To Move or Not to Move: The Economics of Cloud Computing

Byung Chul Tak   Bhuvan Urgaonkar   Anand Sivasubramaniam
The Pennsylvania State University

Abstract

Cloud-based hosting promises cost advantages over conventional in-house (on-premise) application deployment. One important question when considering a move to the cloud is whether it makes sense for 'my' application to migrate to the cloud. This question is challenging to answer due to following reasons. Although many potential benefits of migrating to the cloud can be enumerated, some benefits may not apply to 'my' application. Also, there can be multiple ways in which an application might make use of the facilities offered by cloud providers. Answering these questions requires an in-depth understanding of the cost implications of all the possible choices specific to 'my' circumstances. In this study We identify an initial set of key factors affecting the costs of a deployement choice. Using benchmarks representing two different applications (TPC-W and TPC-E) we investigate the evolution of costs for different deployment choices. We show that application characteristics such as workload intensity, growth rate, storage capacity and software licensing costs produce complex combined effect on overall costs. We also discuss issues regarding workload variance and horizontal partitioning.

1 Introduction

Cloud-based hosting promises several advantages over conventional in-house (on-premise) application deployment. (i) Ease-of-management (although arguments against this have also been made [6]): since the cloud provider assumes management-related responsibilities, the customer is relieved of this burden and can focus on its core expertise. (ii) Cap-ex savings: it eliminates the need for purchasing infrastructure; this may translate into lowering the business entry barrier. (iii) Op-ex reduction: elimination of the need to pay for salaries, utility electricity bills, real-estate rents/mortgages, etc. One oft-touted aspect of Op-ex savings concerns the ability of customer's Op-ex to closely match its evolving resource needs (via usage-based charging) as opposed to depending on its worst-case needs.

The quintessential question when considering a move to the cloud is: should I migrate *my* application to the cloud? Whereas there have been several studies into this question [1, 7, 14], there is no consensus yet on whether the cost of cloud-based hosting is attractive enough compared to in-house hosting. There are several aspects to this basic question that must be considered. First, although many potential benefits of migrating to the cloud can be enumerated for the general case, some benefits may not apply to my application. For example, benefits related to lowered entry barrier may not apply as much to an organization with a pre-existing infrastructural and administrative base. Second, there can be multiple ways in which an application might make use of the facilities offered by a cloud provider. For example, using the cloud need not preclude a continued use of in-house infrastructure. The most cost-effective approach for an organization might, in fact, involve a combination of cloud and in-house resources rather than choosing one over the other. Third, not all elements of the overall cost consideration may be equally easy to quantify. For example, the hardware resource needs and associated costs may be reasonably straightforward to estimate and compare across different hosting options. On the other hand, labor costs may be significantly more complicated: e.g., how should the overall administrators' salaries in an organization be apportioned among various applications that they manage? Answering these questions requires an in-depth understanding of the cost implications of all the possible choices specific to *my* circumstances. Given that these answers can vary widely across applications, organizations, and cloud providers, we believe *the best way is to explore various applications case-by-case in an attempt to draw generalities or useful rule-of-thumbs.* This paper represents our first step towards this endeavor and we make the following contributions.

- We identify an initial set of key factors affecting the costs of a deployment choice (in-house, cloud, and combinations). We classify these as “quantifiable” and “less quantifiable” based on how amenable they are to precise quantification. We also classify costs into the well-regarded “direct” and “indirect” categories: the former’s contribution to the cost can be easily traced and accounted (e.g., server costs) while the latter’s contribution may be ambiguous and require more meticulous accounting (e.g., cooling costs).

- Besides the two extreme deployment choices of pure in-house and pure cloud-based hosting available to an application, we identify a spectrum of hybrid choices that can offer the best of both worlds. Our hybrid choices capture both “vertical” and “horizontal” ways of partitioning an application, each with its own pros and cons.
- Using benchmarks representing two different “commercial-like” applications (built using open-source vs. licensed software), cloud offerings (IaaS vs. SaaS), and workload characteristics (stagnant vs. growing, “bursty” or otherwise) we study the evolution of costs for different deployment choices.

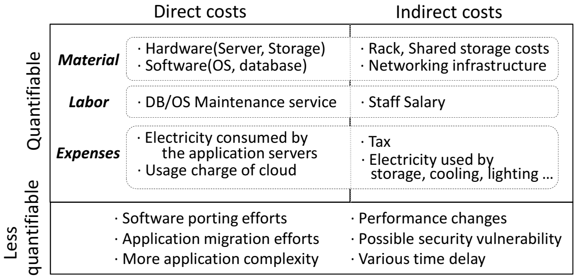
2 Background and Overview

2.1 Net Present Value

The concept of *Net Present Value* (NPV) is popularly used in financial analysis to calculate the profitability of an investment decision over its expected lifetime considering all the cash inflows and outflows. Walker et al. have recently employed this concept in their work, focusing mainly in separately exploring the feasibility of renting computing [14] and storage [15] from the cloud. While we employ the same NPV concept, we go beyond this work: (i) as opposed to comparing rental vs. in-house costs only for a given hardware base, we compare the costs for hosting specific workloads (ii) we incorporate additional costs such as software licenses, and electricity, (iii) we study the impact of workload evolution/variance and cloud models (IaaS vs. Saas), and finally (iv) we consider combinations of in-house and cloud hosting. Borrowing existing notation, we define the NPV of an investment choice spanning Y years into the future as: $NPV = \sum_{t=0}^{Y-1} \frac{C_t}{(1+r)^t}$ where r is the *discount rate* and C_t the cost at time t. The role of the discount rate is to capture the phenomenon that the value of a dollar today is worth more than a dollar in the future, with its value decreased by a factor $(1+r)$ per year.

2.2 Cost Components

Figure 1 presents our classification of costs. Certain cost components are less easy to quantify than others, and we use the phrases “quantifiable” and “less quantifiable” to make this distinction. Examples of less quantifiable costs include effort of migrating an application to the cloud, porting an application to the programming API exposed by a cloud (e.g., as required with Windows Azure), time spent doing the migration/porting, any problems/vulnerabilities that arise due to such porting or migration, etc. Adhering to well-regarded convention in financial analysis, we also employ the classification of costs into the “direct” and “indirect” categories based on their ease of traceability and accounting. If a cost can be clearly traced and accounted to a product/service/personnel, it is a direct cost, else it is an indirect cost. As shown in Figure 1, examples of direct cost include hardware & software costs; examples of indirect cost include staff salaries. It should be noted that certain costs may be less quantifiable yet direct (e.g., porting an application). Similarly, certain costs may be quantifiable yet indirect (e.g., staff salaries, cooling, etc.) In this work, we restrict our focus to only quantifiable costs and leave less quantifiable costs for future work.



		Direct costs	Indirect costs
Quantifiable	*Material*	· Hardware(Server, Storage) · Software(OS, database)	· Rack, Shared storage costs · Networking infrastructure
	Labor	· DB/OS Maintenance service	· Staff Salary
	Expenses	· Electricity consumed by the application servers · Usage charge of cloud	· Tax · Electricity used by storage, cooling, lighting …
Less quantifiable		· Software porting efforts · Application migration efforts · More application complexity	· Performance changes · Possible security vulnerability · Various time delay

Figure 1: Classification of costs related to migration.

2.3 Application Hosting Choices

Besides pure in-house and pure cloud-based hosting, a number of intermediate/hybrid options have been suggested, and are worth considering [4]. We view these schemes as combinations of different degrees of “vertical” and “horizontal” partitioning of the application. Vertical partitioning splits an application into two subsets (not necessarily mutually exclusive) of components - one is hosted in-house and the other migrated to the cloud - and may be challenging if any porting is required [4]. Horizontal partitioning replicates some components of the application (or the entire application) on the cloud along with suitable workload distribution mechanisms. Such partitioning is already being used as a way to handle unexpected traffic bursts by some businesses (e.g., KBB.com and Domino’s Pizza [16]). Such a partitioning scheme must employ mechanisms to maintain consistency among replicas of stateful components (e.g., databases) with associated overheads [1]. Given myriad cloud providers and hosting models (we consider IaaS and SaaS), there can be multiple choices for how a component is migrated to the cloud, each with its own cost

[1] Note that pure in-house and pure cloud hosting can be viewed as extreme cases of both these kinds of partitioning.

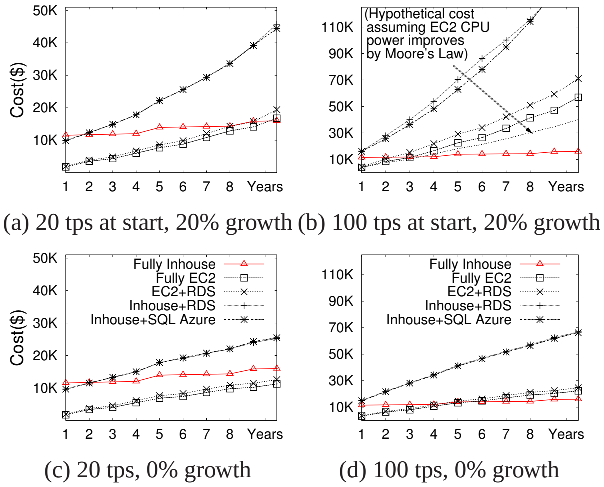
implications. In this work, we choose three such options (in addition to pure in-house and pure cloud-based) that we described next.

2.4 Our Methodology: A Brief Outline

We consider hosting options offered by two prominent cloud providers: Amazon and Windows Azure, including both IaaS (EC2 instances) and SaaS options (Amazon RDS and SQL Azure). We consider the following five hosting options: (i) fully in-house, (ii) fully EC2 (the entire application is migrated to Amazon cloud within appropriately provisioned EC2 instances), (iii) EC2+RDS (similar to fully EC2 except the database which uses Amazon's RDS SaaS), (iv) in-house+RDS (a vertical partitioning where the database is migrated to Amazon's cloud to use its RDS SaaS while the remaining components are in-house), and (v) in-house+SQL Azure (a vertical partitioning similar to (iv) with RDS replaced with Microsoft's SQL Azure SaaS). We compare these hosting options for the following two applications from TPC [12]: (1) TPC-W (a benchmark that emulates an online bookstore) and (2) TPC-E (a benchmark that emulates online transaction processing in a brokerage firm). We assume that TPC-W is built using open-source software components (Apache, JBoss, Mysql) except for the OS (Windows), whereas TPC-E uses licensed software (SQL Server 2008 and Windows Server 2008). Both applications have three tiers: Web, Java-based application logic, database.

Our cost comparisons require us to make a number of projections/assumptions. We allow for a function that describes workload growth over time (increasing, decreasing, or stagnant in its form) and incorporate this into our NPV calculation. We incorporate both hardware and software upgrades to up-to-date products at typical refresh cycles (4 years for both hardware and software). We project CPU, memory capacities based on Moore's Law (similar to [14]). E.g., we assume CPU speed doubles every two years.

Finally, we need to estimate the hardware needs of our applications for a range of workload intensities (expressed in transactions/second or tps). Our goal is to find configurations across hosting options that offer similar, satisfactory performance. We discuss the salient aspects of our estimation here and present the details in a technical report [8]. By running TPC-W on machines in our lab (each containing Intel Xeon 3.4GHz dual-processor with 2GB DRAM), we identify *marginal throughput gains* offered by adding an extra server (and CPU) to a tier. Using microbenchmarks, we determine cloud instance configurations that offer "comparable" computing power and memory (encouragingly our results match well with existing work that has benchmarked RDS [5]). Since EC2 instances come in much smaller sizes, a comparable in-cloud configuration has a larger number of VMs. E.g., the EC2 *small* instance has an effective CPU of 1.1 GHz implying each of our lab machines is equivalent to about three of these. For TPC-E, we are unable to carry out a benchmarking-based estimation since we do not have the license for a MS SQL server. Instead, we employ performance and cost results offered for TPC-E on the TPC Web site for a number of machine, network, and storage configurations [12]. We note that the general problem of modeling resource needs is non-trivial with extensive work for in-house (including ours [13]) and emerging work for the cloud [11], and incorporating such estimates into our costs may be a useful future direction.



(a) 20 tps at start, 20% growth (b) 100 tps at start, 20% growth

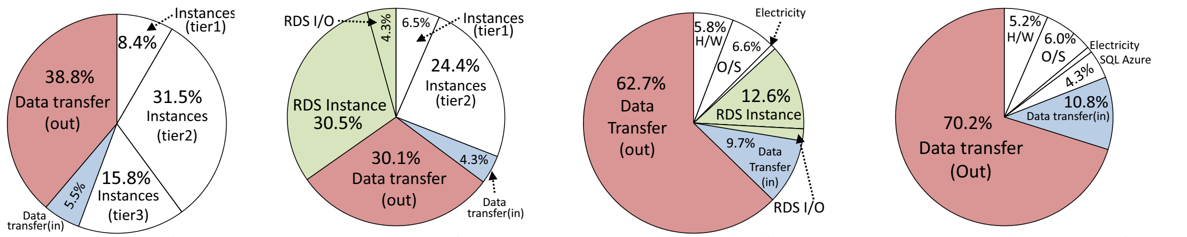
(c) 20 tps, 0% growth (d) 100 tps, 0% growth

Figure 2: NPV over a 10 year time horizon for TPC-W. We consider small (20 tps at t=0) and medium workload intensity (100 tps), as well as stagnant and 20% growth.

3 Our NPV Analysis: Key Results

3.1 Workload Intensity and Growth

Figure 2 presents NPV calculations for up to a 10 year time horizon for TPC-W. We present results with two workload intensities at the beginning: (i) 20 tps and (ii) 100 tps, which represent "small" and "medium" in the overall spectrum we consider [8]. We also present two intensity growth scenarios: (i) stagnant and (ii) 20% increase per year; we have also considered other growth rates. As the workload intensity grows, TPC-W requires more servers and higher IO bandwidth but its storage capacity needs do not change. Overall, we find that *in-house provisioning is cost-effective for medium to large workloads, whereas cloud-based options suit small workloads.* For small workloads, the servers procured for in-house provisioning end up having significantly more capacity than needed (and they remain under-utilized) since they are the lowest granularity servers available in market today. On the other hand, cloud can offer instances matching the small workload needs (due to



(a) Fully EC2 ($22K) (b) EC2+RDS ($29K) (c) In-house+RDS ($70K) (d) In-house+SQL Azure ($63K)

Figure 3: Closer look at cost components for four cloud-based application deployment options at 5th year. Initial workload is 100 tps (Transaction Per Second) and the annual growth rate is 20%.

the statistical multiplexing and virtualization it employs). For medium workload intensity, cloud-based options are cost-effective only if the application needs to be supported for 2-3 years, and become expensive for longer-lasting scenarios. These workload intensities are able to utilize well provisioned servers making in-house procurement cost-effective.

An interesting trend is the *significantly slower NPV increase for in-house compared to cloud-based options*, which may be partly explained as follows. Since we assume hardware capacity growing according to Moore's Law, unless the workload growth matches or exceeds this rate, the number of servers required in-house will actually shrink each year. However, things evolve differently with cloud-based options. The computing power as well as price of a cloud instance are intentionally *engineered* to be at a certain level (via virtualization and statistical multiplexing) even though cloud providers may upgrade their hardware regularly (just as in-house). E.g., since the start of EC2 in 2006, the computing power/memory per instance has remained unchanged while there has been only one occasion of instance price reduction. In other words, while in-house hosting enjoys improvement in performance/$ with time, trends over the last 5 years suggest that the performance/$ offered by the cloud has remained unchanged [2]. Even if we assume the performance/$ offered by the cloud improves with time (say, an instance of given capacity becomes cheaper over time), cloud-based provisioning still remains expensive in the long run since data capacity and transfer costs contribute to the costs more significantly than in-house (See Figure 2(b)).

3.2 Data Transfer, Storage Capacity, Software Licenses

We illustrate in Figure 3 detailed breakdowns of NPV for five-year long hosting of TPC-W for options involving the cloud (i.e., options (ii)-(v) from Section 2). Overall, we find that *data transfer is a significant contributor to the costs of cloud-based hosting* - between 30%-70% for TPC-W. This suggests that *vertical partitioning choices may not be appealing for applications that exchange data with the external world.* Data transfer costs in Figure 3(c),(d) are larger than those in Figure 3(a),(b) because traffic per transaction between Jboss and MySQL (16KB/tr) is larger than between clients and Apache (3KB/tr).

Another determining factor to costs with cloud-based hosting can be storage capacity. Whereas TPC-W poses relatively small costs for storage capacity (its database only needs a few GB and its storage capacity costs do not even show up in Figure 3), TPC-E has significant data storage needs (about 4.5TB). Figure 4 presents the NPV evolution for TPC-E for two initial intensities - 300 tps (medium) and 900 tps (high) with 20% annual growth rate. We only present "Fully in-house" and two cloud ("Fully EC2" and "EC2+SQL server") options since we have already established the high costs of vertical partitioning. We find that in-house provisioning for TPC-E has to make significant investments in high-end RAID arrays (gap A), that constitute about 75% of overall costs. For initial workload intensity of 300 tps, these costs go down substantially with "Fully EC2" (i.e., renting storage from EC2 is cheaper than the amortized cost of procuring this much storage in-house), causing the overall costs to improve by 50% (year 1, shown as gap A) and 28% (year 6, shown as gap B in Figure 4).

The software licensing fee for SQL Server and Windows can also be a significant contributor to TPC-E costs: second (17.4% of overall) and largest (67%) contributor, respectively, for "Fully in-house" and "Fully EC2" options. Using *pay-per-use* SaaS DB allows the elimination of SQL Server licensing fees (shown as gap C in Figure 4) and results in even better costs. *SaaS options can be cost-effective for applications built using software with high licensing/maintenance fee*. Note that these concerns did not arise with TPC-W which employed open-source software, implying a different ordering of cost-efficacy among options.

It is also worth comparing the cost evolution for the two intensities in Figure 4. With medium intensity (300tps), in-house option is less attractive than cloud-

[2] It is important to remember that the improvements in performance/$ for in-house accrue due to investments made in upgrades.

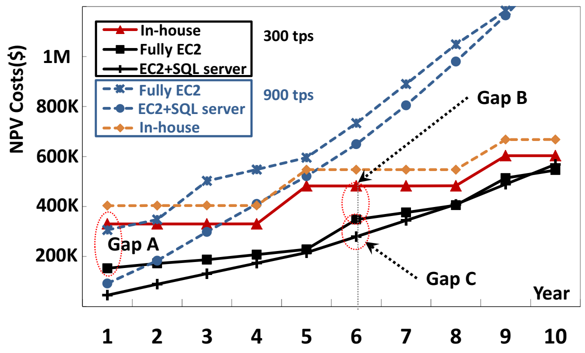


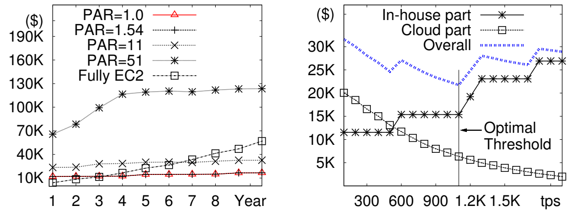
Figure 4: Two sets of TPC-E results at initial workload of 300 tps and 900 tps.

based options for the entire 10 year period without ever having a cross-over. However, at the higher intensity (900tps), cloud-based options quickly (after 2 years for “Fully EC2” and after 4 years for “EC2+SQL server”) become more expensive than in-house. This is qualitatively similar to the observations for TPC-W. However, cloud-based options remain attractive for a larger range of workload intensity than for TPC-W (compare Figure 4 with Figure 2(b) both of which have the same growth rate but differ in intensity by a factor of 9) - the key reasons for this difference are gaps B and C, i.e., the higher storage costs for in-house TPC-E as well as the contribution of software licenses in non-SaaS options.

A final interesting phenomenon arises due to the following: when buying cloud instances for TPC-E database, we do find instances that offer required computational power per core but limited in total number per instance. The most powerful instance has 8 virtual cores of 3.6Ghz clock speed whereas In-house server used in the analysis uses 12 cores. This forces the cloud-based options to procure more number of instances than in-house. In addition to this, the pricing policy of charging the license fee per virtual core drastically increases required SQL server licenses to purchase (since Microsoft charges only for the physical cores in non-virtualized environment). This suggests that *a reconsideration of software licensing structures, particularly as applicable to large-scale parallel machines, may be worthwhile for making cloud-based hosting more appealing.*

3.3 Workload Variance and Cloud Elasticity

Our cost analysis so far were based on *average* workload intensities. Given high burstiness (i.e., high peak-to-average ratio or PAR) in many real workloads, it is common in practice to provision close to the *peak*. Whereas in-house provisioning must continue this practice, the usage-based charging and elasticity offered by the cloud open new opportunities for savings (for both in-cloud and horizontal partitioning). We investigate costs of variance-aware provisioning for three degrees of burstiness corresponding to time-of-day effects and flash crowds. Researchers have reported the magnitude of daily workload fluctuations to be in the 40%-50% range for social networking applications, and about 70% for e-commerce Web site. Flash crowds can cause orders of magnitude higher peaks than the average and become a particularly appealing motivation for considering the use (perhaps partial) of cloud. We choose PAR of 1.54 (min=40, max=135 tps) to represent daily variations and PAR values of 11 and 51 to represent two flash crowd scenarios (i.e., peak of 10 and 50 times the average, respectively).



(a) In-house Cost Change (b) Horizontal Partitioning

Figure 5: Effect of workload variance and horizontal partitioning on in-house cost.

Figure 5(a) illustrates the effect of three levels of burstiness on the in-house provisioning cost. We select the case of in-house with medium & increasing workloads (Figure 2(b)). Provisioning for the diurnal fluctuation of 70% (PAR=1.54) does not impact the cost whereas flash crowd noticeably increases costs. Provisioning for PAR=51 shifts the cross-over point with “Fully EC2” from year 2.5 to year 10. The reason why diurnal fluctuation does not affect the cost is because provisioned servers already have enough capacity to embrace the peak of diurnal fluctuation. But, provisioning for flash crowds can substantially increase the cost.

We explore the benefits offered by a horizontal partitioning scheme that sets a threshold of workload intensity over which we create a replica in the cloud to handle the excess. Fig. 5(b) shows the cost change over a range of threshold at year 1. We assume a lognormal(μ:0,σ:1) distribution (mean:500tps) to simulate the bursty traffic. Blue dotted line in Fig. 5(b) is the overall cost, the sum of two components - in-house and cloud part. As the threshold moves to higher workload intensity, in-house cost rises in order to acquire more capacity, and the cloud cost lessens since the probability of overflowing the in-house server capacity diminishes. The equilibrium point where the cost is minimum is found at 1100 tps. This suggests that *horizontal partitioning can be effectively used to eliminate the cost increase from provisioning for the peak*.

4 Related Work

Walker [14] has looked at issues related to the economics of purchasing or leasing CPU hours using the NPV concept. The focus of his work is to provide a methodology that can aid in deciding whether to buy or lease the CPU capacity from the organization's perspective. His analysis ignores application-specific intricacies. For example, in calculating the cost of leasing the CPU hours from Amazon EC2, total required CPU hours is assumed to be statically fixed. Similarly, Walker et al. [15] also studied the problem of using storage cloud vs. purchasing hard disks. Our study differs from these in a sense that we try to address the question at the level of individual applications.

Gray [3] has looked at economics in the context of distributed computing and he came up with the amount of resource users can buy with one dollar in the year 2003. He found that since data transfer costs are non-negligible for Web-based applications, it is economical to optimize the application towards reducing data transfer. Armbrust et al. [1] have extended the cost analysis of Gray's data into the year 2008 and presented how the cost of each resource type evolved at different rate. They have also pointed out that cost analysis can be complicated due to cost factors such as power, cooling and operational costs, which are in many cases difficult to quantify.

In our prior work we have also addressed some economic issues of cloud migration as they apply to digital library and search systems such as CiteSeer [10, 9]. CloudCmp [7] attempts to develop a set of benchmarks that allow users to compare various cloud providers and select the most economical ones for their applications. Campbell et al. [2] carry out simple calculations to determine the break-even utilization point for owning vs. renting the system infrastructure for a medium-sized organization.

5 Conclusions and Future Directions

In this study we have investigated the migration costs of several deployment options using popular benchmarks. We have shown that application characteristics such as workload intensity, growth rate, storage capacity and software licensing costs produce complex combined effect on overall costs. We have also briefly explained issues regarding workload variance and horizontal partitioning. Overall, we find that (i) complete migration to today's cloud is appealing only for small/stagnant businesses/organizations, (ii) vertical partitioning options are expensive due to high costs of data transfer, and (iii) horizontal partitioning options can offer the best of in-house and cloud deployment for certain applications.

Our work opens up interesting possibilities for future work. We would like to incorporate indirect costs (and also less quantifiable costs in some meaningful way). As immediate work, we are extending our study to a broader set of applications such MapReduce and undergraduate labs at Penn State.

Acknowledgments

This research was supported, in part, by NSF grant CNS-0720456, NSF CAREER award CNS-0953541, and NSF grant 0615097.

References

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., and Zaharia, M. Above the clouds: A berkeley view of cloud computing, Feb 2009.

[2] Campbell, R., Gupta, I., Heath, M., Ko, S. Y., Kozuch, M., Kunze, M., Kwan, T., Lai, K., Lee, H. Y., Lyons, M., Milojicic, D., O'Hallaron, D., and Soh, Y. C. Open cirrustmcloud computing testbed: federated data centers for open source systems and services research. In *Proceedings of the 2009 conference on Hot topics in cloud computing* (Berkeley, CA, USA, 2009), HotCloud'09, USENIX Association, pp. 1–1.

[3] Gray, J. Distributed computing economics. *Queue 6* (May 2008), 63–68.

[4] Hajjat, M., Sun, X., Sung, Y.-W. E., Maltz, D., Rao, S., Sripanidkulchai, K., and Tawarmalani, M. Cloudward bound: planning for beneficial migration of enterprise applications to the cloud. In *Proceedings of the ACM SIGCOMM 2010 conf.*

[5] Kossmann, D., Kraska, T., and Loesing, S. An evaluation of alternative architectures for transaction processing in the cloud. In *Proceedings of the 2010 international conference on Management of data* (New York, NY, USA, 2010), SIGMOD '10, ACM.

[6] Kotsovinos, E. Virtualization: Blessing or curse? *Queue 8*, 40:40–40:46.

[7] Li, A., Yang, X., Kandula, S., and Zhang, M. Cloudcmp: comparing public cloud providers. In *Proceedings of the 10th annual conf. on Internet measurement* (New York, NY, USA, 2010).

[8] Tak, B., Urgaonkar, B., and Sivasubramaniam, A. To Move or Not to Move: The Economics of Cloud Computing. Tech. rep., Penn State Univ. `http://csl.cse.psu.edu/papers/tr_cse_11_002.pdf`.

[9] Teregowda, P., Urgaonkar, B., and Giles, C. L. Citeseerx: A cloud perspective. In *Proceedings of the Second USENIX Workshop on Hot Topics in Cloud Computing* (Jun 2010).

[10] Teregowda, P., Urgaonkar, B., and Giles, C. L. Cloud computing: A digital libraries perspective. In *Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing* (Washington, DC, USA, 2010), CLOUD '10, IEEE Computer Society, pp. 115–122.

[11] Tickoo, O., Iyer, R., Illikkal, R., and Newell, D. Modeling virtual machine performance: challenges and approaches. *SIGMETRICS Perform. Eval. Rev. 37* (January 2010), 55–60.

[12] Transaction Processing Performance Council. `http://www.tpc.org/`.

[13] Urgaonkar, B., and Chandra, A. Dynamic provisioning of multi-tier internet applications. In *Proceedings of the ICAC 2005*.

[14] Walker, E. The real cost of a cpu hour. *Computer 42* (April 2009).

[15] Walker, E., Brisken, W., and Romney, J. To lease or not to lease from storage clouds. *Computer 43* (April 2010), 44–50.

[16] Windows Azure Lessons Learned. `http://channel9.msdn.com/Blogs/benriga/`.