Failure Diagnosis for Configuration Problem in Storage System

Chung-hao Tan
IBM Almaden Research Center
chungtan@us.ibm.com

Abstract

Configuration problem in storage system is one of the well recognized issues in today's enterprise environment. Traditional approaches usually reply on deep knowledge of the underlying system. However, such solution usually cannot scale up when the system become large and complex. Recent research work has started using machine learning techniques to improve the overall performance of the diagnosis process. In this project, we try to identify the proper procedure to transform a configuration problem into a machine learning problem, which lays down the foundation of our future work in the area of configuration and change management.

1. Introduction

One of the key challenges in the current enterprise storage environment is to quickly identify the root cause of the failure so that the system down time can be reduced. Some storage management products [11] have partially addressed this issue by providing an aggregated view of the system status across multiple layers and components. However, even with the help from such management tools, fault localization in a complex storage environment is still a time-consuming task which requires an experienced administrator to manually interpret a vast amount of diagnosis data. As a result, an automated failure diagnosis tool which requires less human intervention would be necessary in a mission critical environment.

Traditional fault isolation techniques usually rely on manually-built system model [18], which is either difficult or impossible to be created in a dynamic, multi-vendor environment. Recent studies have shown the promise of the black-box approach [22, 21, 7, 5, 4, 1, 12], that is, can we perform failure diagnose by observing the system behavior without knowing too many details about it? Not surprisingly, machine learning techniques are widely used in such setup.

In this project, we try to identify the key elements and the proper procedures in a black-box approach that automates the fault diagnosis process. Specifically, we focus on solving SAN (Storage Area Network) configuration problems which display complex characteristics such as multi-tier relationship, interoperability, policy and security. The motivation behind this is that the configuration problem has been recognized as the top reason for system crash [13, 9, 23], which also matches to our observation in the storage environment.

Our goals/contributions are as follows:

1. We survey a wide range of recent papers in the area of fault diagnosis. Since this project is conducted for the machine learning course, we emphasize on those papers that use machine learning techniques.

2. We show solutions that can transform a configuration problem into a machine learning problem in a black-box approach. We highlight their objectives and limitation.

3. We identify future research work that may advance from previous work. For example, multi-fault diagnosis is still an open research problem.

To our knowledge, this is the first work that uses machine learning techniques to automate the failure diagnosis process for the SAN configuration problem. We intend to continue the work after the course. We learn the necessary background and obtain insightful ideas through the work of this project. In the followings, we describe the related work in section 2. Some suitable learning approaches are explained in section 3. Experimental results are shown in section 4. And we list our future work in section 5.

2. Related Work

Failure diagnosis in the distributed system has been studied quite long [18]. Traditional approaches usually rely on explicit modeling of the underlying system. One widely used technique is the knowledge-based *expert system* (e.g. Help Desk). Such approach may work well in a controlled, static environment. However, as the complexity of the system keeps growing, it becomes impossible to encompass all the necessary details. For instance, the device interoperability matrix is so dynamic that no one can ever build a complete matrix. Applications on top of the network add another layer of complexity (e.g. Disk traffic and Tape traffic cannot flow through the same HBA).

Recent research has focused on the *black-box* approach in order to handle the complexity and the non-deterministic behaviors of the managed system. One typical setup is the *symptom-based* approach. By observing system states from the results of probings, a symptom-based approach tries to predict the cause of failure from past experience. Unsurprisingly, machine learning techniques are adopted in most recent papers in the black-box approach. Among those various learning algorithms, *Decision Tree* [5], *Naïve Bayes Classifier* [17, 14, 21, 10, 8], *Bayesian Network* [6, 16] and *Clustering* [4, 7, 12, 10] are widely used due to their advantages of *interpretability* (i.e. easy to explain the learning results) and *modifiability* (i.e. easy to incorporate expert knowledge into the model) [7, 20]. These two properties make these algorithms attractive to the human operators in practice.

In machine learning's perspective, failure diagnosis can be viewed as an *anomaly detection* problem [10, 12, 19, 20], especially if the majority of the training samples are negative case (i.e. good case). The key idea is to describe what a good data looks like, then an anomaly is the case that it cannot fit into any of those good models. The concept of feature selection and model selection are also explored in several recent papers [14, 24]. We will discuss those machine learning approaches in more details in the next section.

History-based approach is another useful technique which reduces the solution space significantly [22, 23]. The idea is to compare two snapshots of the system states, one from current non-working state and the other from last known working state. Then there is a good chance that the failure comes from one of the changes in the difference. Two assumptions are usually made in this approach: First, the difference between two snapshots are not many. Second, constantly changing element is usually not the factor. We can also make the comparison between similar systems, which may increase the accuracy by taking advantage of large number of samples, but it add additional complexity due to the difference between system.

How to increase the overall usability in the diagnosis process is one of the interesting topics in this field. For example, an integrated visualization environment usually increases administrator's confidence on the learning results [3]. A knowledge database can assist human operators to make the final decision when the program cannot do any further diagnosis [22]. Proactive approach [15] is also a useful technique to avoid the problem in advance.

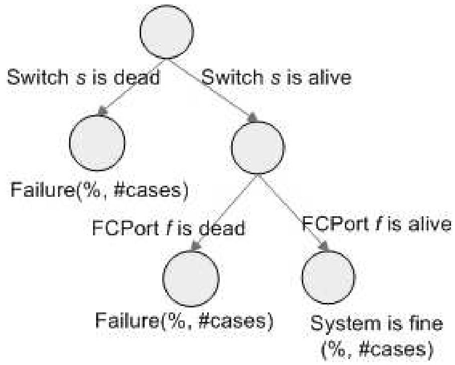

Figure 1. Modeled as Decision Tree

Diagnosis of configuration problem has been studied in several areas such as Windows Registry [22, 21, 12, 9], router configuration [8] and general Internet application [13]. Other recent research focus on either performance problem [1, 7], software failure [4, 5] or general fault localization techniques [16, 17, 14, 2].

3. Machine Learning Approach

Before further discussion, let's see some real examples of SAN configuration problems collected from the field. Table 1 shows three cases which we consider them difficult to diagnose. For instance, in case 1 we see an error code which indicates a hardware fault. However, the real cause is actually from a misconfiguration at the host side. An inexperienced administrator may look into the problem in the wrong direction. To help out this situation, our goal is to give a short list of possible reasons to the users, so that they can spend less time in the diagnosis process.

Machine learning techniques can help achieve the goal at both the design time and the runtime. At the design time, which can be viewed as a *proactive* approach, we try to discover some diagnosis rules that are not trivial. Decision Tree algorithm [5] (see Figure 1) is an ideal solution that fits into this domain. In addition, we can also optimize those diagnosis rules by using the concept of feature selection and model selection. We run a preliminary experiment to validate this approach (see Section 4).

Furthermore, at the runtime (a *reactive* approach) we try to predict the source of failure by taking advantage of past experience or similar configuration from the peers. We can view it as a classification problem in machine learning's terminology. The most commonly used technique is the Naïve Bayes classifier [17, 14, 21, 10, 8] (Figure 2). We can think it as a *symptom-fault map* where each edge represents the conditional probability between the symptom and fault. We usually make some assumption on the prior and even on the

Case	Symptoms	Fault (Reason of Failure)
1	1. Numerous TSM error code ”ANRxxxx” (hardware fault) 2. Loss of tape drive and later re-discover it incorrectly	Disk traffic and Tape traffic are on the same HBA
2	1. Numerous and random loss of external disk (Windows and AIX) 2. Occasional loss of data (Windows only) 3. Rebooting Windows server takes 30+ minutes	# entity in a single zone is too large
3	1. Severe performance reduction 2. See ”... complete I/O failure ...” in the log	# path between host and device is too large

Table 1. Examples of SAN Configuration Problem

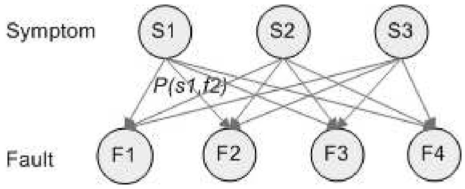

Figure 2. Modeled as Naïve Bayes Classifier

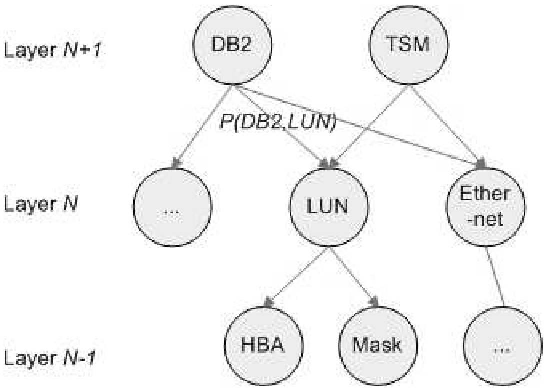

Figure 3. Modeled as Bayesian Network

conditional probability. Sometimes, EM algorithm is used to train the model [10, 3]. Some previous work also use Bayesian Network [6, 16] (Figure 3) to incorporate prior knowledge of the system model (e.g. ISO/OSI network model).

We can also perform the classification by clustering [4, 7, 12, 10] (Figure 4). For example, in the the anomaly detection approach we generate a set of *good* clusters, then anomalies are those samples that cannot fit into any of those good clusters.

We review some learning algorithms that are suitable for the configuration problem in this section. Next, we will show a concrete example of solving probe selection problem by using a machine learning approach.

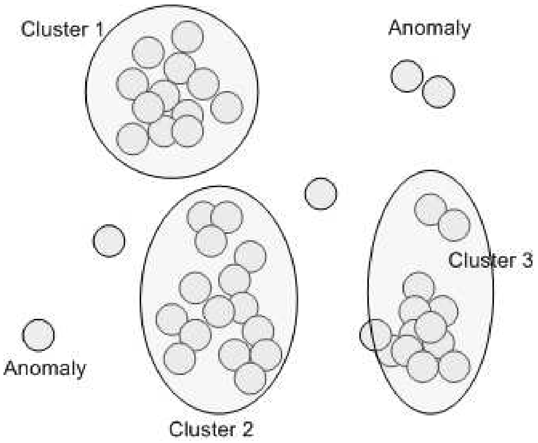

Figure 4. Modeled as Clustering

4. Experimental Result

Configuration problem is different to other type of problems (such as performance problem) is in the sense that the number of variables are usually quite large (e.g. there are about 70K registries in a fresh WindowsXP installation). However, once the system is settled down, the configuration usually doesn’t change often. Note that these two properties bring undesired effects to the learning results (i.e. number of features and number of training samples).

On the other hand, the cost of running a probe in order to gather system state is usually un-ignorable. Ideally, we want to run a minimum set of probes to diagnose a particular failure. But we usually don’t know how to select such set. Optimization is even more difficult if we want to diagnose multiple failures simultaneously.

One possible solution for this problem is to treat it as a *feature selection* problem in machine learning. We run a simulated experiment to validate this concept. We generate the sample data by using the multi-variate Bernoulli event model, given $P(Y = 1) = 0.05$. Those data are trained and test by Naïve Bayes classifier with hold-out cross validation (70%/30%). The greedy forward search procedure is used to generate the set of probes.

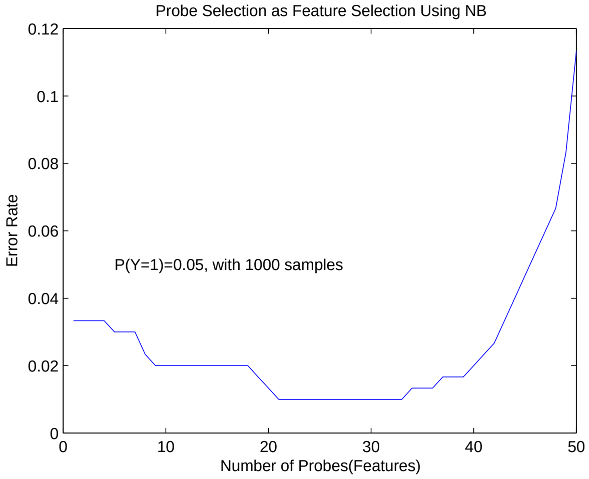

Figure 5. Probe Selection as Feature Selection

Figure 5 shows the result, where X-axis is the number of probes (i.e. features) and Y-axis is the error rate. We can use this information to better decide the number of probes necessary to localize the fault. We can also come up with the set of probes that has better accuracy. Potentially, we can even discover those hidden features that have significant impact.

5. Future Work

The primary reason which we cannot run the experiment on the real configuration data is because it is still not publicly available. We cannot obtain a statistically large set of training samples from our lab environment because its configuration doesn't change too often and thus it cannot provide a meaningful result to support our arguments. Our top priority is to start collecting real configuration data from the field (e.g. from IBM's data center).

In this paper, we focus on solving the configuration problem by using machine learning approaches. However, we believe a combination of learning-based approaches and traditional deterministic approaches may get better results in practice. For example, we can run pre-defined diagnosis scripts for those well-known problems first in a multi-phase design. We then use machine learning approach to capture those hidden relationship in the data, or encode rules that are difficult to define. How do we combine different approaches together is one of our future work.

In addition, many previous studies have shown that an integrated fault diagnosis environment (e.g. knowledge database, GUI) is an essential component to improve the overall diagnosis experience. The goal is to implicitly pull administrators into the decision process. In particular, machine learning technique (e.g. data clustering) can play a role during the query on knowledge database.

There are still a lot of open research problems in the area of failure diagnosis, such as root-cause analysis for multi-fault in a multi-layer environment. More importantly, we still don't have a truly automated tool that can simplify administrator's jobs in practice, which motivates us to continue this work.

6. Conclusion

In this project, we review a wide range of recent papers in the area of failure diagnosis. We focus on machine learning approach and provide a concrete example of probe selection by the simulation experiment. We also highlight the future research work that may distinguish ourselves from the previous work. We will continue this project since this is a real challenging problem in both industry and research community.

7. Acknowledgments

We wish to thank Sandeep Gopisetty, Prasenjit Sarkar and Kaladhar Voruganti from IBM Almaden and Andrew Ng from Stanford for inspiring this research. We would also like to thank Eric Butler, Roberto Pineiro and Ramani Routray from IBM Almaden for their helpful comments.

References

[1] M. K. Aguilera, P. Reynolds, and A. Muthitacharoen. Performance debugging for distributed system of black boxes. In *Proc. 19th ACM SOSP*, October 2003.

[2] A. Beygelzimer, M. Brodie, S. Ma, and I. Rish. Test-based diagnosis: Tree and matrix representations. In *Proc. 9th IFIP/IEEE IM*, May 2005.

[3] P. Bodik, G. Friedman, L. Biewald, H. Levine, G. Candea, K. Patel, G. Tolle, J. Hui, A. Fox, J. Michael, and D. Patterson. Combining visualization and statistical analysis to improve operator confidence and efficiency
for failure detection and localization. In *Proc. 2nd IEEE ICAC*, June 2005.

[4] M. Chen, E. Kiciman, E. Fratkin, A. Fox, and E. Brewer. Pinpoint: Problem determination in large, dynamic internet services. In *DSN*, June 2002.

[5] M. Chen, A. X. Zheng, J. Lloyd, M. Jordan, and E. Brewer. Failure diagnosis using decision tree. In *Proc. 1st IEEE ICAC*, May 2004.

[6] I. Cohen, M. Goldszmidt, T. Kelly, J. Symons, and J. S. Chase. Correlating instrument data to system states: A building block for automated diagnosis and control. In *Proc. 6th USENIX OSDI*, December 2004.

[7] I. Cohen, Z. Steve, M. Goldszmidt, J. Symons, T. Kelly, and A. Fox. Capturing, indexing, clustering, and retrieving system. In *Proc. 20th ACM SOSP*, October 2005.

[8] K. El-Arini and K. Killourhy. Bayesian detection of router configuration anomalies. In *Proc. ACM SIGCOMM Workshop on Mining Network Data*, August 2005.

[9] A. Ganapathi, Y.-M. Wang, N. Lao, and J.-R. Wen. Why pcs are fragile and what we can do about it: A study of windows registry problems. In *DSN*, June 2004.

[10] G. Hamerly and C. Elkan. Bayesian approaches to failure prediction for disk drives. In *Proc. 18th ICML*, June 2001.

[11] IBM. Tivoli productivity center. http://www-03.ibm.com/servers/storage/software/center/index.html.

[12] E. Kiciman and Y.-M. Wang. Discovering correctness constraints for self-management of system configuration. In *Proc. 1st IEEE ICAC*, May 2004.

[13] K. Nagaraja, F. Oliveria, R. Bianchini, R. P. Martin, and T. D. Nguyen. Understanding and deailing with operator mistakes in internet services. In *Proc. 6th USENIX OSDI*, December 2004.

[14] I. Rish, M. Brodie, and N. Odintsova. Real-time problem determination in distributed system using active probing. In *Proc. 9th IEEE/IFIP NOMS*, April 2004.

[15] A. Singh, M. Korupolu, and K. Voruganti. Zodiac: Efficient impact analysis for storage area networks. In *Proc. 4th USENIX FAST*, December 2005.

[16] M. Steinder and A. S. Sethi. End-to-end service failure diagnosis using belief networks. In *Proc. 8th IEEE/IFIP NOMS*, April 2002.

[17] M. Steinder and A. S. Sethi. Non-deterministic event-driven fault diagnosis through incremental hypothesis updating. In *Proc. 8th IFIP/IEEE IM*, May 2003.

[18] M. Steinder and A. S. Sethi. A survey of fault localization techniques in computer networks. *Science of Computer Programming*, 53, 2004.

[19] I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6, March 2005.

[20] P.-N. Tan, S. Michael, and K. Vipin. *Introduction to Data Mining*. Addison Wesley, 2006.

[21] H. J. Wang, J. C. Platt, Y. Chen, R. Zhang, and Y.-M. Wang. Automatic misconfiguration troubleshooting with peerpressure. In *Proc. 6th USENIX OSDI*, December 2004.

[22] Y.-M. Wang, C. Verbowski, J. Dunagan, Y. Chen, H. J. Wang, C. Yuan, and Z. Zhang. Strider: A black-box, state-based approach to change and configuration management and support. In *Proc. 17th USENIX LISA*, Oct 2003.

[23] A. Whitaker, R. S. Cox, and S. D. Gribble. Configuration debugging as search: Finding the needle in the haystack. In *Proc. 6th USENIX OSDI*, December 2004.

[24] S. Zhang, I. Cohen, M. Goldszmidt, J. Symons, and A. Fox. Ensembles of models for automated diagnosis of system performance problems. In *DSN*, June 2005.