2012 IEEE Seventh International Conference on Networking, Architecture, and Storage

C-IRR: An Adaptive Engine for Cloud Storage Provisioning Determined by Economic Models with Workload Burstiness Consideration

Jianzong Wang[1], Rui Hua[1], Yifeng Zhu[2], Changsheng Xie[1*], Peng Wang[1], Weijiao Gong[1]
[1] School of Computer Science and Technology, Huazhong University of Science and Technology, China
[1] Wuhan National Laboratory for Optoelectronics,Wuhan,China
[2] Department of Electrical and Computer Engineering, University of Maine, USA
[*] Corresponding Author: cs_xie@mail.hust.edu.cn

***Abstract*—Being the long dreamed vision of computing as a utility, cloud enables convenient and on-demand access to a large centralized pool of resources via network. The emerging of cloud storage offers a rather feasible solution to handle the sheer amount of information. It is maturing and becoming an alternative for on-premise storage. Thus, for IT enterprises with high demand of storage, a big concern is to determine whether it is more cost-effective to lease storage service over clouds. In this paper, we introduce a cloud storage provisioning engine called C-IRR to help users rationally evaluate the benefits of purchasing new disk drives and comparing it against leasing cloud storage offered by Infrastructure as a service (IaaS) providers. We also discuss issues regarding workload burstiness to achieve potential benefit for each applications in local data centers of SaaS providers. The C-IRR migrates the bursty workloads to the clouds and keeps stable ones locally. Such hybrid storage method achieves at least 20% costs saving in total for SaaS companies by experimental evaluation. In addition, C-IRR engine is of adaptivity about fluctuation of storage pricing and manpower cost increasing after the sensitivity studying.**

***Index Terms*—cloud storage, IaaS and SaaS, provisioning, internal rate of return, workload burstiness;**

I. Introduction

Cloud computing provides a great opportunity for data-intensive companies to get an easy access to high-performance computing and a mass of storage infrastructure by web connections. As a result, other than owning and maintaining disk drives and collectors in local data center, cloud storage serves as an alternative which could also be beneficial.

Under this tendency, it is predicted that thousands of medium-size enterprises with tens or hundreds of servers, which make up almost 50 percent of all data centers installed in the US, will be in a critical dilemma (i.e. "migrate to clouds or not" problem) in the foreseeable future. Another "to purchase or to lease from clouds" problem lies in the cost savings of actual applications for SaaS providers, where the costs of workloads are highly sensitive to their burstiness attributions. Because burstiness has substantial impact on workload placement and such placement scenario determines the final dollar cost.

In this paper, we import internal rate of return economic model into clouds (short for **C-IRR**) and design an adaptive engine with three core components: **Trace Engine**, **IRR Module**, and **Burstiness Filter**, which helps address "to migrate or not" problem for companies in terms of increased storage demands per year. Besides, we also evaluate the effectiveness of hybrid storage solution made by our workloads burstiness filter and experiment 20 workloads collecting from actual shared environment to certify the engine's superiority on financial saving. We have implemented the C-IRR engine as an agent between our local environment and real public cloud platform, and found it feasible and efficient in cost saving. The main contributions of our C-IRR engine are as follows:

- C-IRR evaluates the future storage demand by tracing previous data increment tendency (actual applications processed locally), which is completely customer made for growth-oriented enterprises. Solutions for companies can change with their different scales.
- C-IRR uses the widely-used Internal Rate of Return (IRR) in economics to address the problems of "lease or purchase", "where to lease" with regard to cloud storage provisioning. Obeying by the real market circumstances and public cloud storage providers' pricing, we have shown the appropriate suggestions to various types of enterprises and help their mangers to make cloud provisioning decisions.
- In regular services stage, we optimize our engine from workload utilization perspective to further complete workload provisioning into clouds for the purpose of cost saving as well.

The remainder of this paper is organized as follows. Section II shows the overview of our framework, working flow and trace engine. We then discuss the IRR along with NPV models and the influence of workloads' burstiness to the final costs. A case study using IRR module for different SaaS companies under Amazon Web Service is presented in Section IV, and then we describes the burstiness filter and evaluates its actual performance. Section V introduces related works, and we come to the conclusion in Section VI.

II. Overview of Modeling Approach

A. Work flow Description

Our C-IRR modeling involves the following three key steps, as shown in Figure 1. Firstly, the *trace engine* detects and records one year's increasing amount of workloads in local data center, and calculates the approximate storage growth. Then we bring in *Internal Rate of Return (IRR) model* to decide whether companies should purchase new disk drives or lease remote cloud storage service. Then we come to the challenge to estimate the unpredictable peak resource utilization (e.g. CPU, Disk, Network) of each workload is always a

978-0-7695-4722-0/12 $26.00 © 2012 IEEE
DOI 10.1109/NAS.2012.13

priority and risk in the local data center, because a bursty (i.e. high peak-average utilization ratio) workload actually causes a less dense workload placement possible on the server and hence much lower average server utilization, which renders in deployment of more resources and higher cost. Thus, in the following phase, we come up with a module called *burstiness filter* to identify those bursty workloads and then migrate them to the cloud storage service providers for the benefit of cost savings and risks avoiding.

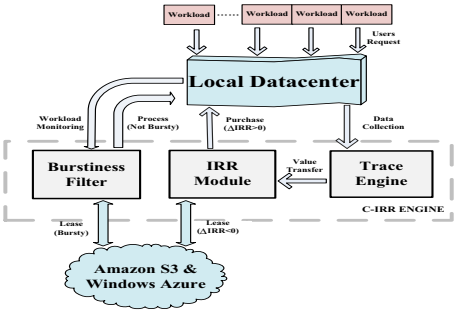

Fig. 1. Framework of C-IRR

In other words, as shown the working flow in Figure 1, we apply a *Trace Engine* to analyze previous records in the local data center to predict data extension tendency and estimate future demands in storage. Then predicted results will be input into *IRR Module*, where we use our IRR approach to draw graphic images of calculated ΔIRR values for a storage life expectancy from 0 to N years. Based on the results, the enterprises can make a quick purchase or lease decision in a more straight way. A positive ΔIRR value motivates companies to purchase new storage devices. Conversely, a negative ΔIRR stimulates companies to lease remote cloud storage instead of buying new ones. After disk drives bought as a new part of local clusters within the firms, the *Burstiness Filter*, a module in our C-IRR, continues to monitor users' request (i.e. workloads). The administrator sets a threshold as a guide to detect the bursty workloads and then migrates them to the clouds.

B. Trace Engine

The Trace Engine takes following three steps to complete the data extension tendency:

1) Based on the amount of new applications flowed into the local data center every year, the engine calculates and documents the storage increase records.
2) The predicted result of current disk demands is calculated by Eqn. 1 decided by the past storage growth and weight.
3) Trace Engine then transmits data to IRR module, where the "to purchase or not" decision is made.

The disk demands are influenced by the past years' statistics and their weights (A closer year is often endowed with a higher weight). The equation for storage demands prediction for year N is as follows:

$$T_N = \sum_{t \in N} T_t \times \mu_t \tag{1}$$

where μ_t means impact factor of increased storage in year t, which can be set on the basis of company's business expansion performance, market needs and long-term plan.

III. IRR Module and Workload Burstiness

A. The NPV and IRR model

In this paper, we use IRR model to help companies make decisions about "whether or not to migrate to cloud". Because the value of IRR takes into considerations of the time value of money, which can be considered as the interests earned from a risk-free investment. Compared with the NPV method, which only indicates the value or magnitude of the investment, the IRR method on the other hand reveals the efficiency, quality, or yield of an investment.

We introduce how to calculate the ΔIRR from economics perspective in this section. IRR provides the expected return rate of the project, it is the rate (represented by r in Eqn. 2) that makes NPV equal 0. To get an IRR of a project, we should first simplify the NPV equation form, and get the cash flow per year.

The simplified NPV equation is shown in Eqn. 2, and the parameters are reflected in Table I.

$$NPV = \sum_{t \in n} \frac{C_t}{(1+r)^t} \tag{2}$$

TABLE I
NOTATIONS OF NPV MODEL

Notations	Description
C_t	Cash flow in year t. if $C_t > 0$, it is in flow; Otherwise, out flow.
C_0	Initial disk drive investment. $C_0 < 0$
r	Discount rate
t	Time period (years)
n	Life cycle of this project (years)

So we can infer IRR of purchasing new disk drives (IRR_P) according to Eqn. 3, and similarly the IRR of leasing over the clouds (IRR_L) is obtained according to Eqn. 4.

$$NPV_P = \sum_{t \in n} \frac{C_t}{(1+IRR_P)^t} = 0 \tag{3}$$

$$NPV_L = \sum_{t \in n} \frac{C_t}{(1+IRR_L)^t} = 0 \tag{4}$$

Generally IRR [3] cannot be solved analytically if C_t is arbitrary. However, in the above equations, we have single outflow and multiple inflows, i.e. $C_0 \leq 0$ and $C_t \leq 0$ for $\forall t > 0$. We can have simplified numeric solution according the following two equations: Eqn. 5 and Eqn. 6 [4]. The initial two items, r_0 and r_1, are estimated by predicting the cash flow of the first two years. Then we can calculate ΔIRR by

using Eqn. 7. If ΔIRR is positive, companies are motivated to purchase new disk drivers rather than leasing from IaaS providers:

$$r_{n+1} = (1 + r_n)(\frac{1 + r_{n-1}}{1 + r_n})^K - 1 \quad (5)$$

$$K = \frac{\log(NPV_{n,in}/|C_0|)}{\log(NPV_{n,in}/\log NPV_{n-1,in})} \quad (6)$$

In this equation, $NPV_{n,in}$ refers to the NPV's inflow only (set C_0 = 0, and compute NPV).

$$\Delta IRR = IRR_P - IRR_L \quad (7)$$

Four parts make up a purchased asset's NPV:

1) The cost of purchasing storage, C
2) The revenue made by the purchased storage in year t, R_t^P
3) The purchased storage's expected operational *e*xpense in year t, E_t^p

$$E_t^p = 365 \times 24 \times \eta * (\rho_c + \rho_d * \Omega_d) + \Omega_a * W \quad (8)$$

where η means utility cost, ρ_c and ρ_d represent power demands for every disk controller and drive respectively, and Ω_a is number of operators, Ω_d is number of disks, while W is the administrator's salary per year.

4) The storage *s*alvage value at the end of its useful life N, S

$$S = \rho_d \times \Gamma \times P * e^{-0.438*t} \quad (9)$$

where Γ means depreciation factor and P expresses disk driver price, and the $e^{-0.438*t}$ derives from [1] in which disk price trends are predicted using regression analysis.

The standard capital budgeting for bringing new disk drives into local data center is as follows:

$$NPV_p = \sum_{t \in N} \frac{R_t^p - E_t^p}{(1 + r)^t} - \frac{C}{(1 + r)^0} + \frac{S}{(1 + r)^N} \quad (10)$$

Similarly, the NPV of a leased asset includes the following four components:

1) The lease payment in year t, L_t

$$L_t = S_L \times \Psi \quad (11)$$

where Ψ means the IaaS providers' charge for renting their servers per year, and S_L is total storage leased over the clouds.

2) The revenue made by leasing storage in year t, R_t^L
3) The leased storage's expected operational expense in year t, E_t^L

$$E_t^L = \Omega_c \times P \quad (12)$$

where Ω_c indicates the number of operators over clouds.

4) Based on economic Law of One Price [2] , firm's cost of capital and interest rate for renting clouds can be substituted by interest rate r.

So the equation for calculating the NPV of an asset leased over the cloud is as follows [5]:

$$NPV_p = \sum_{t \in N} \frac{R_t^L - E_t^L - L_t}{(1 + r)^t} \quad (13)$$

B. Implementation

Our C-IRR engine is implemented primarily in Matlab, with small portion of C/C++ and Python. To access and manage the elements of the cloud storage, we integrate libs3 and libcurl into C-IRR for interaction with Amazon S3 to grab the real-time pricing information every five hours. The monitoring and burstiness filter functions are implemented in the back-stage management for the workloads tracking. As described in more detail later in the evaluation section, C-IRR lowers cost and improves performance by adopting an adaptive engine for the determination how many storage resources stored on the cloud provider.

IV. Components Evaluation

A. Case Study of IRR method

In this study of cases, we adopt Amazon S3 as IaaS providers and aim to examine and find the scenarios lying in different companies when they face the "to lease or not" problem. Suppose the enterprise purchases new disk drives for its local data centers, we should take following elements into account:

- Assume a new disk controller ($1800) must be bought, which consumes 0.5 KW of power.
- A hard drive with 1TB capacity costs approximately $100, and each consumes 0.01 kilowatt/hour of power.
- The electricity price η is $0.045 per kilowatt hour.
- The depreciation factor for the salvage disks Γ is 0.1.
- Assume the administrators' payment (W) is $1200 per year for maintenance of 1TB.
- Suppose the income that 1TB brings is $4000 per year

Otherwise, if the company chooses to store data over clouds, the charge policy is $0.093 per GB under S3's RRS pricing strategy. Besides, we assume the administrators' salary in this case is $600 per year for preserving 1TB. To better evaluate the IRR performance for both situations, as a part of investment, we assume all salaries have been paid off before the first year and the income is as much as that of purchased scenario.

Based on above mentioned conditions and the equations showed in Section IV, we can get the cash flow. Figure 4, 5, 6 show the results when using S3's RRS, through which we are able to calculate the IRR value shown in Figure 3.

These graphs of cash flow show:

- The costs and profits both increase with the scale. The more storage demands those companies require, the more costs and profits will be in the next future years
- The cost of purchasing new disk drives is relatively higher than that of leasing from IaaS providers.

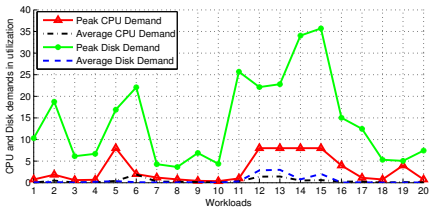

Fig. 2. CPU and Disk Demands in Utilization for 20 Workloads in Shared Environment

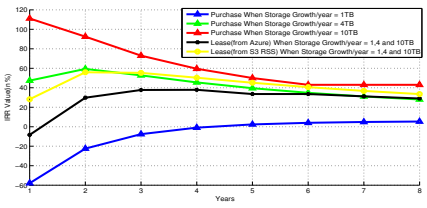

Fig. 3. The value of IRR under different purchase and lease scenarios when the annual storage growth rate is 1, 4 and 10TB

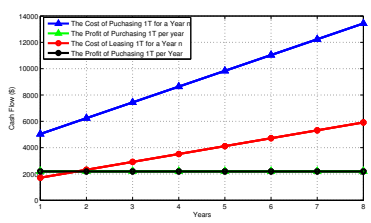

Fig. 4. Cash flow when annual storage growth is 1T (small-sized) in S3's RRS Environment

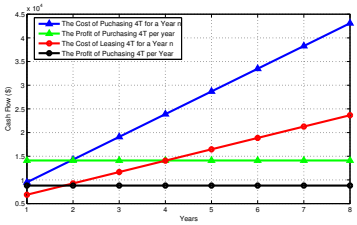

Fig. 5. Cash flow When annual storage growth is 4T (medium-sized) in S3's RRS Environment

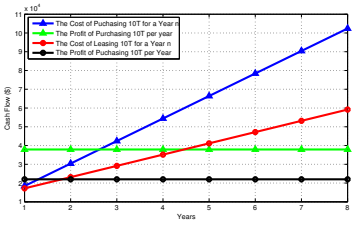

Fig. 6. Cash flow When annual storage growth is 10T (large-sized) in S3's RRS Environment

- The difference of puchasing cost/profit and leasing rises over time.
- The profit of purchasing new disk drives is pretty close to that of leasing for small-sized company, while the profit of purchasing is much higher when the demands go beyond 1 TB.

Figure 3 shows the IRR results under different circumstances, where we can infer that the value of IRR is the same for distinct annual storage growth when leasing from the cloud. The hypothetical numbers of increased storage requirement for small and medium-sized enterprises are 1TB and 4TB respectively, and the number for Large-sized enterprises is 10TB.

For small-sized enterprises aforementioned, the blue line in Figure 7 shows the approximate ΔIRR trend in recent 8 years. We find that the IRR of leasing over the clouds exceeds that of purchasing new disk drives and human capital for operation and monitoring. For the large enterprises with a data center of thousands servers, their trend of ΔIRR is indicated by red line. As described in the curves, the investment of purchasing new devices becomes more profitable, the tendency is quite clear and starts to even out after 6 years, and such investment benefit those far-sighted enterprises with servers of long expectancy. As with the medium-sized companies described by green curve, their choices are on the borderline. The decision highly depend on the IaaS's providers' charge strategies.

B. Test of Burstiness Filter

Our Burstiness Filter aims to detect those applications with high burstiness and transport them to the IaaS providers. In this way, the resources in local data centers can be utilized more efficiently and more profits can be made by SaaS providers.

In the burstiness filter, the bursty workloads migration process can be divided into three steps:

- Firstly, Once a new application enters into the local data centers, our filter begins to monitor it for three days and record its utilization every five minutes.
- After three days' trace, we get 864 pieces of data for each workload, where the system can automatically calculates the average and peak demands, and thus gather information about peak-average ratios.
- In the mean time, the administrator sets a threshold to differentiate bursty workloads and normal workloads. For example, if the threshold is set at 10, it means when a workload's peak-average ratio exceeds 10, it will be migrated to the clouds and managed by IaaS providers.

To more clearly test our filter's performance, we consider the threshold equals 50 and do not expect additional networking cost has much impact on total cost. Figure 8 shows their overall cost, Where we can infer that our method saves cost for 70% of all 20 traced workloads. The combined total for original cost is $800, while burstiness saves 25% cost and makes the actual expense $600 in the aggregate, then we draw following conclusions:

- Our burstiness filter helps to save at least 20% total expense if threshold is set beyond 20. It perfectly testifies the effectiveness of our proposed hybrid storage for cost saving.
- The total costs of different threshold in the burstiness filter vary from each other. Taking the fee of transferring in and transferring out over clouds and operating costs into consideration, the leasing cost of workload with lower peak-to-average ratio may sometimes exceed that of workload locally hosted. That is a vital reason why the cost of threshold in 10 is much higher than that of threshold in 50.
- When the threshold is under 50, the burstiness filter is

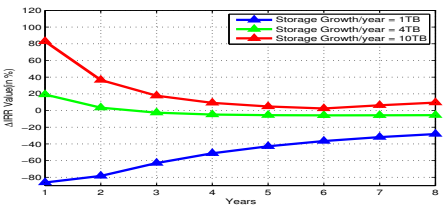

Fig. 7. ΔIRR when annual storage growth is 1, 4 and 10TB using Amazon S3 RRS pricing

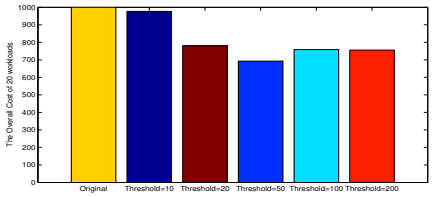

Fig. 8. The overall cost estimates of 20 workloads before and after using Burstiness Filter when the threshold is 10, 20, 50, 100, and 200 respectively

much less efficient due to the costs of I/O scheduling process and operating charge of burstiness filter compared with limited cost saving over clouds.

- When the threshold is beyond 50, the efficiency of filter also decays, because some bursty applications are not counted in and migrated to the IaaS ends.

V. Related Works

A few researches have been done to help evaluate the cost saving over clouds through scientific experiments. Study [6] discusses the viability of Amazon's Simple Storage Service (S3) to save cost for IT enterprises. However, we observe that there is a lack of generalization for those applications not used in the study. For Net Present Value (NPV) model, Ref. [1] has recently employed this concept in their work, focusing mainly on separately exploring the feasibility of renting computing and storage from the cloud. While we go beyond this work to consider four problems: (I) We employ IRR other than NPV as a guide to determine whether to cloud or not, because IRR has a strength in showing the efficiency of the project's investment and its simplicity; (II) As opposed to comparing rental vs. in-house costs only for a given hardware base, we compare the costs for hosting specific workloads; (III) We incorporate additional costs such as electricity; and finally (IV) we study the impact of workload evolution/variance and cloud models (IaaS vs. SaaS). Ref. [7] shows nearly 20% cost savings by bringing forward "right-virtualizing" method, but this study aims to solve "to virtualize or not" problem and only considers the licensing fees of virtualization technologies for IaaS providers. Hence, in this project, we attempt to build an integrated engine with generalization and suitability to Software as a service (SaaS) providers. Ref. [8] compares and contrasts costs of cloud and grid models using server measurements and financial expenses collected from real VC projects, but it fails to consider from economic angle. Ref. [9] considers the question: should the application be migrated to the cloud by exploring various workloads case by case and then attempt to draw generalities, and their results show workload intensity, growth rate, and storage capacity produce complex combined effect on the costs. Their discussions are different from ours, as our work includes IRR, which is the perfect use of time value of money theory. Moreover, our engine includes burstiness filter for the benefit of application' cost saving.

VI. Conclusion

We raise the "to lease or not" problem in storage demands growth and actual applications for SaaS providers. To better solve this challenge, we introduce our C-IRR which includes three major components: Trace Engine, IRR module and Burstiness Filter. The C-IRR engine not only determines whether to cloud in the purchase phase, but also transport those workloads that possibly increase the local costs to IaaS providers. Through evaluation, we are able to find it is better to lease from the cloud for most small-sized companies, while in contrast purchasing new disk drives could be beneficial for medium and large-sized companies. Briefly, C-IRR not only serves to solve purchase or lease problem for IT firms, but also automatically monitors and inspects workload condition in the local data center and categorizes workloads according to burstiness attribute. The goal is to keep workloads with stable and predictable utilizations in the local data center.

ACKNOWLEDGEMENT

This Project supported by the National Basic Research Program (973) of China (No. 2011CB302303), the National Natural Science Foundation of China (No. 60933002) and National Science Foundation (CNS #1117032, EAR 1027809, IIS #091663, CCF #0937988, CCF #0737583, CCF #0621493).

References

[1] Edward Walker, Walter Brisken, and Jonathan Romney To Lease or not to Lease from Storage Clouds. *Computer*, 43:44-50, 2010.

[2] O.A. Lamont and R.H. Thaler. Anomalies: The Law of One Price in Financial Markets. *J.Economic Perspective*, vol. 17, no.4,2003.

[3] Hazen, G. B., A new perspective on multiple internal rates of return. The Engineering Economist, 48(2), 2003

[4] Hartman, J. C., and Schafrick, I. C., The relevant internal rate of return. The Engineering Economist, 49(2), 2004

[5] Bruce J. Feibel, Investment Performance Measurement. New York: Wiley, 2003

[6] M. Palankar et al. Amazon S3 for Science Grids: A Viable Solution?. *Proc. Int'l Workshop Data-Aware Distributed Computing*, ACM Press, 2008, pp. 55-64.

[7] D. Gmach, J. Rolia, and L.Cherkasova, Resource and Virtualization Costs up in the Cloud: Models and Design Choices. *Proc. of the International Conference on Dependable Systems and Networks, (DSN20011)*,Hong Kong, China, June 27-30, 2011.

[8] D. Kondo, B. Javadi, and P. Malecot, Cost-benefit Analysis of Cloud Computing versus Desktop Grid. *In Proceeding of the 2009 IEEE International Symposium on Parallel and Distributed Processing*,Pages 1-12, USA, 2009

[9] Byung Chul Tak, Bhuvan Urgaonkar and Anand Sivasubramaniam To Move or Not to Move: The Economics of Cloud Computing. *Proceedings of the Third USENIX Workshop on Hot Topics in Cloud Computing (HOTCLOUD 2011)*,Portland, OR, June 2011