A BEHAVIOR CLUSTER BASED AVAILABILITY PREDICTION APPROACH FOR NODES IN DISTRIBUTION NETWORKS*

Jiali You[1], Jiao Xue[1,2], Jinlin Wang[1]
[1] National Network New Media Engineering Research Center,
Institute of Acoustics, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China

ABSTRACT

To predict the availability state of a node in a distribution network, its history trace is usually used. Sometimes, some usage behavior patterns cannot be captured precisely from the insufficient trace, which may lead to unreliable predictors. In this paper, to alleviate the data sparseness problem, the nodes with the similar behaviors are clustered, and all history information in a same cluster is seen as another information source for any node in it. For each node, an N-gram model is used to train the predictor by the combination of the new source and the node's own trace. In addition, because it is hard to capture the trace of all nodes in large scale networks, such as P2P networks, a bagging based prediction algorithm is proposed, which can be applied in the distribution environment and relieve the effect of the noisy data. In our experiments, three datasets are evaluated. Results show that the prediction performance of our cluster based N-gram predictor is better than the results of several other predictors. And the bagging based prediction algorithm presents its validity in the distribution environment.

Index Terms— Availability prediction, distribution network, N-gram, cluster, K-means, bagging algorithm

1. INTRODUCTION

Recently, a lot of work has proved that if one could predict the availabilities of even a portion of nodes with reasonable accuracy, the performance of distribution systems of different applications can be improved obviously[1-6], such as P2P storage system, and Grid network. Therefore, it is very important to predict the availabilities of nodes correctly and use them to guide the scheduling process in various applications.

In the current work, most of the prediction algorithms tracked fine-grained, per-node up-down states to train the predictor by some machine learning approaches[1-7].Usually, in different applications, just on-line or off-line states are traced, and the logged information is used to predict the node's succeeding availability. Mickens and Noble[5, 8, 9] proposed a hybrid predictor to predict the probable state of a host for various lookahead periods. In their approach, multiple simple predictors are generated and cooperated as a hybrid one by a mixture-of-experts approach. In [10], the availabilities of machines are modeled by different distribution functions, such as Weibull, hyper-exponential and Pareto. Andrzejak etc.[11] used a Naïve Bayes classifier to forecast the availability of nodes, and several features are compared. Besides, another work focuses on the data that both the "up" or "down" information and the unavailable reasons are recorded. These traces are commonly obtained from Grid computing networks. In some scenarios, different reasons[2, 6, 7]cause the unavailable states of nodes, for instance, memory thrashing and resource unavailability. The transition probabilities from available state to other states are captured by Semi-Markov model. Apart from these, for predicting the rare target events from time-series data, several machine learning techniques are also used to mine the patterns which precede the target events[1, 12, 13]. Actually, in some distributed systems, it is hard to obtain the exact reasons of the unavailability events. Therefore, predicting the future availability with limited information, such as up/down states, is crucial for a lot of applications, and our work focuses on this one.

As we know, if just the trace of a node is used, the sparseness problem will be hardly avoided, and the predictor trained based on it may be unreliable. In real applications, although people may operate their machines irregularly, some of them may have similar usage patterns, like online in work time and offline in midnight[14], or watching some programs in some specific time[15, 16]. In addition, in some networks, different nodes may cooperate in computing or communicating[17] and the failures appearing in one node may be propagated across the network. In other words, some people's behavior may be captured from the trace of others who have similar habits or interests to him/her. However, in the distribution network, the global information, like the trace of all nodes, is hard to be obtained. Therefore, in this paper, we propose a novel approach to design a predictor in a distribution network and evaluate it with various datasets. To summarize, the paper's main contributions include:

- Designing a cluster based N-gram predictor, which can learn more behavior patterns and obtain better performance than several other predictors.

* The work is supported in part of the NSF of China(60903218), 863 Program(2011AA01A102), Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06010301) and the Innovation Project of Institute of Acoustics, Chinese Academy of Sciences(Y154211601).

- Proposing a bagging based prediction framework, in which the N-gram predictor is applied to obtain the preferable performance in distribution applications.

The remainder of this paper is structured as follows. Section 2 illustrated the cluster based N-gram predictor. The bagging based prediction algorithm in a distribution network is introduced in section 3. The section 4's experiment evaluates our approaches. Finally, conclusions are drawn in section 5.

2. THE BEHAVIOR CLUSTER BASED N-GRAM PREDICTOR

In our work, the "up" and "down" states are represented by binary values 1 and 0, respectively. And each point is recorded with some specific sampling rate, for example, one record every 30 minutes. Then, for each node, a 0/1 sequence S is collected to represent its states transitions during the observation period. In data pre-preparing stage, all collected nodes should be clustered as several clusters based on their traces. Then, the traces of the nodes in a same cluster will be seen as a new information source for any node in it. For any node, the collected traces of the corresponding cluster and its own trace are combined together as the training set, which can alleviate the data sparseness problem. Finally, a predictor will be trained based on this combined data set. Here, the N-gram model is used for generating the predictor. The process of the predictor training is shown in figure 1, and the details of the clustering algorithm and the N-gram model will be introduced following.

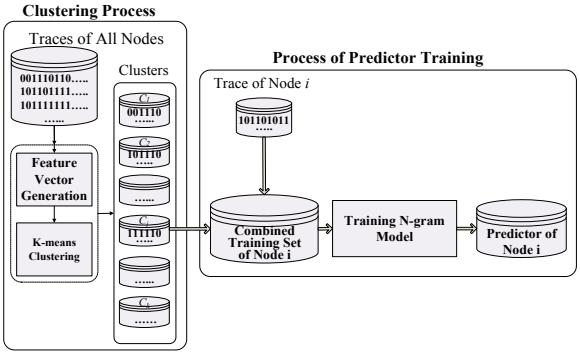

Figure 1. The training process of the cluster N-gram model

2.1 K-Means based clustering algorithm for similar behavior nodes

In general, clustering algorithm can be usually used to identify the similar patterns in a sequence or category. K-means is a simple algorithm that can be implemented easily for clustering. In our paper, we use it to cluster the nodes based on their up-down behaviors. Actually, the nodes may randomly enter or leave a network and they may be scattered in different time zones around the world. It is hard to cluster based on the original trace. To solve this problem, we should extract the feature vectors from it.

Given a node h, the length of the time during its first online state to its last recorded state is denoted as OT_h. Firstly, the trace of each node should be segmented as the same length fragments, which is based on the usual periods of people, e.g. 24 hours, and 48 hours. The time length of a fragment is denoted as ST; and then l_h fragments will be generated, where $l_h = \lceil OT_h / ST \rceil$. Thus, the state vector of any fragment is SV which contains d states. Here, $d = \lceil ST / \Delta t \rceil$ and Δt is the time interval between two sequential states in the trace. Finally, the feature vector BV_h can be calculated by formula (1), where $SV_{h,i}$ is the state vector of fragment i.

$$BV_h = \frac{\sum_{i=1}^{l_h} SV_{h,i}}{l_h} \tag{1}$$

Based on the above procedure, all nodes will get their behavior feature vectors. These vectors are the input of a standard K-means algorithm, in which the objective function is to minimize the summation of the distances of all nodes to their cluster centroids. Then, the node trace in a same cluster becomes a new information source for any node in it.

2.2 N-gram predictor

N-gram statistical modeling techniques have been applied successfully to speech, language and other areas with the data of sequential nature[18]. For the availability prediction problem, the observed samples have the time-sequence property. The m[th] state may be related to the previous m-1 states. Accordingly, the relationship among sequential states is calculated by formula (2).

$$\begin{aligned} cs' &= \arg\max_{cs \in CS} \{P(cs \mid S_h)\} \\ &= \arg\max_{cs \in CS} \{P(cs, S_h)\} \\ &= \arg\max_{cs \in CS} \{p(cs \mid s_{h,m-1} \ldots, s_{h,1}) \times \prod_{i=1}^{i=m-1} p(s_{h,i} \mid s_{h,i-1}, \ldots, s_{h,1})\} \end{aligned} \tag{2}$$

where $CS = \{cs_1, \ldots, cs_e, \ldots, cs_E\}$ is the collection of all candidate states and E is the total number; $S_h = \{s_{h,1}, s_{h,2} \ldots, s_{h,m-1}\}$ is the state sequence in the history trace of node h, and the m[th] state is to be predicted. In our paper, $CS = \{0,1\}$ where 0 represents "down" and 1 represents "up". The CMU statistical language modeling toolkit[19] is used to train the conditional probabilities among neighbors with back-off smoothing and linear discounting[20], and a unique predictor is trained for each node. When predicting, formula (2) is calculated and the candidate state which is contained by the sequence with the highest probability is the final decision.

3. BAGGING BASED PREDICTION ALGORITHM IN DISTRIBUTION NETWORKS

The basic idea of bagging algorithm[21, 22] is to generate different models and use them to get an aggregated predictor, in which the final result is decided by voting. This algorithm can reduce variance and help to avoid overfitting for classifiers, which can be used in the distribution environment that just part of the trace of nodes is collected.

In our work, when training a predictor for node *h*, it is assumed that $L=\{S_i, i=1,2,...,I\}$ is the trace dataset of nodes which are in a same cluster, and S_i is the state sequence of node *i*. To get a reliable result, we use the resampling approach that *r*% of all nodes in *L* are randomly selected, like $L_g=\{S_g, n=1,2,...,\lfloor I\times r\% \rfloor\}$, where *r* is a preset experience parameter. This sub-set is combined with the node *h*'s own trace for training the predictor as the approach shown in section 3. Then, a cluster based N-gram predictor $\Phi_{L_g}(X)$ is generated. This process loops *G* times, and *G* different predictors $\{\Phi_{L_1}(X),\Phi_{L_2}(X),...,\Phi_{L_G}(X)\}$ are obtained, where *G* is an odd number and *X* is the history trace of the predicted node. For the m[th] observed point, the predicted state is decided by formula(3). Here, $S_{h,m-1}=\{s_{h,m-w-1}, s_{h,m-w}...,s_{h,m-1}\}$ means the history trace of node *h* and $s\in\{0,1\}$.

$$s_{h,m}=\begin{cases}1 & if \sum_{g=1}^{g=G}\Phi_{L_g}(S_{h,m-1})\geq\left\lceil\frac{G}{2}\right\rceil\\ 0 & otherwise\end{cases} \quad (3)$$

In real applications, the trace of nodes can be collected by the overlay or network maintaining process, which cannot cause huge overhead.

4. EVALUATION AND DISCUSSION

4.1 Experiments setup

To evaluate the adaptability of our approach in real applications, we tested it on three availability traces from different distribution systems, including Overnet[23], Skype[24] and Microsoft Corporate Network[25]. After filtering the nodes with few states changing, just 568 nodes are left for Overnet dataset. Also, we randomly select 568 nodes for any other dataset. For each one, the preceding 80% samples from the first point are set as the training data. Details of the other information are listed in table 1.

Table 1.Used Traces

Data Set	Overnet	Skype	Microsoft
Stability	high churn	steady	steady
Number of Samples	504	688	840
Sampling Interval	20 Minutes	30 Minutes	1 Hour
Number of Training Points	360	480	504

In our experiments, we tuned the parameters with different values, and finally N=6 for the N-gram model and K=3 for the K-means algorithm are selected. In addition, because our work is similar to the one in[5], several predictors referred are also implemented for comparing, including saturating counter predictor, state-based predictor and hybrid predictor. And the N-gram predictor trained by the node's own trace is also compared.

4.2 Performance analyzing

4.2.1 Comparison of the performance from different predictors

During the evaluation period, the accuracy of one lookahead interval forecast(One-Step), which is determined by predicting the state of a point with its all previous states trained is compared. Table 2 gives the average value of the One-Step accuracy of different datasets.

Table 2. The Average One-Step Accuracy (%)

	Overnet	Skype	Microsoft
Cluster N-gram	87.85	98.77	98.59
N-gram	87.15	98.77	98.42
Hybrid	86.62	98.42	98.06
State-based	86.22	98.73	98.03
Saturating counter	78.17	92.96	96.48

From this table, we can see that the performance from steady network, such as Skype and Microsoft, are better than the result of Overnet network. Because the potential law of the state transitions in steady networks can be learned easily by predictors. Saturating counter just records the number of states and ignores the relationship among the sequential states, which gets the lowest accuracy. Contrarily, N-gram model takes the advantage in sequential data studying, especially for the cluster based N-gram. The back-off smoothing and discounting strategies give the reasonable probabilities of unseen usage patterns, which may ensure the robustness of the predictors.

4.2.2 Comparison of the predicting capability for different predictors

In [5], hybrid predictor takes good advantages from the combination of the different predictors by a mixture-of-experts approach. Accordingly, we also investigate whether our cluster based N-gram should be combined with other predictors for future improvement. Thus, we compare the non-overlap part of the predicted results from different predictors as Table 3, which means the portion of the testing points correctly predicted just by one predictor.

Table 3. The portion of the non-overlap part of the results (%)
CR: Cluster based N-gram right and other predictor wrong
CW: Cluster based N-gram wrong and other predictor right

	Overnet		Skype		Microsoft	
	CR	CW	CR	CW	CR	CW
N-gram	1.08	1.18	1.3	0.09	3.0	0.24
Saturating	3.15	1.66	1.51	0.21	1.45	0.43
State-based	3.35	1.84	0.03	0.05	0.28	0.14

From Table 3, we know that the cluster based N-gram can capture most behavior patterns, especially for the nodes with steady patterns. Therefore, the predicting capability of our predictor is sufficient enough to get the satisfying result.

4.2.3 Performance of other metrics

The accuracy of multiple lookahead intervals forecast(M-Step) is also an important metric, which is to predict the state of the M[th] point with the previous M-1 states predicted by the predictor. This situation is usually used in the applications, like P2P storage system, in which the states of a node after a long time should be known in advance. In our experiments, we found that if the One-Step performance is satisfying, which means fewer errors of the M-1 previous states involving in the M-Step forecasting process, the corresponding M-Step performance is also good. Also, the average M-Step result of cluster N-gram is better than the

other predictors in most observed points. Moreover, although the performance of some other predictors is comparable with the cluster N-gram model, the predicted states of some nodes are prone to be skewed to one state. To analyze this problem, we design a balance factor $\eta = \frac{\text{Average } Accuracy\ of\ Up\ Class}{\text{Average } Accuracy\ of\ Down\ Class}$ to evaluate whether the performance of different classes is balance. Apparently, our purpose is to get the predictor whose balance factor is close to 1. Figure 2 shows the comparison results. From this figure, we know that the cluster based N-gram achieves the flattest curve among the compared predictors, which means that it can capture the usage patterns more precisely than others and reduce the incorrectly classifying for some class that just contains a few samples.

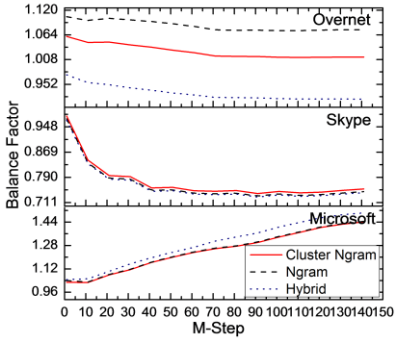

Figure 2. The balance factor in the M-Step forecast

4.3 Evaluating the performance of the bagging algorithm

In a real distribution network, a node can just see some nodes around it. Thus, we simulate a distribution environment, where each node just collects a certain portion z% trace of all nodes for training. In addition, in bagging process, *r*% training set is selected randomly to train each predictor. Here, we select 5%, 10% and 20% for z%, which cannot cause too much communication cost in a network, such as P2P applications. Also, the value of *r*% is tested from 50%~100%.

Figure 3 proves that if parts of the traces are collected and the appropriate parameters are selected, the bagging based predictor shows the comparable performance with the result from the model trained by 100% dataset, evenly better than it. The probable reason is that the effect of the noisy data which is mixed in the training set is reduced by the voting process of the bagging algorithm. Accordingly, in distribution networks, just observing some neighbors' information is sufficient enough to train a reliable predictor.

4.4 Discussion

In a P2P application, it is not difficult for each node to observe 50 nodes from its neighbors or neighbors' neighbors. Usually, the state sampling rate is about 10-30 minutes, which means that a node just check the state of one neighbor per 0.2-0.6 minutes. This time interval is very large and it cannot raise too much communication burden for the network. Consequently, the bagging based predictor can be easily applied in different applications, such as P2P streaming systems and P2P storage systems.

Besides, although an N-gram model is used in this paper, the cluster information and the bagging framework can also be applied to other prediction models, for example, Markov chain and the Naïve Bayes Classifier, and we will deeply investigate it in our future work.

5. CONCLUSION

In this paper, we use the k-means algorithm to cluster the nodes into different clusters based on their user behavior similarities, and the collection of all trace in a cluster is seen as a new information source to enhance the robustness of training set for any node in it. To apply this predictor in a distribution network, a bagging based algorithm is proposed. Evaluation indicates that the N-gram model can obtain good performance in availability prediction. Also, in a distribution network, the bagging based algorithm can achieve satisfying performance. If the parameters are selected carefully, it even gets the better result than the predictor which is trained by the all nodes information with the help of a global center.

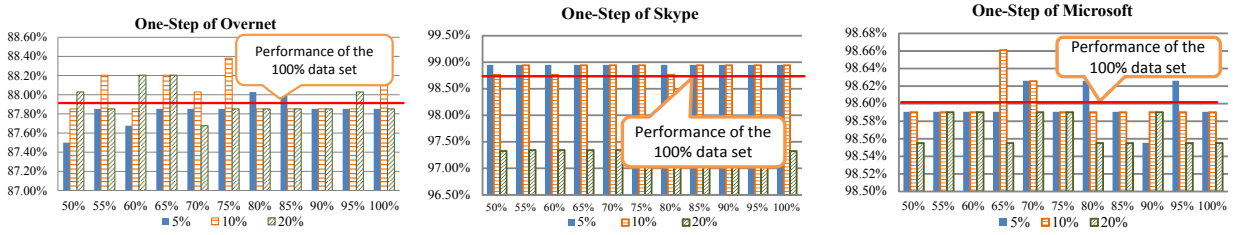

Figure 3. The performance in the distribution environment

6. RELATION TO PRIOR WORK

The purpose of the presented work is to predict the node availability in distribution networks. Similar problem has been investigated in [1-5,11]. For these works, authors considered several different simple predictors[1-5,11], different features[11], or used a combined predictor by mixture-of-experts approach to enhance the prediction performance[5]. Commonly, the history trace of a node is used for training the model. However, as we know, we cannot have too much time to observe the states of nodes before training the predictor, which may lead to a data sparseness problem. Therefore, our work focuses on how to train a reliable predictor by the help of other nodes and how to use it in the distributed environment. Moreover, this approach may be also useful for different prediction models.

7. REFERENCES

[1] R. Bhagwan, K. Tati, Y. Cheng, and etc., "Total Recall: System Support for Automated Availability Management." In Proc. of NSDI (2004), pp. 337-350.

[2] X. Ren, S. Lee, R. Eigenmann, and S.Bagchi, "Resource Failure Prediction in Fine-Grained Cycle Sharing System," Journal of Grid Computing (2006), Volume: 5, Issue: 2, pp. 173-195.

[3] R B.ood, M. J.Lewis, "Resource Availability Prediction for Improved Grid Scheduling," In Proceedings of the 2008 Fourth IEEE International Conference on eScience Table of Contents, pp.711-718, 2008.

[4] B. Rood, M. J. Lewis, "Multi-State Grid Resource Availability Characterization," In International Conference on Grid Computing, pp.42-49, 2007.

[5] J. W. Mickens, B. D. Noble, "Exploiting Availability Prediction in Distributed Systems," In Network Systems Design and Implementation, pp. 73-86, 2006.

[6] X. Ren, R.Eigenmann, "Empirical Studies on the Behavior of Resource Availability in Fine-Grained Cycle Sharing Systems," In Proc. ICPP'06, pp. 3-11, 2006.

[7] X. Ren, S. Lee, R.Eigenmann, and S. Bagchi, "Prediction of Resource Availability in Fine-Grained Cycle Sharing Systems Empirical Evaluation," Journal of Grid Computing, 5(2):173-195, 2007.

[8] J. W. Mickens, B. D. Noble, "Improving Distributed System Performance Using Machine Availability Prediction. SIGMETRICS Performance Evaluation Review (SIGMETRICS)," 34(2):16-18 (2006).

[9] J. W. Mickens, B. D. Noble, "Predicting Node Availability in Peer-to-peer Networks. In International Conference on Measurement and Modeling of Computer Systems," pp. 378-379, 2005.

[10] D. Nurmi, J. Brevik, and R. Wolski, "Modeling Machine Availability in Enterprose and Wide-Area Distributed Computing Environments," In Europar, pp. 432-441, 2005.

[11] A. Andrzejak, D. Kondo , and D. P. Anderson, "Ensuring Collective Availability in Volatile Resource Pools via Forecasting', 19th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2008), Samos Island, Greece, September 22-26, 2008.

[12] R. K. Sahoo, A. J. Oliner, I. Rish, and etc., "Critical Event Prediction for Proactive Management in Large-scale Computer Clusters," In Special Interest Group on Knowledge Discovery and Data Mining, pp. 426-435, 2003.

[13] G. M. Weiss, H. Hirsh, "Learning to Predict Rare Events in Categorical Time-series Data," In International Conference on Machine Learning, pp. 83-90, 1998.

[14] S. Saroiu, P. K. Gummadi, S.D. Gribble, "A Measurement Study of Peer-to-peer File Sharing Systems," In Proceedings of Multimedia Computing and Networking (MMCN) 2002, January 2002.

[15] B. Chang, L. Dai, Y. Cui, and Y. Xue, "On Feasibility of P2P On-demand Streaming via Empirical VoD User Behavior Analysis," The 28th International Conference on Distributed Computing Systems Workshops, pp.7-11, 2008.

[16] M. Vilas, X.G.Paneda, R. Garcia, and etc, "User Behaviour Analysis of a Video-on-demand Service with a Wide Variety of Subjects and Lengths," EUROMICRO-SEAA, pp.330-337, 2005.

[17] S. Fu, C. Xu, "Exploring Event Correlation for Failure Prediction in Coalitions of Clusters," Proc. of the ACM/IEEE Supercomputing Conference (SC), SC2007:41, 2007.

[18] C. Manning, H. Schütze, "Foundations of Statistical Natural Language Processing," MIT Press. Cambridge, MA: May 1999.

[19] http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html

[20] H. Ney, U.Essen, and R.Kneser, "On Structuring Probabilistic Dependencies on Stochastic Language Modeling. Computer Speech & Language," 8(1): 1-38(1994).

[21] E. Bauer, R. Kohavi, "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants," Machine Learning, vol.36, pp.105-139 (1999).

[22] L. Breiman, "Bagging Predictors," Machine Learning, vol.24, pp.123-140(1996).

[23] R. Bhagwan, S. Savage, and G.M.Voelker,"Understanding Availability," IPTPS pp.256-267, 2003.

[24] S. Guha, N. Daswani, and R. Jain, "An Experimental Study of the Skype Peer-to-Peer VoIP System," In Proceedings of The 5th International Workshop on Peer-to-Peer Systems (IPTPS), pp. 1-6, 2006.

[25] W. J. Bolosky, J. R. Douceur, D. Ely and etc., "Feasibility of a Serverless Distributed File System Deployed on An Existing Set of Desktop PCs," In Proc. SIGMETRICS pp. 34-43, 2000.