Where the length of time we looked back was τ in the reactive strategy, for the predictive strategy τ is the amount of periods we anticipate. Following a certain lead time, we are able to deduce the optimal ratio of reserved to on-demand instances using the discrete model presented above. Further, we add a $\beta(e)$ function to include the workload effect of expectations e which can be both positive and negative.

$$S^{predictive}(x_r, t, \tau) = \begin{cases} e \rightarrow \beta(e) \neq 0 & x_r(t) = avg\big(x_r(t-\tau)\big) + \beta(e) \\ else & x_r(t) = avg\big(x_r(t-\tau)\big) \end{cases} \quad \text{(Eq. 8)}$$

The predictive strategy aims to provide a cost-efficient level of reserved instances at any point in time. When workload processes are accurately forecast, the resulting infrastructure can be run at considerably low costs.

In the following section we evaluate the reactive and predictive strategy using the workload traces generated for the discrete model evaluations above.

Real-Time Strategy Evaluation

In this work we propose to use the reactive strategy introduced above to predict the baseline workload for the future plus an additional prediction mark-up to harbor modeling the expectations as our predictive workload forecast. This allows our forecast to directly include managerial inputs for effective long-run

planning. The workload forecasting method used is however not binding for the strategic investment model to function. The forecasting component of this work can be seen as a module and is replaceable with any other workload forecasting technique. For example, to include some short-term workload forecasting, most mechanisms in literature usually deliver point estimates for the workload in the near future. In other work (Hedwig, Malkowski, et al. 2010) we have proposed the use of distribution estimates for more sensitive operation strategies. Here, we focus on the long-term (mostly linear) prediction of workloads, where the fidelity of the short-term prediction mechanisms is no longer required.

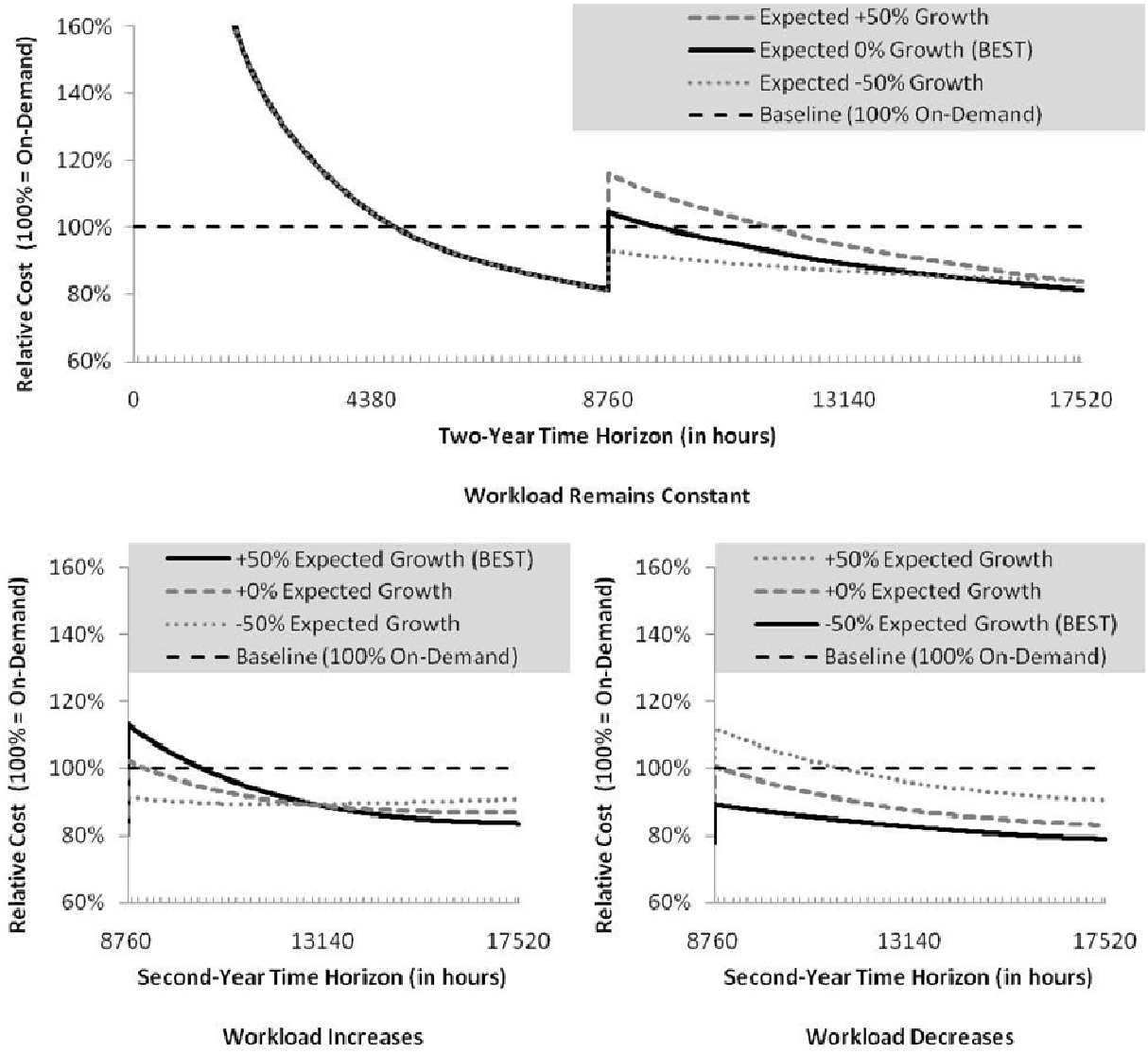

Figure 8. Strategy Evaluation

Figure 8 shows the evaluation of the strategic forecast strategy. We chose to use a two-year horizon since a purchase decision of a reserved instance is only valid for one year. As a result, the evaluation timeframe needn't exceed a year past the decision point. To evaluate the cyclic nature of the workload, we used a one year training period. Again the generated workloads used for the discrete model were used to evaluate the strategies. Here, the simulated traces were grouped by how their workload changed over time. As a result, we formed groups for traces displaying an increase, decrease and no change in workload. Figure 8 shows how the operation costs develop according to the purchase decision stipulated by the respective strategies depending on the development of the workload over time. For our evaluations we used the three strategies:

- In year two, buy 50% more reserved instances
- After one year, purchase the same amount of reserved instances
- In year two, buy 50% less reserved instances

than the optimal number of reserved instances needed to optimally serve the first year. For example, if we assume that purchasing 12 reserved instances was the best solution for *t*, then we should purchase either 18 (if we expect the workload to increase by 50%), 12 (if we expect the workload not to change) or 6 reserved instances in t+1, if we expect the workload to drop. Figure 8 (top) shows the evaluation of the three strategies when the workload trend does not change (0% growth). As can be expected, the best strategy in this case is to keep the number of reserved instances constant. Of interest is how the other two strategies work out. Since by the end of the second year the +50% and -50% growth strategies are roughly equal, we can deduce result 2:

Result 2: When the workload remains constant, it is better to underestimate the required number of reserved instances than to purchase too many reserved instances.

In other words, mistakenly underestimating the amount of required reserved instances is not as bad as overestimating them. While this may result in higher costs in some cases, it is a safe choice for risk averse decision makers. Figure 8 (bottom left and right) show the evaluation of the three strategies when the workload trend increases and decreases over time. Again, the most obvious and best strategy is to increase the number of reserved instances or decrease the number of reserved instances respectively. Looking at the remaining two strategies it is no surprise to observe that when the workload tends to increase (decrease), it is better to leave the number of reserved instances unchanged than to underestimate (overestimate) the required number of reserved instances. With a cost savings potential of up to 20% against the option to serve the workload using only on-demand instances, the strategies are on-par with the optimal solutions.

Managerial Implications

In this work we presented a novel set of coherent decision models for IaaS infrastructure investment decisions specifically as a tool for CIO's. In investigating the performance of these models we were able to extract some major implications for IT-managers. Recall our result that the optimal choice in the share of reserved instances was heavily dependent on the workload distribution (Result 1). As a result, a major factor to the cost efficiency of a chosen IaaS instance portfolio is the precision of the workload forecast. Additionally, due to the pricing model we found that the minimal cost configuration closely depends on the duration of the contracts. For the chosen workload sample based on the Wikipedia page access trace we conclude that the optimal share of reserved instances ranges between 50-65% to the highest recorded recurrent workload. However, these results were obtained using an ex-post analysis of a workload sample; a choice CIO's often do not have. At best, IT-Managers have a log of their past workloads (if at all) and a rough estimate of future workload levels.

For this reason we analyzed the effect of real-time decisions and their effect on various different future workload scenarios. As expected, if workload durations are short and continuous, opting for reserved instances, while still feasible, is an expensive choice. Conversely, only operating with on-demand instances where the workload duration surpasses 158 days is more costly than instead running with reserved instances. More interesting are our results for combined portfolios for volatile workload requirements. We found that mistakenly underestimating the amount of required reserved instances is not as costly as overestimating them. Keep in mind, that the underlying hardware is the same for both instances, so there is no operational difference between choosing on-demand and reserved instances. As a result, risk-averse decision makers would be well advised as shown in figure 8 to keep a lower ratio of reserved to on-demand instances (Result 2).

Conclusion

Today, the costs and personnel know-how to operate in-house IT infrastructures are continually increasing. At the same time, technological innovations in the service science industry – cloud computing in particular – have enabled new measures that promise a radical cost reduction potential. Instead of

building own datacenters, the IT infrastructure can be leased as a service using well-defined interfaces. In light of the multitude of available infrastructure services, we have investigated the economic implications of combining the pricing model for computing resources to create a cost minimal portfolio of computing instances. In a model-based approach, we derived a novel framework that allows us to assess the total costs of operating a portfolio of rented infrastructure consisting of reserved long-term and on-demand rented instances. This framework facilitates users to formulate sound, long-term investment strategies and as a result to operate at a cost minimal level.

The main contribution of this work is the novel set of coherent decision models for IaaS infrastructure investment decisions. These models allow the systematic economic analysis of the costs involved in leasing resources from IaaS providers when faced with highly volatile demand patterns. In our evaluation we brought forward the intertwined effects of highly volatile workload processes, system utilization and the costs involved when selecting the various pricing options. The continuous version of our model allows us to analyze the impact of an increase (or decrease) of a long-term share of resources (i.e. Amazons reserved instances). In the same effort we looked at the effect of duration on the operation costs. We found that the minimal cost configuration closely depends on the duration of the leasing contracts. To be deployed as a practical decision support tool, we reformulated the model as a discrete problem, which can be used to derive cost optimal investment strategies for practical use. Both the continuous and the discrete model formulations conclude that the optimal share of reserved instances ranges between 50-65% to the highest recorded recurrent workload. This ratio has proved beneficial to both models when compared to the naïve strategies to use only on-demand instances. The cost savings potential for the Wikipedia workload trace from 2010 and the simulated variations based on it, reached up to 20% (potentially up to ¼ million Euro's annually).

We proceeded to use the insights gained from the ex-post models in a strategic evaluation, where we set up a naïve strategy for decision makers more interested in a good choice for either long-term (i.e. reserved instances) or short-term (i.e. on-demand instances) resources. While this strategy was shown to be rather inefficient, its simplicity and straightforward support character could become useful for short-term investments (up to a year). For more long-term strategic support, we developed the reactive and predictive operation strategy. While a reactive strategy bases its decision on what was good for the past, the predictive strategy builds on the suggestion of the reactive strategy and adds exogenous trend expectations to the forecasted workload. This is important since reserved instances are paid for in advance. By allowing decision makers to add their future expectations, we show that our predictive strategic model performs equally well against the on-demand instance benchmark as our discrete model. We found that mistakenly underestimating the amount of required reserved instances is not as costly as overestimating them. This may have to do with the binding nature of reserved instances and the current high initial payment. While this may result in higher costs in some cases, it seems beneficial for risk-averse decision makers.

In future, we intend to expand the analysis using other real workload traces, as well as include spot instances, based on the idea of financial spot markets, to potentially increase the savings potential. A comparison in the form of a case-study to evaluate the economic difference of operating in-house architecture, or the above solution might also be of interest. Further, the model minimizes the total leasing payments; however, since the repeated year end peaks often cause high payments to be made to the IaaS provider, the optimization and balancing of the cash flow might be of higher significance from a capital supply perspective. By working with interest rates, we might be able to increase the model's significance from a venture capital perspective.

Appendix

Proof Sketch to Proposition 1 (Distribution Dependancy):

For any fixed point in time t, we can choose an optimal infrastructure size by minimizing the costs with respect to x. We redefine the cost function $c(x)$ as a function of the ratio of reserved instances $c(x_r;t)$, where c_o is the cost of purchasing an on-demand instance for one hour, c_r, the cost of purchasing a reserved instance, and c_i, the "loss" of not using a reserved instance due to the upfront payment option:

$$\underset{x_r}{Min}\, c(x_r;t) = c_0 \cdot f_o(x_r;t) + c_r \cdot f_r(x_r;t) + c_i \cdot f_i(x_r;t) \quad \text{(Eq. 9)}$$

Since $f_o(x_r;t) + f_r(x_r;t) + f_i(x_r;t) = 1$, we can rewrite the cost function $c(x)$ as a function of $f'(x_r;t)$

$$c(x_r;t) = \int_0^{x_r} \left(c_r \cdot f'(x_r;t) + c_i \cdot \left(1 - f'(x_r;t)\right)\right) dx_r + \int_{x_r}^{1} \left(c_o \cdot f'(x_r;t)\right) dx_r \quad \text{(Eq. 10)}$$

By deriving $c(x_r;t)$ by x_r and solving the derivative equation reveals the cost minimal instance size x_r^*. If the marginal resource utilization is equal to the given fraction of the cost factors,

$$\frac{\partial c(x_r;t)}{\partial x_r} = c_r \cdot f'(x_r;t) + c_i \cdot \left(1 - f'(x_r;t)\right) + c_o \cdot f'(1;t) - c_o \cdot f'(x_r;t) = 0 \quad \text{(Eq. 11)}$$

$$f'(x_r^*;t) = \frac{c_i}{c_i + c_o - c_r} \quad \text{(Eq. 12)}$$

Determining the inverse function of the marginal resource utilization function delivers the cost minimal reserved instance investment:

$$x_r^*(t) = f'^{-1}\left(t; \frac{c_i}{c_i + c_o - c_r}\right) \quad \text{(Eq. 13)}$$

As a result the cost minimal $x_r(t)$ is a function of the marginal resource utilization and the various instance prices. Since c_i, c_o and c_r are exogenously given we can rewrite equation 13 as:

$$x_r^*(t) = f'^{-1}(t;k) \quad \text{(Eq. 14)}$$

From equation 13 we can directly see that the cost minimal infrastructure size only depends on the marginal resource distribution and time. This solution is universal for any given marginal resource utilization function. A similar proof is given in (Hedwig, Bodenstein, et al. 2010). □

Proof Sketch to Proposition 2 (Model Complexity):

To prove that the discrete IPP is solvable in polynomial time, we transform the IPP problem as an instance of the shortest path problem, which is solvable in polynomial time. Let V define a set of vertices, whereby each vertex represents one valid configuration of x for every period.

$$v_{t,x_t} \qquad \forall\, t \in T, x_t \in [x_{min}, x_{max}] \quad \text{(Eq. 15)}$$

Let the set of edges between these vertices, v_t, be defined by E as the costs to use x instances in t.

$$E := < v_{t,x_t}, v_{t+1,x_t} >, \qquad \forall\, t \in T, x_t \in [x_{min}, x_{max}] \quad \text{(Eq. 16)}$$

As a result, finding the minimal cost of operating the workload is an instance of the shortest path problem, as we have to find the path through the network with the minimal costs. Therefore, the discrete optimal IPP has a polynomial runtime complexity. □

References

Amazon 2011. "Amazon Web Services Purchase Options - http://aws.amazon.com/de/ec2/purchasing-options/ - (01.05.2010)."

Azar, Y., Bartal, Y., Feuerstein, E., Fiat, A., Leonardi, S., and Rosén, A. 1999. "On Capital Investment," *Algorithmica* (25:1), pp. 22-36.

Azure 2011. "Windows Azure Pricing - http://windows.microsoft.com/windows/cloud - (01.05.2011)."

Bahli, B., and Rivard, S. 2005. "Validating measures of information technology outsourcing risk factors," *Omega* (22), pp. 175-187.

Bettis, R., Bradley, S., and Hamel, G. 1992. "Outsourcing and Industrial Decline," *Academy of Management Executive* (6), pp. 7-22.

Channabasavaiah, K., Tuggle, E. J., and Holley, K. 2003. *Migrating to a Service-Oriented Architecture Part 1 and 2*.

Fichman, R., and Kemerer, C. 2005. "The assimilation of software process innovations: an organizational learning perspective," *Management Science* (43:10), pp. 1345-1363.

Fleischer, R. 2001. "On the bahncard Problem," *Journal of Theoretical Computer Science* (268:1), pp. 161 - 174.

Foster, I., Yong, Z., Raicu, I., and Lu, S. 2008. "Cloud Computing and Grid Computing 360-Degree Compared," in *Grid Computing Environments Workshop, 2008. GCE '08*, pp. 1-10.

Gilley, M., and Rasheed, A. 2000. "making More by Doing Less: An Analysis of Outsourcing and its Effects on Firm Performance," *Journal of Management* (26:4), pp. 763-790.

GoGrid 2011. "GoGrid Pricing - http://www.gogrid.com/cloud-hosting/cloud-hosting-pricing.php - (01.05.2011)."

Gray, J. 2003. "Distributed Computing Economics," *Microsoft Research Tech Report 2003-24* (March).

Greco, J. 2001. "Outsourcing: the new partnership," *Journal of Business Strategy* (18:4), pp. 48-54.

Harmon, R., Demirkan, H., Hefley, B., and Auseklis, N. 2008. "Pricing Strategies for Information Technology Services: A Value-Based Approach," in *42nd Hawaii International Conference on System Sciences*, pp. 1-10.

Hedwig, M., Bodenstein, C., and Neumann, D. 2010. "Datacenter Investment Support System (DAISY)," in *Proceedings of 43rd Hawaii International Conference on System Sciences HICSS'10*.

Hedwig, M., Malkowski, S., Bodenstein, C., and Neumann, D. 2010. "Towards Autonomic Cost-Aware Allocation of Cloud Resources," in *Proceedings of the International Conference on Information Systems ICIS 2010*.

Jagannathan, S., Altmann, J., and Rhodes, L. 2003. "A Revenue-based Model for Making Resource Investment Decisions in IP Networks," in *IFIP/IEEE Eighth International Symposium on Integrated Network Management, 2003.*, pp. 185-197.

Johnson, M. 2003. "Colliding with Customers," in *Computerworld*.

Joyent 2011. "Joyent - http://www.joyent.com/ - (01.05.2011)."

Karlin, A., Kenyon, C., and Randall, D. 2001. "Dynamic TCP acknowledgement and other stories about e/(e-1)," in *STOC '01 Proceedings of the thirty-third annual ACM symposium on Theory of computing*.

Ketler, K., Kliger, D., and Vale, B. 1993. "The Outsourcing Decision," *International Journal of Information Management* (13:6), pp. 449-459.

Khajeh-Hosseini, A., Greenwood, D., and Sommerville, I. 2010. "Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS," in *IEEE 3rd International Conference on Cloud Computing, 2010*, pp. 450-457.

Lacity, M., Khan, S., and Willcocks, L. 2009. "A review of the IT outsourcing literature: Insights for practice," *Journal of Strategic Information Systems* (18:3), pp. 130-146.

Lacity, M., and Hirschheim, R. 1993. *Information Systems Outsourcing: Myths, Metaphors and Realities*, John Wiley & Sons, Inc. New York, NY, USA.

Lee, J.-N., Huynh, M. Q., Kwok, R. C.-W., and Pi, S.-M. 2003. "IT outsourcing evolution ---: past, present and future," *Communications of the ACM - Wireless Networking Security* (46:5), pp. 84-89.

Mituzas, D. 2011. "Wikistats - http://dammit.it/wikistats."

Motahari-Nezhad, H., Stephenson, B., and Singhal, S. 2009. "Outsourcing Business to Cloud Computing Services: Opportunities and Challenges," *HP Laboratories Report, Submitted to IEEE Internet Computing, Special Issue on Cloud Computing*.

Padala, P., Shin, K. G., Zhu, X., Uysal, M., Wang, Z., Singhal, S., et al. 2007. "Adaptive Control of Virtualized Resources in Utility Computing Environments," in *EuroSys '07 Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, pp. 289-302.

Progent Research 2002. "Small Business IT Outsourcing White Paper: Outsourcing Advantages. How Small Businesses can Benefit from Outsourcing IT Services," *White Paper, Progent Corporation*.

Rackspace 2011. "Rackspace - http://www.rackspace.com/index.php - (01.05.2011)."

Ragusa, C., Longo, F., and Puliafito, A. 2008. "On the Assessment of the S-Sicilia Infrastructure: A Grid-Based Business System," *Grid Economics and Business Models, Lecture Notes in Computer Science* (5206), pp. 113-124.

Risch, M., and Altmann, J. 2008. "Cost Analysis of Current Grids and Its Implications for Future Grid Markets," in *GECON '08 Proceedings of the 5th international workshop on Grid Economics and Business Models*, pp. 13-27.

SearchCloudComputing.com 2010. "Top Cloud Computing Providers 2010."

Strassmann, P. 2004. "Most outsourcing is still for losers," in *Computerworld*.

Verizon 2011. "Verizon Pricing - http://www.verizonbusiness.com/Products/it/ - (01.05.2011)."

Weigelt, C. 2009. "The Impact of Outsourcing New Technologies on Integrative Capabilities and Performance," *Strategic Management Journal* (30:6), pp. 595-616.