Article can be accessed online at http://www.publishingindia.com

A Dynamic Predictive Approach to Identify an Optimal Cloud Availability Zone with Maximum Satisfaction Level

Santhosh R.*, Ramya A.**

Abstract

Cloud infrastructure service provider allows the users to place their business application into availability zone across various regions world-wide by enabling enterprises like amazon with sufficient capability. Amazon EC2 is an instance to be placed in a zone to provision their resources to run their own applications. Multiple availability zones are composed and located in single region each one is different from its character specification because of its hardware and software built version. These zones allows achieving higher availability with lower failure rates but may results in different quality of service against user requirements. This situation may not advertised by cloud infrastructure service provider. In this paper, we proposed the prediction model build by repeated k-fold cross validation, which overcomes the prediction error Type-I exist in k-fold cross validation. This proposed technique helps us to predict optimum availability zone with best user satisfaction level for amazons EC2 placement in heterogeneous environment.

Keywords: Availability Zone, Machine Learning Technique, Repeated k-fold Cross Validation, Prediction Model

1. Introduction

A cloud infrastructure service provider offers organizations to provision resources [Bogumil, 2015]. Provision of resources will perform by availability zones located across world-wide. Availability zone is a data centre which is located in various regions. A region composed of multiple availability zones, each one is isolated from another availability zone. The built version of availability zone may different in hardware and software configuration, this will results in different QoS. Here moreover zone selection expected to be satisfying the amazon EC2 requirement satisfaction. Amazon EC2 (Amazon Elastic Compute Cloud) is introduced in 2006. It supports wide range of instance deployment [Farley, 2012] [Gitte, 2012]. Amazon EC2 provisions their resources by help of availability zones. It allows to increase and decrease availability within minutes. The selection of availability zone should satisfy the required characteristics, so that resource can get placed into the location. The runtime performance, instance type, machine type, memory capacity are the attributes to be satisfied when new instance deployment. Deployment of same type of instance in different availability zone results in different solution even provided by same provider. This will lead to wrong selection of zone. Amazon EC2 was located in US and EU. The same instance located in different availability zone differs in performance. The benchmark a result shows that the amazon EC2 instances results in performance variation and selecting a fast instance type within the same instance type reduces the users cost [Juan, 2010]. The both large and small instances suffer by performance variation [Kohavi, 1995]. This inconsistency is unwelcome challenge in availability zone selection. In this paper, a prediction tool is in introduced to automatic

* Assistant Professor, Department of Computer Science and Engineering, Faculty of Engineering, University, Coimbatore, Tamil Nadu, India.. E-mail: santhoshrd@gmail.com

** PG Scholar, Department of Computer Science and Engineering, Faculty of Engineering, Karpagam University, Coimbatore, Tamil Nadu, India. E-mail: ramya12.eng@gmail.com

selection of availability zone with satisfaction level. This tool generates the user satisfaction level by receiving the QoS attributes which are needed to predict. By using these attributes the utility value will generated, which will helpful in training the model to predict the zone.

1.1 Related Work

The prediction model is built by using machine learning technique which uses the requirements to train the data and make prediction on same data. The cross validation technique is used in machine learning for training the model [Li, 2010] [Schad, 2010]. The traditional k-fold cross validation technique is used often for training the model for selection process [Mark, 2006] [Merve Unuvar, 2015] [Payam Refaeilzadeh, 2008]. The only drawback in this technique is there exist Type-I error, it may results in true null hypothesis which leads to incorrect selection of zone of given requirements. This may happen because of minimum number of trails taken in training phase. In this paper we introduced the repeated k-fold cross validation [Valerio Persico, 2015], it increases the number of trails for training process and performing around 50 estimation [Schad, 2010]. This overcomes the drawback in k-fold cross validation. The training process accommodates the time varying changes by continuously changing input data. This prediction model is proposed for availability zone selection process.

2. Availability Zone Working Model

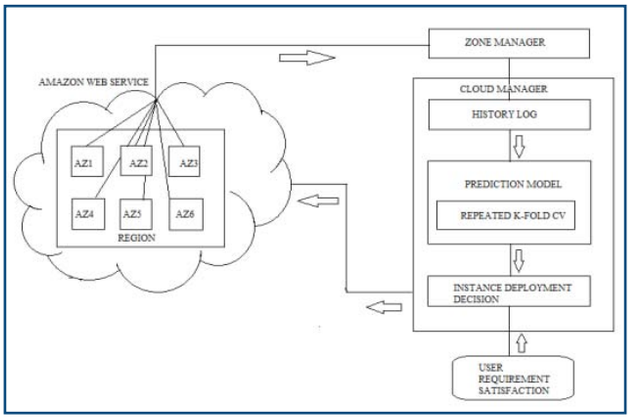
Amazon EC2 may get more difficulty while selecting an availability zone [Yoshua Bengio, 2004]. The fig 1 depicts the availability zone located in various regions. Selection of zone is done by major two parts shown in fig 1 Zone manager and cloud manager. Zone manager monitors the availability zones and calculates the satisfaction level. The monitoring process is done by monitoring tool like cloud watch which is provided by amazon EC2. This cloud watch monitors each zone and calculates satisfaction level vector. This measurement is send to the cloud manager. The prediction model and instance deployment decision are the two parts located inside cloud manager. The satisfaction vector and historical log are taken as input and training will be conducted using prediction model. The prediction model uses repeated k-fold cross validation for training process. The instance deployment decision is responsible for selecting a zone based on prediction result and user input requirement specification.

A user requirement specification is requested from user and represented as requirement vector (Rvec).

$$Rvec = [Rvec1, Rvec2, \ldots, Rvecn]$$

where $Rvec = 1, 2, \ldots, n$.

Figure 1. Architecture Diagram



The n numbers of requirement specifications collected are required to train the prediction model. These Rvec1, Rvec2… Rvecn are the various type of requirement received from the user, that are expected to be satisfied by the selected availability zone. The requirement attributes include QoS criteria, resource capacity, performance Constraints, nature of instance, type of machine ect [Juan, 2010].

Availability zone differ in characteristics. Zone size, hardware and software configuration, infrastructure type, deployment policies are differing from zone to zone. This variation may results in different QoS, reliability and performance. Before instance deployment in a particular zone, the attributes of such instance are not known. The zone manager shown in a fig 1 is responsible for monitoring the zone and captures the characteristics of zone. It gather the zone size, infrastructure type, hardware and software built version, recovery and failure rate attributes of the zone before instance deployment [Juan, 2010] [Kohavi, 1995]. Zone manager calculates the satisfaction level based on the measurements according to the satisfaction level and historical log, the prediction model predicts the suitable zone. Historical log consist of historical utility value of same user requirements. Based on utility value the prediction will perform.

Satisfaction level (Slev)

$$Slev => \{yes, no\}$$

$$Slev = \begin{cases} yes & if\ satisfied \\ no\ if\ not\ satisfied \end{cases}$$

$$Slevs = [Slev1, Slev2, \ldots, Slevn]$$

The satisfaction levels are only input given to cloud manager from zone manager. Based on the various levels of satisfaction level the cloud manager will perform prediction and deployment of the instance. To do prediction a prediction model is introduced in this paper.

3. Prediction Model

The prediction method explains how to predict the optimum availability zone across multiple availability zones [Yoshua Bengio, 2004]. It is done by learning process. Historical log is used to learn the zone behaviour. The log consists of historical utility value of same kind of instance. The requirement vector mapped to the incoming requirement specification using prediction model, which is denoted as utility value of Rvec.

$$F(Rvec) \in [0,1]$$

If the Rvec satisfies the Slev then it reaches maximum value of utility function otherwise reaches minimum utility value. The weight vector (Wvec) denotes the various significant levels of Rvec.

$$Wvec = [Wvec1, Wvec2, \ldots, Wvecn]$$

The utility value is defined as a linear combination of Wvec and Slev of each incoming request multiplied by indicator function (I(Rvec)). This indicator function is used to indicate the acceptance and rejection of request by setting the I(Rvec).

$$WS = Wvec * Slev$$

$$F(Rvec) = I(Rvec) * WS$$

$$I(Rvec) = \begin{cases} 0\ if\ request\ rejected \\ 1\ if\ request\ accepted \end{cases}$$

Each availability zone results in different characteristics for same instance even they are provided by same cloud provider [Juan, 2010]. We can predict the characteristics of zone for particular instance before its deployment. To predict the optimum zone with satisfaction level before deployment, a prediction model is used. The incoming request is randomly distributed to each zone using random selector and Slev is computed for each request and stored in historical log. Based on the Slev of incoming request the training table and prediction model is built for each zone. The training table shown in table: 1 consist training dataset which useful in building prediction model. The prediction model is built by using machine learning technique. It uses the training table dataset for both learning and prediction. A prediction model is denoted by Pk, it will trained by sample records and instances available in history log. The prediction model for given requirement vector is Pk(Rvec).

Table 1: Training Table

S.no	*Requirement vector*	*Learning target*
1	Rvec1	F(Rvec1)
2	Rvec2	F(Rvec2)
.	.	.
N	Rvecn	F(Rvecn)

After training phase Pk, learns the prediction process and it tries to predict the satisfaction level for the Rvec.

$$Pk(Rvec) = fu(Rvec)$$

Algorithm for Training and Validation Process

Step 1: Get "*Rvec*" from user

Step 2: Select sample data size "*Hs*"

Step 3: Define *fold = F*

where,

F = no. of folds

Step 4:Calculate

$$Fsize = \frac{Hs}{F}\ where\ Fsize = Fold\ size$$

Step 5: Splitting Calculation

$$Rk = V * Hs$$

Where,

Rk = Repeated k-fold split

$$V = F - \frac{1}{F}$$

Hs = Sample data size

Step6: Training Set

$$Th = RK$$

Where,

Th = Training set

Step 7: Validation Set

$$Vh = Hs - Th$$

Where,

Vh = Validation set

Hs = Sample data size

Th = Training set

Step 8: check Satisfaction vector

if (*Th satisfies Rvec*)

"training completed"

else

"*to be retrained*"

End

Step 9: Repeat until " *Th*" Satisfies *Rvec*

The error estimation of prediction process is calculated by using following calculation,

$$Err = \frac{1}{Ts + Zr} \sum_{k=1}^{Ts+Zr} \left[f - fu(Rvec) \right]$$

where,

Ts = Training set size

Zr = no. of request to a zone

Cross validation techniques often used in machine learning for model prediction. The data set which is available in training table is used for building prediction model. The dataset is split into two phases for cross validation, one for training and another for validation. Here we introduced "repeated k-fold cross validation" to build prediction model [Valerio Persico, 2015]. It allows to perform large number of estimation to avoid Type-I error. In case of performing cross validation for single time it may results in incorrect rejection of a true null hypothesis. Increasing the number of estimation will give accurate results. We recommend 50 estimations to perform and recommended fold size is 10 [Schad, 2010].

4. Availability Zone Selection

The prediction model Pk trained and started calculating utility value of satisfaction level for each zone. Based on the utility value stored in historical log, Pk started selecting an appropriate zone which satisfies requirement specification most. The zone satisfies the most will selected as

An availability zone for deployment of instance.

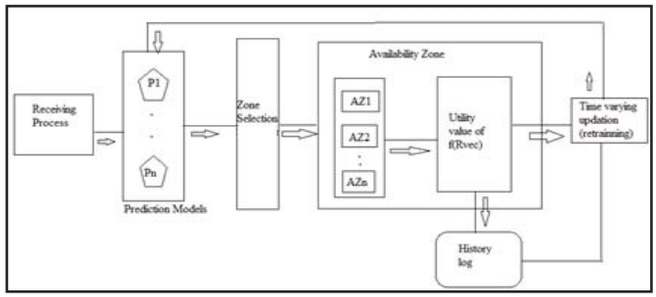
$$Pu(Rvec) = \max_{k} \{ Pk(Rvec) \}$$

After selecting an availability zone the respective instance will be placed in respective zone. The placed instance will tested for requirement satisfaction. The training phase accommodates time varying changes to adapt dynamic update. The placement decision in the cloud manager takes the decision of zone selection process. Based on the requirement specification, weight vector and satisfaction level in history log the placement decision will take.

5. Experimental Results

Availability zone selection with user satisfaction level will performed by three various modules namely receiving process, zone selection and availability zone. The first component is availability zone it simulates the empty zone with its characteristics. The second component is receiving process it will randomly distribute the received requirements across availability zones and the zone manager monitors the arrival process and stores the satisfaction level in history log using receiving process. The built prediction model selects the optimum zone which satisfies the requirement specification most using third component zone selection. The instance will deploy in the zone after the cloud manager takes the decision about zone selection using availability zone. Thus the simulation process will performed to automatic selection of optimum availability zone with user satisfaction.

Figure 2. Working Model



We compare random policy, prediction based policy, least utilized, match instance policy and match machine policy to deploy and zone using k-fold CV and repeated k-fold CV. In our results the trails taken are around 50 for each policy to deploy the instance in optimum zone. In k-fold CV the trails might low compared to repeated k-fold CV. It overcomes the Type-I error occurred in k-fold Cross Validation. The increasing number of trails will give accurate selection of zone to deploy the instance shown in the fig: 3. these policies give best results only by performing repeated estimation. Here the number of trails is decreased while performing repeated k-fold cross validation. It allows to perform large number of estimation to avoid Type-I error. In case of performing cross validation for single time it may results in incorrect rejection of a true null hypothesis. Increasing the number of estimation will give accurate results. We recommend 50 estimations to perform and recommended fold size is 10. The following figures show the trail and zone match percentage.

Figure 3. Trails Taken to Deploy Instance

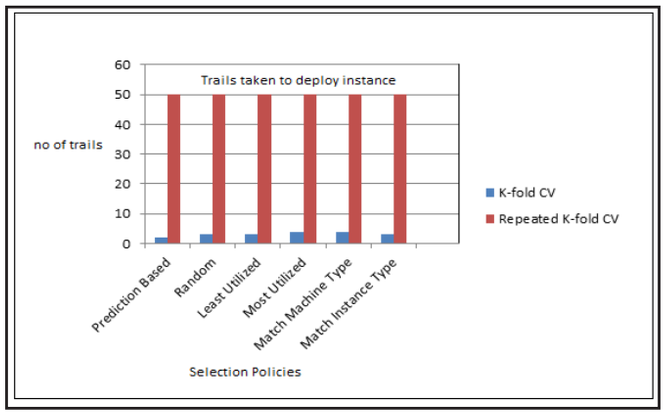
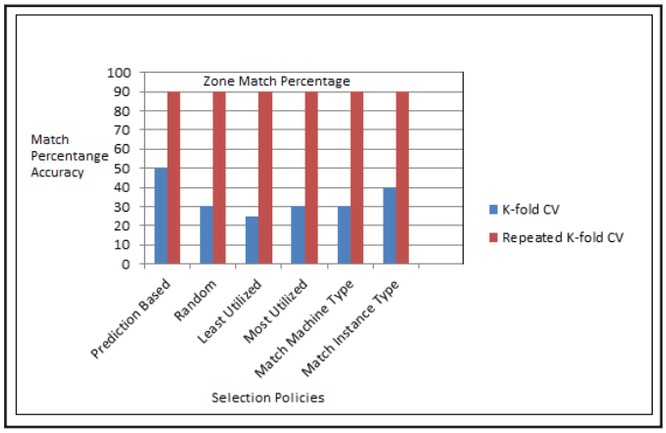


Figure 4. Zone Match Percentage



Zone match percentage is matching percentage of requirement vector and accurate zone performance. Repeated k-fold CV matches the requirements up to 90% [Schad, 2010] shown in fig: 4 compared to k-fold CV, repeated k-fold CV achieves more accuracy.

6. Conclusion

The automatic selection of availability zone to deploy the amazon EC2 instance is performed to increase user satisfaction level. The prediction model is developed for zone selection. The machine learning technique, repeated k-fold cross validation is used for training the prediction model. As a result the prediction model built using repeated cross validation accommodates the time varying changes and dynamic updates by performing 50 estimates. The type-I error is removed in prediction process to achieve accurate prediction in zone selection. Then the optimum availability zone selection is made for an instance.

References

1. Kaminski, B., & Szufel, P. (2015). On optimization of simulation execution on Amazon EC2 spot market. *Simulation Modelling Practice and Theory*, 58, 172-187.
2. Farley, B., Juels, A., Varadarajan, V., Ristenpart, T., Bowers, K. D., & Swift, M. M. (2012). *More for your money: Exploiting performance heterogeneity in public clouds*. In Proceedings 3rd ACM Symposium Cloud Computing, (pp. 1-20).
3. Vanwinckelen, G., & Blockeel, H. (2012). *On Estimating Model Accuracy with Repeated Cross-Validation*. Appearing in Proceedings of Bene Learn and PMLS.
4. Rodrıguez, J. D., Perez, A., & Lozano, J. A. (2010). *Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, March, 32(3), 569-575.
5. Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation and model selection*. In Proceedings of 14th International Joint Conference on Artificial Intelligence, 2(12), 1137-1143.
6. Li A., Yang X., Kandula S., & Zhang, M. (2010). *Cloud CMP: Comparing Public Cloud Providers*. In

Proceedings of 10th ACM SIGCOMM Conference on Internet Measurement, (pp. 1-14).

7. Last, M. (2006). *The uncertainty principle of cross-validation*. 1--4244--0134--8/06/$20.00 © 2006 IEEE Unuvar, M., Tosi, S., Doganata, Y. N., Steinder, M. G., & Tantawi, A. N. (2015). *Selecting optimum cloud availability zones by learning user satisfaction levels.* IEEE Transactions on Services Computing, March/April, 8(2), 199-211.
8. Refaeilzadeh, P., Tang, L., & Liu, H. (2008). *Cross-validation.* Retrieved from Bvijayalakshmigalleys0000875816 (accessed on November 6, 2008)
9. Schad, J., Dittrich, J., & Quiane-Ruiz, J. A. (2010). *Runtime measurements in the cloud: Observing, analyzing, and reducing variance*. Proceedings of VLDB Endowment, 3(1-2), 460-471.
10. Persico, V., Marchetta, P., Botta, A., & Pescapè, A. (2015). *Measuring network through put in the cloud: The case of Amazon EC2 computer networks*, Elsevier, (pp. 1-15).
11. Bengio, Y., & Grandvalet, Y. (2004). No unbiased estimator of the variance of K-fold cross-validation. *Journal of Machine Learning Research*, 5(1), 1089-1105.