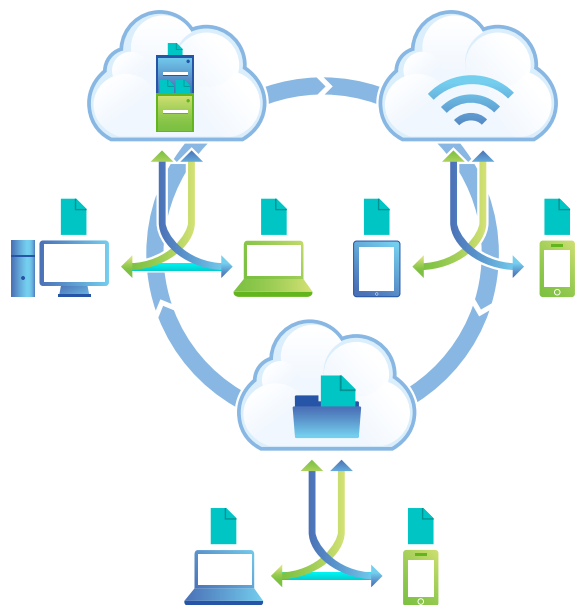


Understanding the Cost of Cloud:

Cost analysis of In-house vs. Cloud-based Hosting Options

By Byung Chul Tak, Bhuvan Uргаonkar, and Anand Sivasubramaniam



Will cloud-based hosting be economically feasible for any given application if it migrated into the cloud? It is not straightforward to answer this question.

Cloud-based hosting presents many advantages over traditional in-house (on-premise) hosting such as better scalability, ease of management, and cost savings. It is not difficult to understand how cloud-based hosting can be used to address some of the existing limitations and extend the capabilities of many type of applications. However, one of the most important questions is whether cloud-based hosting will be economically feasible for any given application if it migrated into the cloud. It is not straightforward to answer this question because it is not clear how an application will benefit from the claimed advantages, and, in turn, be able to convert them into tangible cost savings. Within cloud-based hosting offer-

Cloud-based hosting presents many advantages over traditional in-house (on-premise) hosting such as better scalability, ease of management, and cost savings.

ings, there is a wide range of hosting options one can choose from, each impacting the cost in a different way. Answering these questions requires an in-depth understanding of the cost implications of all the possible choices specific to an application's circumstances.

In this study we identify a diverse set of key factors affecting the costs of deployment choices. Using benchmarks representing two different applications (TPC-W and TPC-E) we investigate the evolution of costs for different deployment choices. We consider important application characteristics such as workload intensity, growth rate, traffic size, storage and software license to understand their impact on the overall costs. We also discuss the impact of workload variance and cloud elasticity, and certain cost factors that are subjective in nature.

Application migration to the Cloud

Cloud-based hosting promises several advantages over conventional in-house (on-premise) application deployment. First, it offers ease-of-management since the cloud provider assumes management-related responsibilities, (e.g., procurement, upgrades, and maintenance of hardware/software), thus the customer is relieved of this burden and can focus on its core expertise. Second, it offers capex (capital expenditure) savings. It eliminates the need for purchasing the entire infrastructure. This may translate into lowering the business entry barrier for some organizations. Cloud's elasticity and high scalability enables customers to operate their applications at higher regions than it was possible with existing in-house infrastructure. Third, it can offer opex (operational expenditure) reduction. This can result from elimination of the need to pay for salaries, utility electricity bills, real-estate rents/mortgages, etc. One aspect of opex savings concerns the ability of customer's opex to closely match its evolving resource needs via usage-based charging as opposed to provisioning the worst-case needs. There are also other important benefits such as fast-time-to-market and potential increase in reliability.

The most fundamental question in regards to cost savings is whether it makes sense for any given application to move to the cloud. There has been a general belief that the cloud would be more economical due to pay-as-you-go pricing models and the cloud's capability to match the resource requirements. However, as more factors are taken into the picture, it becomes significantly difficult to answer this question. For example, it is not straightforward to estimate how many and what type of cloud instances to purchase as there is no information about how an application will perform on them. There is also an overwhelming number of cloud-based hosting options and cloud providers, each with different capacities and pricing schemes. For these reasons, there is no consensus yet on whether the cost of cloud-based hosting is efficient enough compared to in-house host-

Does it make sense for any given application to move to the cloud? As more factors are taken into the picture, it becomes significantly difficult to answer this question.

ing. There are several aspects to this 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. 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 vari-

Cost Analysis Methodology

Net Present Value: Investigating the suitability of an investment involves assessing the overall costs expected to be incurred over its lifetime. The decision of whether to migrate an application to the cloud can be viewed as an investment decision problem. 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. Borrowing existing notation, we define the NPV of an investment choice spanning T years into the future as: $NPV_T = \sum_{t=0}^{T-1} C_t / (1+r)^t$ where r is the discount rate and C_t the expenditure during t^{th} year. 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.

Cost Components: Hosting an application poses various types of costs including hardware, software costs and/or operational costs. Each cost type has its own idiosyncrasies requiring users to scrutinize them one by one in order to understand how, and in what circumstances, the cost is incurred. 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. "Quantifiable costs", on the other hand, can be accurately quantified and we further divide these into three sub-categories: material, labor and expenses. We also employ the classification of costs into "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, or personnel, it is classed as a direct cost, otherwise it is an indirect cost. As shown in Figure 1, examples of direct cost include hardware and software costs; examples of indirect cost include staff salaries.

Hosting Options: 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.

- Fully in-house: The entire application is hosted locally in the private hosting infrastructure.
- Fully EC2: The entire application is migrated to Amazon's EC2 cloud with components hosted within EC2 instances.
- EC2+RDS: Similar to Fully EC2 except the database, which uses Amazon's RDS.

ous applications that they manage?

Answering these questions requires an in-depth understanding of the cost implications of all the possible choices specific to the application's circumstances. Given that these answers can vary widely across applications, organizations, and cloud providers, the best way is to explore various applications case-by-case in an attempt to draw generalities or useful rule-of-thumbs. We first identify an initial set of key factors affecting the costs of deployment options. 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 "component based" and "workload-based" 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 present the evolution of costs for different deployment choices.

Summary of insights and further thoughts

Application characteristics such as workload intensity, growth rate, storage capacity and software licensing costs produce complex

		Direct costs	Indirect costs
More 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

- In-house+RDS: A component-based partitioning where the database is migrated to Amazon's cloud to use its RDS SaaS while the remaining components are in-house.
- In-house+SQL Azure: Similar to "In-house+RDS" with RDS replaced with Microsoft's SQL Azure SaaS.

We compare these hosting options for the following two applications: (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.

Refresh Cycles & Operational Period: We incorporate both hardware and software upgrades to up-to-date products at typical refresh cycles (4 years for both hardware and software). At the beginning of each new refresh cycle, we determine the size of hardware volume to purchase and also determine if a new software license is required for the new hardware. We also define the operational period which designates how long the owner of the application expects to operate the application. Due to many uncertainties in the fast-changing IT industry, it would be meaningless to consider the operation cycle of longer than ten years.

Determining Hardware Needs: 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 a similar, satisfactory performance.

In order to determine the number of machines required to meet the desired throughput, we empirically obtain the marginal throughput gain offered by adding an extra server to the application set-up when all other tiers are sufficiently provisioned. Once we have this information we can determine the total number of servers to achieve certain level of throughput by dividing it with the marginal throughput. From empirical measurements, we obtain the capacity of each of our machines for Jboss and Mysql tier to be 146.6 tps (Transaction Per Second) and 311.5 tps per machine. As for the Apache tier, the resource consumption was insignificant and we estimated from observed CPU utilization that one machine could handle about 4k tps.

For cloud-based hostings we need to find the number of cloud instances that are likely to offer the performance of TPC-W comparable with that offered in-house. Amazon EC2's small instance type claims to provide a computing power equivalent to 1.0-1.2 GHz CPU. It is known that Amazon EC2 multiplexes two VM instances on one physical core making it effectively 1.3GHz. We ran micro benchmarks to verify this and to establish computing power relative to our machines. EC2 small instance is found to operate at 1.09GHz, about 1/3 speed of our reference machine, which matches the claim of 1.0-1.2 GHz. Using this benchmark information we set the throughput limit of single EC2 small instance for the Jboss and Mysql tiers to be 57.34 tps (Transaction Per Second) and 121.86 tps, respectively.

Cost Analysis

Workload Intensity and Growth

Figure 2 presents NPV cost 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) 500 tps, which represent "small" and "high" in the overall spectrum we consider. Workload growth is projected to be 20% increase per year. Overall, we find that in-house provisioning is cost-effective for high workloads, whereas cloud-based options suit small workloads. For small workloads, the servers procured for in-house provisioning end up having significantly more computational power than needed 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 the statistical multiplexing and virtualization it employs. For high workload intensity, cloud-based options are not cost-effective starting from year 1. These workload intensities are able to utilize well-provisioned servers making in-house procurement cost-effective.

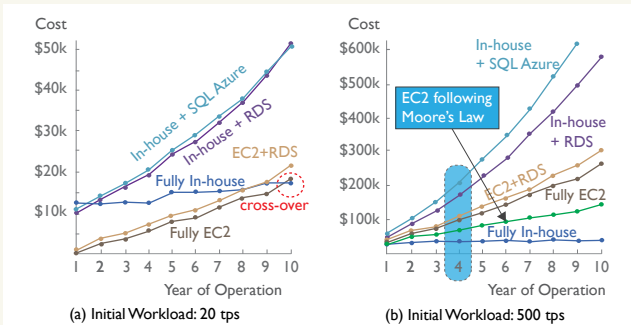


Figure 2: Cost evolution over a 10 year time horizon for TPC-W. We consider small (20 tps at $t=0$) and high workload intensity (500 tps), with 20% growth.

An interesting trend is the significantly slower NPV increase for in-house compared to cloud-based options. Since we assume that hardware capacity doubles biannually, unless the workload growth matches or exceeds this rate, the number of servers required in-house will actually shrink each year. This is why the cost of 'Fully In-house' case does not show a steep increase. However, the computing power, as well as price of a cloud instance, is engineered to be at a certain level even though cloud providers may upgrade their hardware regularly.

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. While in-house hosting enjoys improvement in perfor-

mance/\$ with time, trends over the last 5 years suggest that the performance/\$ offered by the cloud has remained unchanged. Even if we assume the performance/\$ offered by the cloud improves with 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. This hypothetical cost curve is plotted in Figure 2 (b) between 'Fully In-house' and 'Fully EC2'.

Network Usage, Storage Capacity, and Software Licenses

We illustrate in Figure 3 detailed breakdowns of NPV for four-year long hosting of TPC-W. 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 component-based partitioning may not be appealing for applications that exchange data with the external world. Data transfer (in & out) costs of hybrid options in Figure 3 are larger than non-hybrid hosting options because traffic per transaction between Jboss and MySQL (16KB/tr) is larger than between client and Apache (3KB/tr).

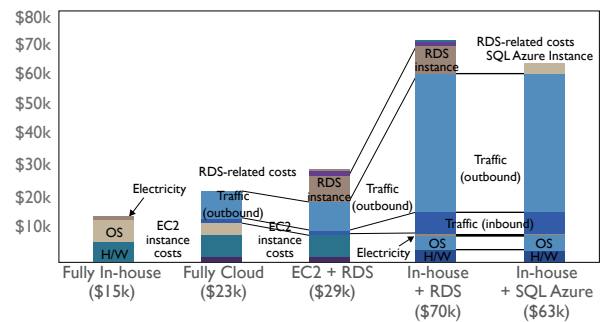


Figure 3: Cost break-down for deployment options at the 4th year. Initial workload is 100 tps (Transaction Per Second) and the annual growth rate is 20%.

Another determining factor to costs with cloud-based hosting can be storage capacity. Whereas TPC-W poses relatively small costs for storage capacity, TPC-E has significant data storage needs (about 4.5TB). From our analysis of the NPV cost evolution for TPC-E at two initial intensities - 300 tps and 900 tps with 20% annual growth rate, we find that in-house provisioning for TPC-E has to make significant investments in high-end RAID arrays 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% in year 1 and 28% in year 6.

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 and results in even better costs. SaaS options can be cost-effective for applications built using software with high licensing/maintenance fee.

Comparing the cost evolution for the two intensities of low (300 tps) and high (900 tps) workloads, under 300 tps, in-house option is less attractive than cloud-based options for the entire 10 year period without ever having a cross-over. However, at the higher intensity (900 tps), cloud-based options quickly 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. The key reasons for this difference are the higher storage costs for in-house TPC-E as well as the contribution of software licenses in non-SaaS options.

Workload Variance and Cloud Elasticity

Our cost analysis so far has not taken into account the variances in workload intensity. One potential cost benefit of cloud-based hosting is from the ability to dynamically match the resource capacity to the workloads at finer time-scale than in-house hosting. 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 have investigated NPV costs of variance-aware provisioning for three degrees of burstiness corresponding to time-of-day effects and flash crowds. Diurnal patterns (or

time-of-day effect) are periodic at the day granularity and predictable to some degree. It is known that the magnitude of daily workload fluctuations is around 40%-50% range for social networking applications, and about 70% for e-commerce Web sites. 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 represent the workload variance using peak-to-average ratio (PAR) which we define as $\max(x_t)/E(x_t)$ where x_t is the time series representing workload intensity (e.g., number of arrivals/sec). We choose PAR of 1.54 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).

For cloud-based hosting, we assume the following. Cloud provides a mechanism to monitor and detect the occurrence of sudden burst of workloads. Also, cloud provides a mechanism to scale out at run time to match the change of observed workloads. These assumptions are safe to make since those features are already supported by mainstream cloud providers. Therefore, regardless of PAR, the cost of cloud-based hosting is theoretically equal to the overall average workload intensity. In the real world, however, since most of the cloud charges the usage at the granularity of hours, the actual cost can be slightly higher than the theoretical costs.

According to our analysis using TPC-W and in-house hosting, provisioning for the diurnal fluctuation of 70% (PAR=1.54) does not impact the cost whereas flash crowd noticeably increases costs. Provisioning for PAR=11 shifts the cross-over point with "Fully EC2" from year 2.5 to year 6.5, implying that "Fully In-House" option becomes less economical. Provisioning for PAR=51 becomes highly uncompetitive compared to "Fully EC2" over the entire 10 year period. Note that the effect of diurnal fluctuation is miniscule because provisioned servers already have enough capacity to embrace the peak of diurnal fluctuation.

Cloud based hosting is not always more economical than the in-house hosting even under cloud's pay-as-you-go charging model and elasticity.

combined effect on overall costs. We have also explained issues regarding workload variance and workload-based partitioning. We find that (i) complete migration to today's cloud is appealing only for small/stagnant businesses/organizations, (ii) component-based partitioning options are expensive due to high costs of data transfer, and (iii) workload-based partitioning options can offer the best of in-house and cloud deployment for certain applications.

Our cost analysis of various hosting options using benchmark applications suggests a couple of points. First, the general cost trend of cloud-based hosting will be an increasing curve over time and the slope is sensitive to the workload size. On the other hand, the cost of in-house hosting tends to be more flat and stable. This implies that initial cost of in-house and cloud-based hosting options provide a rough indication of whether there will be a cross-over point or not. If the cost of cloud-based hosting is cheaper than in-house at year 1, there is a good chance that cloud-based hosting may become more expensive in the future. Therefore, if this happens, it is recommended to carry out detailed cost analysis by incorporating more accurate projections of workloads and taking into account various factor as we have shown in this study.

Second, the decision of application migration to the cloud should be a periodic task, rather than a one time activity at the beginning of

application deployment. This is because many conditions constantly change over the course of operating the applications. Cloud service providers introduce new pricing models and new type of services that might affect the way applications are being hosted. In order to make reasonably accurate assessments, users have to understand the implication of newly introduced services and pricing models. It is likely that new cloud services or features will be more economical since cloud providers actively try to enhance their services in order to stay ahead of competition. The application characteristics may also change. As time passes, the workload growth rate or the degree of workload variance may deviate from the earlier projections and they need to be adjusted.

Overall, we learn that answering the question of whether to migrate the application to the cloud or not is not a simple task. The difficulty lies in understanding the complexities of application characteristics as well as diversities of available cloud services. However, we do confirm from this study that cloud based hosting is not always more economical than the in-house hosting even under cloud's pay-as-you-go charging model and elasticity. Incorrect decision of hosting options may incur the costs that are several times higher than necessary. Future direction of our study is to develop a framework that can incorporate wider range of cost factors that are 'less quantifiable' and application types.

About the authors

Byung Chul Tak is a Ph.D candidate of computer science and engineering at the Pennsylvania State University. He joined Penn State as a Ph.D student in the Fall of 2006. Prior to joining Penn State, he worked as a researcher in the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. His research interest includes virtualization, operating systems and cloud computing. Current research focus is on developing accurate resource accounting frameworks of user applications in virtualized environment.

Bhuvan Urgaonkar is an associate professor in the department of computer science and engineering at the Pennsylvania State University. He is a recipient of the NSF CAREER Award, research awards from HP Labs and Cisco, and has co-authored best student papers at IEEE MASCOTS 2008 and ICAC 2005 conferences. His research involves applying ideas from distributed computing, resource management, scheduling, performance evaluation, and analytical modeling to the design and evaluation of data centers, networked systems, operating systems, virtualization techniques, and storage systems.

Anand Sivasubramaniam is a professor at CSE (Computer Science and Engineering) department, Penn State University. Dr. Sivasubramaniam's research interests are in designing, implementing, evaluating and tuning computer systems that span the spectrum of application domains from high performance clusters and shared memory multiprocessors, to embedded resource-constrained systems. His research has been funded by several NSF grants, including the CAREER award, a grant from the EPA, and gifts from industries including, IBM, Microsoft, and Unisys Corp. He is a recipient of an IBM Faculty Award. In 2010, he has been named a Distinguished Scientist by the Association for Computing Machinery (ACM).