Intelligent Cloud Capacity Management

Yexi Jiang*, Chang-Shing Perng†, Tao Li*, Rong Chang†
*School of Computing and Information Sciences
Florida International University, Miami, FL, 33199
Email: {yjian004, taoli}@cs.fiu.edu
†IBM T.J Watson Research Center, Hawthorne, NY, 10532
Email: {perng, chang}@us.ibm.com

***Abstract*—Cloud computing as a service promises many business benefits. The cost to pay is that it also faces many technique challenges. One of the challenges is to effectively manage cloud capacity in response to the increased demand changes in clouds, as computing customers now can provision and de-provision virtual machines more frequently. This paper studies cloud capacity prediction as a response to the challenge. We propose an integrated solution for intelligent cloud capacity estimation. In this solution, a novel measure is introduced to quantify and guide the prediction process. Then an ensemble method is utilized to predict the future provisioning/de-provisioning demands respectively. The cloud capacity is estimated using the active virtual machines and the future provisioning/de-provisioning demands altogether. Our proposed solution is simple and with low computational cost. The experiments on the IBM Smart Cloud Enterprise trace data shows our solution is effective.**

***Index Terms*—cloud service; capacity management; service quality maintenance.**

I. Introduction

As cloud providers promise pay-as-you-go style for cloud service, the resource demand becomes more volatile than traditional IT environments. Such kind of service style brings new challenges to IT infrastructure capacity management, including fast virtual machine provisioning [3, 4], effective resource utilization [2] and effective patch management [6].

Capacity management in cloud is one of the challenges that has a critical impact on the service quality and the profitability of the cloud. On one hand, if the actual demand is higher than the existing capacity, the cloud has to turn down new customers and lose potential revenue. If the shortage is severe, even provisioning requests from existing customers have to be rejected, which defeats the promise that application in cloud can scaling-up whenever workload increases. On the other hand, overestimating demands may result in resource idling and unnecessary utility costs. The unused hardware not only causes under-utilized capital, it also causes more early purchase costs as the price of the same computing equipment is always going down. Another associated cost with idled equipments is the cost of network, labor, facility (floor space, cooling systems, power systems, etc.), and utility (electricity and water). Among these factors, power consumption is a major cost: the US Environment Protection Agency (EPA) estimates that the energy usage at cloud data centers is successively doubling every five years. In the year of 2011, the annual electricity cost at these data centers would be approximately $7.4 billion [5].

978-1-4673-0269-2/12/$31.00 © 2012 IEEE

In this paper, we focus on the problem of capacity estimation and management of cloud environment. We handle capacity estimation using a two-stage approach: an estimating stage and a controlling stage. Figure 1 illustrates the framework of the capacity estimation system. The core of the system is the *Capacity Predictor*, which predicts the future cloud capacity based on the information of historical provisioning data, and notifies the *Power Management Module* to prepare the required resources (include active servers, active coolers, other auxiliaries, and scheduled labor) in advance. The *Capacity Controller* would keep tuning the capacity according to the actual workload of the cloud environment.

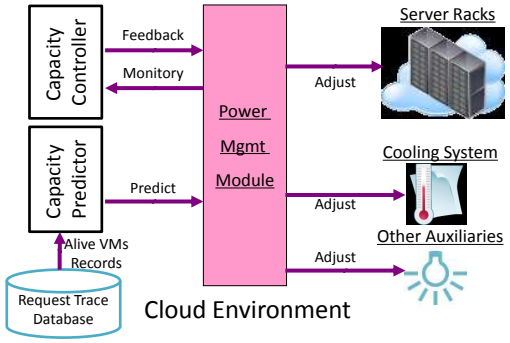

Fig. 1. System Framework

The contribution of this paper can be summarized as follows: (1) We solve the capacity estimation problem by decomposing it into the provisioning demand and de-provisioning demand. By leveraging online ensemble learning to predict the provisioning demand, we propose a method to utilize life-span distribution for estimating the de-provision demand. (2) We introduce a novel measure based on the uniqueness of cloud service to quantify the prediction result. By incorporating this measure, the prediction preference can be dynamically tuned according to actual running status of the cloud.

II. Methodology

The problem of cloud capacity management is to ensure that the amount of prepared resources in cloud is consistent with the real demands in the near future. Without loss of generality, we model the cloud capacity as the number of virtual machines

(VMs). Let C_t be the future cloud capacity at time t, it can be modeled as the sum of the number of active VMs in the cloud $active_{t-1}$ at time $t-1$, the number of upcoming provisioned VMs $prov_t$, and the number of upcoming de-provisioned VMs $deprov_t$, i.e,

$$C_t = active_{t-1} + prov_t - deprov_t. \tag{1}$$

In Equation (1), $active_{t-1}$ can be easily obtained from the system. $prov_t$ and $deprov_t$ must be estimated or predicted.

Given the number of active VMs and the historical requests up to time $t-1$, the problem of *estimating the future cloud capacity* at time t is to predict $prov_t$ and $deprov_t$ that minimize the error E, i.e,

$$E = f(prov_t, \hat{prov}_t) + f(deprov_t, \hat{deprov}_t), \tag{2}$$

where $f(\cdot,\cdot)$ denotes the cost function and $\hat{prove}_t$/$\hat{deprov}_t$ denotes the error made by provisioning/de-provisioning estimation respectively. The best capacity predictor should be the one that minimizes the sum of errors over multiple timestamps, i.e. $argmin \sum_t E$.

A. Performance Evaluation Criteria

The consequences of over-estimation and under-estimation are different in cloud capacity prediction. For provisioning estimation, an over-estimation has no negative effects on the customer and only causes idled cloud resources while an under-estimation degrades the service quality. For de-provisioning, an over-estimation degrades the service quality while an under-estimation causes idled resources and revenue loss. Traditional regression measures such as mean absolute error (MAE), mean square error (MSE), mean absolute percentage error (MAPE) only focus on the absolute accuracy of the estimated results but ignore the differences between over-estimation and under-estimation. Considering the uniqueness of cloud demand, we propose a novel error measure called Demand Estimation Error (DEE). DEE is an asymmetric measure that models the over-estimation and under-estimation of the demand as different kinds of costs: the cost of idled resources and the penalty of SLA (Service Level Agreement).

Suppose the real demand at time t is $v(t)$ and the estimated demand at time t is $\hat{v}(t)$. In DEE, we use cost function $P(v(t), \hat{v}(t))$ to quantify the penalty of SLA and $R(v(t), \hat{v}(t))$ to quantify the cost of idled resources respectively.

1) Cost of SLA penalty: When the service quality violates the pre-defined service level agreement (SLA), there is a penalty for the provider. The under-estimation of capacity is one of the reasons that would cause the decrease of service quality. Without loss of generality, we use the delay of response time to quantify the SLA penalty caused by under-estimation. The form of $P(v(t), \hat{v}(t))$ can be modeled as Equation (3),

$$\begin{aligned} P(v(t), \hat{v}(t)) = & \, min(v(t), \hat{v}(t)) T_{available} \\ & + max(0, v(t) - \hat{v}(t)) T_{wait}, \end{aligned} \tag{3}$$

where $T_{available}$/T_{wait} denotes the preparation time for VM when there is/isn't enough resources. Based on the definition, the more serious the under-estimation, the larger the cost. There are more sophisticated forms of the P function. For example, a common SLA typically specifies a penalty threshold for provisioning time. The time cost of a non-violated requests would have zero values for P function.

2) Cost of idled resources: This is the non-billable cost of resources including the cost of electricity and the labor etc. For simplicity, we use R_{vm} to denote the average cost of all resources for a single VM by assuming the idled resources cost for each VM in a unit time is identical. The R function is defined as:

$$R(v(t), \hat{v}(t)) = max(0, \hat{v}(t) - v(t)) R_{vm}, \tag{4}$$

Combining Equation (3) and (4), the total estimated cost is as:

$$C = \begin{cases} \alpha v(t) T_{available} + (1-\alpha)(\hat{v}(t) - v(t)) R_{vm} & \text{if } v(t) < \hat{v}(t), \\ \alpha(\hat{v}(t) T_{available} + (v(t) - \hat{v}(t)) T_{wait}) & \text{if } v(t) \geq \hat{v}(t). \end{cases} \tag{5}$$

For different clouds, the trade-off between SLA penalty and idled resource cost may be different, we use the parameter α to tune the preference of P function and R function. As mentioned before, capacity estimation aims to achieve the minimum total cost quantified by Equation (5).

B. Capacity Prediction

Incorporating the estimated provisioning demands and de-provisioning demands, the capacity change in one time unit can be estimated as $\Delta_t = prov_t - deprov_t$.

1) Demand Prediction : Similar to the retail supply scenario, individual customers may come at any time to purchase arbitrary amount of items, there is no trivial way to infer the demands of individuals. In the absence of deep understanding of customer behavior, one possible way of estimating the provisioning demand is to leverage the time series techniques to model and infer the global behavior of all the customers.

We propose to use the ensemble learning method to combine the power of multiple prediction techniques for predicting. There are two reasons why we utilize the ensemble method: (1) The robustness of ensemble method mitigates the risk of large deviation for prediction results. (2) We can dynamically tune the preference of predictor to be more optimistic or more pessimistic according to the actual workload of the cloud.

The ensemble method proposed in our work is motivated from the online majority voting algorithm for classification [1]. We propose a weighted linear combination strategy for provisioning demand prediction. Suppose the predicted value for predictor $p \in \mathcal{P}$ at time t is $\hat{v}_p(t)$ and its corresponding weight at time t is $w_p^{(t)}$, the ensembled predicted value at time t is

$$\hat{v}^{(t)} = \sum_p w_p^{(t)} \hat{v}_p(t), \text{subject to } \sum_p w_p^{(t)} = 1. \tag{6}$$

To quantify the error, at each iteration, we calculate the relative error $e_p^{(t-1)}$ caused by predictor p at time $t-1$

according to

$$e_p^{(t)} = \frac{c_p^{(t-1)}}{\sum_i c_i^{(t-1)}} w_p^{(t-1)}, \tag{7}$$

where $c_p^{(t-1)}$ is the prediction cost/error and can be quantified by DEE. Afterwards, we normalize the error and update the weight of each predictor according to Equation (8)

$$w_p^{(t)} = \frac{e^{(t)}}{\sum_p e_p^{(t)}}. \tag{8}$$

In this paper, we employ five different time series predictors, their names and categories are listed in Table I.

Method Name	Category
Moving Average	Naive
Auto Regression	Linear Regression
Artificial Neural Network	Non-linear Regression
Support Vector Machine	Linear Learner with Non-linear Kernel
Gene Expression Programming	Heuristic Algorithm

TABLE I
TIME SERIES PREDICTION TECHNIQUES USED FOR ENSEMBLE

2) Estimation of De-provisioning Demand: In order to estimate the de-provisioning demand, we make use of the temporal characteristics of the VM. Through exploration, we find that knowing the current life time of individual VM is helpful for estimating the amount of de-provisioning. In other words, we can infer when a certain VM would be de-provisioned through the life span distribution of the images. Under the stationarity assumption, the life span distribution of the VMs does not depend on the time the VMs are provisioned. Let $life(VM)$ denotes the current life time of a VM, n_i denote the frequency of VMs with life span t_i, we estimate the CDF of the life span as follows:

$$\begin{aligned}\hat{F}(x) &= P(life(VM) \leq x) \\ &= \begin{cases} n_1 / \sum_i n_i & t_1 \leq t < t_2, \\ (n_1 + n_2) / \sum_i n_{t_1} & t_2 \leq t < t_3, \\ ..., & ... \\ (\sum_{i=1}^{n-1} n_i) / \sum_i (n_i) & t_n \leq t. \end{cases}\end{aligned}$$

The output of the estimated CDF denotes the probability of a VM that would be de-provisioned at time t_i. Utilizing $\hat{F}(x)$, the de-provisioning demand can be estimated by

$$\sum_{i \in VM_{active}} \hat{F}(life(i) \leq t_{now} - t_{start(i)}),$$

where t_{now} denotes the current time and $t_{start(i)}$ denotes the provisioning time of VM i.

III. EXPERIMENTAL EVALUATION

Our experimental evaluation is based on the real historical VM trace log obtained from IBM's Smart Cloud Enterpriese (SCE) platform. The goal of the experiments is trying to answer the following questions: (1) Whether the estimation of provisioning and de-provisioning is accurate? (2) Whether the estimation of de-provisioning demand based on the life span distribution outperforms the time series prediction method?

A. The Choice of Aggregation Granularity

Data preprocessing is the first step of capacity estimation. The raw request records cannot be directly used for demands estimation for two reasons: (1) Directly building estimation/predicition models based on the low-level representation of the time series is very difficult. (2) The raw data contains irrelevant features that are useless.

The time series used in our work is generated by aggregating the request records via the VM types and request timestamps. The aggregation can be conducted at different time granularities. Figure 2 shows the time series of the same VM type in weeks, days and hours.

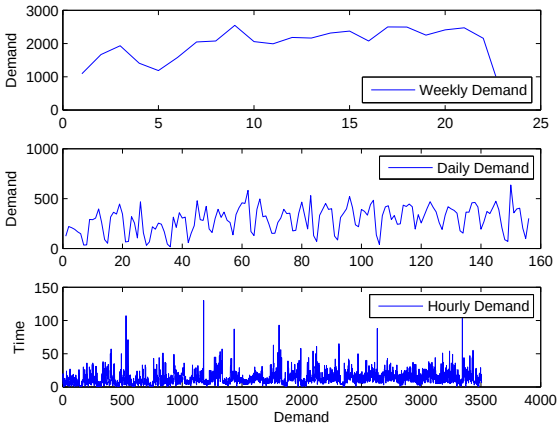

Fig. 2. Time series with different granularity. The top one is aggregated weekly, the middle on is aggregated daily, the bottom one is aggregated hourly.

It is trivial to know that the coarser the granularity, the larger the demand in each time slot. The weekly aggregated time series requires the system to prepare too much VMs and most of the prepared VMs would be idle. Based on our empirical study, the life span of most of the VMs are shorter than one week, the weekly aggregated values cannot reflect the real capacity required. Moreover, compared with a finer granularity, even a smaller portion of deviation would result in large consequence for weekly aggregated data.

On the contrary, aggregation at the finest granularity, i.e. hourly, is meaningless since the life span of the VMs cannot be so short. Furthermore, such a fine granularity would make the value on each timestamp lack of statistical significance. Therefore in our work, the time series is aggregated daily.

B. Evaluation of Provisioning Estimation

For provisioning demand prediction, we compare the performance of the individual predictors with our proposed ensemble method. We use DEE as the measures. For the parameters of DEE, we set $\alpha = 0.5$, $T_{wait} = 1200$ (the time for on-demand preparation, include server boot up, server registration etc.),

$T_{available} = 10$ (resource is available instantly), $R_{vm} = 500$ (the cost of one idled hardware unit).

1) Prediction Precision Comparison: We partition the time series horizontally into two parts and use the data before May for training. We evaluate the precision of these predictors on data of May, June and July respectively. In order to eliminate the randomness of some predictors (Random, Artificial Neural Network and Gene Expression Programming), the results are computed by averaging 10 runs of each predictor. For each individual predictor, the grid search strategy is used to find the best parameters. Table II shows the evaluation results of all the predictors on the test time series, it clearly illustrates that the best predictor is different for different test data sets. On average, the ensemble method achieves the best performance.

Predictor	May	June	July	Average
Random	2281555	3507600	3080320	2956491
Moving Average	1295550	1293620	1293620	1294263
Auto Regression	504912	760110	1047275	770765
Artificial Neural Network	980780	1095102	1577127	1217669
Gene Expression Programming	866117	640405	1037705	848075
Support Vector Machine	3746005	2199010	1147240	2364085
Ensemble	538302	626585	1072840	**745909**

TABLE II
THE COST OF DIFFERENT PREDICTORS UNDER DIFFERENT MEASURES

C. Evaluation of De-provisioning Estimation

In this section, we experiment four different variations of the life span distribution methods as well as the time series prediction method to estimate the de-provisioning demands. The detail of the four different variations are listed as follows:

1) **Using 90 days of global life span distribution.** We estimate the distribution as well as the CDF with the latest 90 days of historical data. For each day, we estimate the expected number of VMs that would be de-provisioned based on their living time. We name this method as *Dist 90*.
2) **Using 60 days of global life span distribution.** This variation is the same as the first one, but with only 60 days of historical data. We name this method as *Dist 60*.
3) **Using 90 days of individual life span distribution.** For this variation, we estimate the distribution and CDFs of individual image types. This method is a finer granularity version of the first method. We name this method as *Individual*.
4) **Using 90 days of hybrid life span distribution.** We combine the estimation of global distribution and individual distribution together and weight them as $\beta\hat{F}_i + (1-\beta)\hat{F}_g$, where β equals to the fraction between the frequency of specified image type and the most popular image type. We name this method *Hybrid*.

For all these variations, the experiments are conducted on the last 60 days of de-provisioning data.

Figure 3 shows the errors of these methods for prediction. Based on the experiment results, the *Dist 90* has the best performance among all the methods. It seems strange that the finer version methods *Individual* has worse performance comparing with the global version. This phenomenon is caused by the sparse occurrence of individual image types in real data. In ideal case, the empirical distribution can be very close to the real distribution if there are plenty of requests for each image types. However, the requests distribution for different image types is unbalanced and some popular image types dominate the requests, so the life span distribution of the minority image types should be deviated from the real distribution due to lack of statistical significant. The erroneous estimated distribution further causes the bad prediction performance. Similarly, the precision of *Hybrid* is also heavily affected by the imprecision of individual distributions since it incorporates the *Individual*.

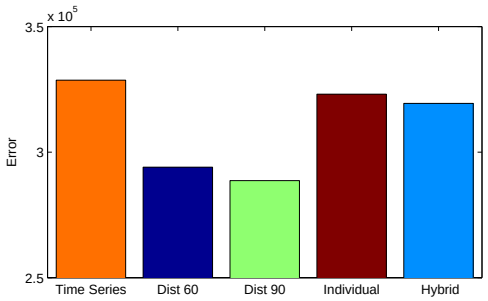

Fig. 3. Errors of different methods

IV. CONCLUSION

In this paper, we proposed an effective framework for cloud capacity prediction based on the knowledge learnt from the requests history. We leveraged the time series ensemble method to predict the provisioning demands and utilized the life span distribution of VM to estimate the de-provisioning requests. The experimental results demonstrated that our method is precise on capacity estimation and can effectively reduce the maintenance cost of the cloud environment.

ACKNOWLEDGEMENT

We thank the reviewers for their constructive and helpful suggestions. The work is partially supported by National Science Foundation (NSF) under grant IIS-0546280.

REFERENCES

[1] Avrim Blum. On-line algorithms in machine learning. In *Proc. of the workshop on Online Algorithms*, 1996.
[2] Zhenhuan Gong and Xiaohui Gu. Pac: Pattern-driven application consolidation for efficient cloud computing. In *MASCOTS*, 2010.
[3] Yexi Jiang, Chang-Shing Perng, Tao Li, and Rong Chang. Asap: A self-adaptive prediction system for instant cloud resource demand provisioning. In *ICDM*, 2011.
[4] Shicong Meng, Ling Liu, and Vijayaraghavan Soundararajan. Tide: Achiving self-scaling in virtualized datacenter managment middleware. In *MiddleWare*, 2010.
[5] Debprakash Patnaiky, Manish Marwah, Ratnesh Sharma, and Naren Ramakrishnany. Sustainable operation and management of data center chillers using temporal data mining. In *KDD*, 2009.
[6] Bo Yang, Sai Zeng, Naga Ayachitula, and Rajeev Puri. Sla-driven applicability analysis for patch management. In *IM*, 2011.