Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing

A Survey on Failure Prediction of Large-Scale Server Clusters

Zhenghua Xue, Xiaoshe Dong, Siyuan Ma, Weiqing Dong
Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
E-mail: zhxue@stu.xjtu.edu.cn

Abstract

As the size and complexity of cluster systems grows, failure rates accelerate dramatically. To reduce the disaster caused by failures, it is desirable to identify the potential failures ahead of their occurrence. In this paper, we survey the state of the art in failure prediction of cluster systems. The characteristic of failures in cluster systems are addressed, and some statistic results are shown. We explore the ways of the collection and preprocessing of data for failure prediction, and suggest a procedure for preprocessing the records in automatically generated log files. Focused on the main idea of five prediction methods, including statistic based threshold, time series analysis, rule-based classification, Bayesian network models and semi-Markov process models, are analyzed respectively. In addition, concerning the accuracy and practicality, we present five metrics for evaluating the failure prediction techniques and compare the five techniques with the five metrics.

1. Introduction

With the increase of the computational demand, high performance clusters, as an enormous computational power, are gaining popularity for their flexibility, high scalability and excellent cost-performance ratio, and they have been an alternative to the traditional supercomputers [1]. Computing capacity of clusters has increased dramatically in the past decades. However, a linear increase of cluster size results in an exponential failure rate. The transient hardware errors are increasing. System software and applications running on cluster systems is becoming more and more complex, which makes them prone to bugs and other software failures [2]. To lower the risk of failures and keep the failure rates at an acceptable level, fault tolerance techniques based on redundancy have been developed. However, compared to providing redundancy after the occurrence of failures, it is preferable that one could anticipate failures just ahead of their occurrence and perform proactive counter-measures. Predictive failure analysis will provide an opportunity to gracefully handle failures before potential outages occur.

The remainder of this paper is organized as follows. Section 2 addresses the characteristic of failures in clusters. In section 3, the collection and preprocessing of data is presented. Section 4 represents five failure prediction techniques. Several metrics on evaluating the techniques are proposed in section 5. Finally, we summarize our conclusion.

2. Characteristic of failures in clusters

2.1. Classification and statistic of root faults

Classification of fault can provide useful guidance, in particular with regard to planning means of fault prevention, tolerance, diagnosis, and removal, based on the designer's assumptions about likely faults [3]. In general, the root fault of clusters can be classified into six categories [4]: hardware, software, network, environment, human and unknown. Many studies have performed a statistic on root fault of clusters, the results differ in the percentage of various fault type. However, they agree on the fact that the hardware fault and software fault mainly contribute to the failure of clusters. Based on a public release of a large set of failure data [5], we conducted a statistic on root faults of several clusters, and the results are shown in Figure 1. Figure 2 illustrates the statistic of nodes in a cluster where the "Facilities" denotes environment faults.

The results show that hardware account for the faults. Among hardware faults, CPU, disk and memory mainly contribute to the faults. For example, in cluster

This work is supported by the China Ministry of Science and Technology 863 Program (Grant No. 2004AA111110 *and* 2006AA01A109)

0-7695-2909-7/07 $25.00 © 2007 IEEE
DOI 10.1109/SNPD.2007.284

IEEE computer society

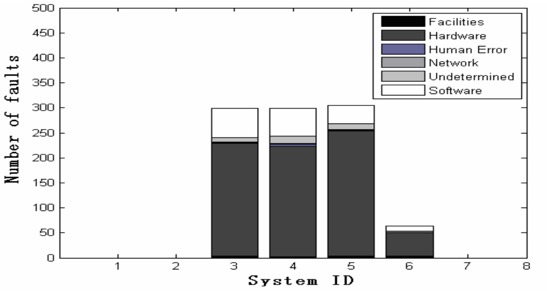

Figure 1. Statistic on root faults of several clusters

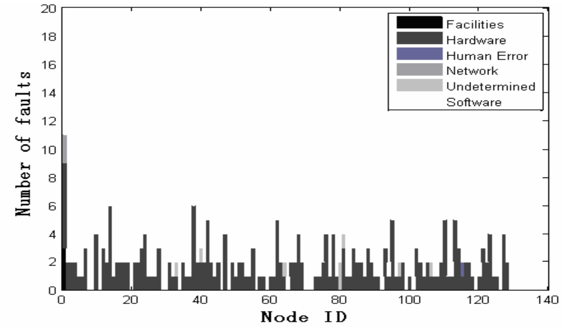

Figure 2. Statistic on root faults of cluster 3

3, the percentages of the three types of faults are about 79%, 6%, and 4% respectively. The high percentage of CPU faults may owe to the fact that cluster 3 is mainly for computing, which indicates the failure of clusters is related to the workload.

2.2. Statistical model of time between failures

Mean Time Between Failures (MTBF) is one of the most important metrics evaluating the availability of cluster systems, and statistical model of time between failures may be used to predict the arrival of failures in clusters, though it is not an effective method concerning the long duration to produce the statistical model. Several studies [6, 7] analyzed the time between failures (TBF), and the results showed the Weibull distribution is a good fit. We made an analysis on the TBF of several clusters whose failure data is provided by [5], and fit the data to several theoretical distributions, including the exponential, gamma, Weibull and lognormal . Figure 3 and figure 4 demonstrate the results. The statistic results show, in addition to Weibull distribution, gamma and exponential distribution seem good fits for the TBF of both a cluster and a node.

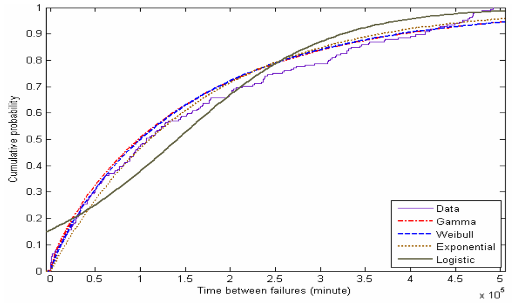

Figure 3. Empirical CDF for inter-arrival Time of failures in cluster 3

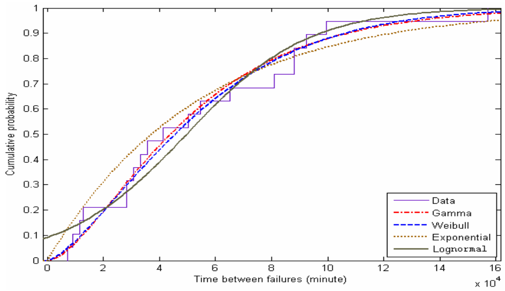

Figure 4. Empirical CDF for inter-arrival time of failures at node 1 in cluster 3

3. Collection and preprocessing of data related to failures

3.1. Collection of data related to failures

The information about the system can be generalized into two basic categories [8]: event logs, and system active reports (SAR) data. The SAR and event log data provide the temporal status of system health and environment. The SAR data can be obtained from API of operating system, and the event logs can be acquired from the automated event-logging mechanisms or from the manual logs maintained by administrators [4]. The automated error logs are often generated by the underlying operating system and applications running on the system.

3.2. Preprocessing of data related to failures

Predefined by the designers for system analysis, the SAR data and manual log can be easily processed. However, the raw data in an automatically generated log files are difficult to deal with for some reason. For example, different cluster systems usually have different log file formats; A single event may be recorded repetitively; Most log records contain messages in natural language, it is impossible to extract key knowledge without semantic analysis. To address the issues above, some preprocessing measures have been taken and a guideline on creating well structured log files has been proposed in [9]. Based on the current methods [9, 10, 11] on data preprocessing for failure prediction, we suggest the following procedure for preprocessing the automatically generated records:

- Defining a well structured record format to facilitate the paring of records.
- Filtering the records to decrease the data to be analyzed by removing some repetitive events in a specified time window.
- Extracting the essential information from the filtered data set, and reorganizing the information in a predefined well structured format.
- Classifying the reorganized records to facilitate the data analysis.

After this procedure, some concise, comprehensive and well organized error event sets may be achieved, and the accuracy of failure prediction will be improved by the refined data.

4. Techniques of failure prediction

Techniques of failure prediction have been developed in recent years, and many predicting techniques have been proposed. This section shows the methodology of failure prediction in cluster systems.

4.1. Statistic based threshold

This technique is based on the detection of large error clusters, i.e. a threshold numbers of errors might be precursors of an imminent crash. This method has been adopted by [12] to predict the failures in clusters. The following variables are used to describe this method:

Tcrash: Time of the system crash

Tstart: Time at which system restarted

Terror: Time at which the error occurred

A variable *K*, to be associated with each error record, is defined as follows:

K = (Terror - Tstart) / (Tcrash - Tstart) (1)

A frequency chart for each type of error versus K will be produced by computing K for each error of the specific type. A 6-month period measurement [12] for two types of errors demonstrated that the error generation rates prior to the occurrence of the crash started to gradually increase when K was around 0.75, and 0.75 may be set as a threshold indicating the upcoming failures. In addition, analysis of system utilization prior to a crash was performed, and the results suggested that the probability of failures drastically increased when CPU utilization was over 75%. This indicates that high utilization means the high probability of system failures.

4.2. Time series analysis

Time series analysis can be used to forecast the failures dealing with the exhaustion of system resources by predicting time continuous variables such as CPU utilization, memory usage, and network traffic, etc. Several time series models have been exploited to predict the failures, including *MEAN*, *LAST*, *BM(p)*, *AR(p)*, *MA(q)*, *ARMA(p,q)*.

Measurements for the above models were conducted, and comparisons on the performance of prediction were presented in [8]. Among the models above, ARMA is the most widely used model for prediction, and many studies [13, 14] have leveraged this model to predict time continuous variables of cluster systems. In general, ARMA modeling proceeds by the following steps.

- **Identifying the model.** Identification consists of specifying the appropriate structure (AR, MA or ARMA) and order of model. Identification may be done either by looking at plots of the autocorrelation function and partial autocorrelation function or by an automated iterative procedure, i.e. fitting many different possible model structures and orders and using a goodness-of-fit statistic to select the best model.
- **Estimating the coefficients.** Coefficients of AR models can be estimated by least-squares regression. Estimation of parameters of MA and ARMA models usually requires a complicated iteration procedure.
- **Verifying the model.** The verification includes ensuring the residuals of the model are random and the estimated parameters are statistically significant.

4.3. Rule-based classification

Different from time series analysis for predicting the time continuous variables, the rule-based classification is used to predict the rare events. The main idea behind this method is as follows:

- **Finding all event types which frequently preceding the target events within a fixed time window.** On every occurrence of a target event, all event types within the window are stored, and all event sets above a minimum user-defined confidence will be found by an association-rule algorithm.
- **Validating event sets that uniquely characterize target events.** In addition to computing the confidence of the frequent event sets, the validation phase ensures the probability of an event set *Z* appearing before target events is significantly larger

than the probability of Z not appearing before target events, which discards any negative correlation between Z and target events and decreases the number of candidate patterns used to build a rule-based model for prediction.

- **Combining validated event sets to build a probabilistic rule-based system for prediction.**

According to the results performed by [8], this technique could predict the critical rare event with up to 70% accuracy based on the associative data mining rule within a specified time window. By including the warning window parameter into the analysis, rule based classification results were further improved in terms of prediction accuracy.

4.4. Bayesian network models

Bayesian networks (BN) are widely used for knowledge representation and reasoning under uncertainty. In a general form, the structure of a BN is a directed acyclic graph where nodes correspond to random variables of interest and directed arcs represent direct causal or influential relation between nodes. The uncertainty of the interdependence of the variables is represented by the conditional probability table $Pr(x_i / \pi_i)$ associated with each node x_i, where π_i is the parent set of x_i. The graphical structure of BN allows an unambiguous representation of interdependency between variables. This, together with the independence assumption, leads to one of the most important features of BN: the joint probability distribution of $X = (x_i, \ldots, x_n)$ can be factored out as a product of the conditional distributions in the network,

$$Pr(X = x) = \prod_{i=1}^{n} Pr(x_i / \pi_i) \qquad (2)$$

The strategy of the Bayesian network models applied in predicting the failures is as follows:

- **Defining some variables from the filtered error logs and SAR logs.** These variables may include event timestamp, event ID, event class, as well as event severity etc., as depicted in [8].
- **Learning the graphic structure (structure learning) whose nodes are the defined variables and estimating the probabilities (parametric learning) from data.** The learning algorithm consists of two parts [15]: A quality measure which is used for computing the quality of the candidate BNs and a search algorithm which is used to efficiently search the space of possible BNs to find the one with highest quality.
- **Predicting events.** Once a model describing the relationships among the set of variables has been selected, the BN can be used to predict the probabilities of certain variables denoting the failure events by the observation of their parent set.

Predicting failure probability of clusters has been conducted in [8], the results showed that BN can be effectively used to determine the statistical relationships among the variables of a node or a number of nodes in a cluster.

4.5. Semi-Markov process models

Semi-Markov process (SMP) models have been widely used to predict failures. In this model, the transition from state i to state j only relies on the state i and sojourn time in state i. SMP is determined by three probabilities: s_i , p_{ij} and $D_{ij}(t)$. s_i is the initial probability of state i , p_{ij} denotes the transition probability from state i to state j, and the probability distribution $D_{ij}(t)$ represents the duration of the transition from state i to state j . Generally, the training of SMP model includes three steps:

- **Constructing the state sets of SMP.** Some error events preceding the occurrence of failures are recorded by historical logs, and similar error events are grouped by a clustering algorithm to form the states of the SMP model.
- **Defining transition duration $D_{ij}(t)$ and estimating transition probabilities p_{ij} .** The $D_{ij}(t)$ is procured from the statistic of the information provided by log files, and p_{ij} is determined by the ratio of the transition times from state i to state j to the ones from state i to all its next states
- **Computing $P_F(t)$ A predicted failure warning will happen if $P_F(t)$ exceeds a threshold.** Computation of $P_F(t)$ follows the formula (3) [16]:

$$P_F(t) = \sum_i F_{iF}(t)\pi_i \qquad (3)$$

Where π_i is the estimation of state probabilities and

$F_{iF}(t)$ denotes the first passage time distribution to the failure state given that the process is in state i at present time.

In [16, 17], Similar Events Prediction (SEP) method based on SMP was applied to predict the failures of software, the measured results showed SEP could predict failures of software with high accuracy. In [18, 19], SMP was adopted to estimate resource exhaustion rates of operating system and time to exhaustion for the resource. Ren, Lee, and Eigenmann etc. [19] demonstrated that the prediction achieved accuracy above 86% on average and outperforms linear time series models, while the computational cost is negligible.

5. Metrics of Evaluation on techniques of failure prediction

Before discussing the metrics of evaluation on the techniques of failure prediction, we will introduce a few terms often used to describe the metrics of prediction techniques:

- *true positives*: the number of correctly predicted failures within a specified duration.
- *false negatives*: the number of failures happened but without being predicted within a specified duration
- *false positives*: the number of failures not happened but predicted mistakenly within a specified duration.

To assess the techniques of failure prediction, the following metrics has been proposed by [16, 17]:

- *Precision*: the ratio between the numbers of correctly identified failures and predicted failures. It can be depicted by the defined variables:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \quad (4)$$

- *Sensitivity*: the ratio between the number of correctly identified failures and the sum of the true positive and false negative. It can be presented as:

$$Sensitivity = \frac{true\ positives}{true\ positives + false\ negatives} \quad (5)$$

- *Tension*: a balanced relation between the Precision and the Sensitivity. There is trade-off between Precision and Sensitivity. For example, improving the Sensitivity, i.e. reducing the number of false negative, mostly also decreases the number of true positives, i.e. reducing precision. Tension can be represented as follows:

$$Tension = \frac{2 \cdot Sensitivit\ y \cdot Precision}{Sensitivit\ y + Precision} \quad (6)$$

The three metrics above concentrate on the accuracy of failure prediction. However, as far as practicality is concerned, we consider the following metrics should be evaluated:

- *Complexity*: the complexity in implementing the predictor. It is a comprehensive metric reflecting the complexity on both modeling and training of failure prediction. For example, in the Bayesian network model, with the increase of the variables, the candidate models exponentially increase, which increases complexity of the structure learning as well as the parametric learning.

- *Sample amount:* the average amount of samples required to identify the statistical trend or build a model for prediction.

Based on the analysis on the principle of the predicting techniques pre-described, combined with some measured results presented in some literatures [7, 15, 19], we make a coarse grain comparison about the techniques of failure prediction in table 1. The results demonstrate the statistical estimation based threshold prediction technique has low accuracy, but low complexity, due to lack of strict mathematical analysis, while with support of some mathematical theories, several techniques gain high accuracy with high complexity.

6. Conclusion

Table 1 Evaluation on techniques of failure prediction

Technique	Precision	Sensitivity	Tension	Complexity	Sample amount
Statistic based threshold	low	low	low	low	large
Time series analysis	medium	medium	medium	low	small
Rule-based classification	medium	medium	medium	medium	medium
BN	high	high	high	high	medium
SMP	high	high	high	medium	medium

In this paper, we focus on surveying the state of the art in failure prediction of cluster systems. The classification and statistic of root faults are addressed, and statistical models of time between failures are analyzed based on a public release of a large set of failure data, the results show, in addition to Weibull distribution, gamma and exponential distribution seem good fits for the TBF of both a cluster and a node. We explore the ways of the collection and preprocessing of data providing for failure prediction, and suggest a procedure for preprocessing the records in automatically generated log files. By the proposed procedure, some refined data may be achieved to facilitate the failure prediction. We represent five failure prediction techniques, including statistic based threshold, time series analysis, rule-based classification, BN models and SMP models. The essential of these techniques are analyzed, and the applied scenarios are pointed out, e.g. time series analysis is used to predict the time continuous system variables, while the rule-based classification is adopted for predicting the rare events. Finally, concerning the accuracy and practicality, we present five metrics for evaluating the failure prediction techniques and bring the five techniques into comparison.

References

[1] Z. Xue, X. Dong, and W. Wu, "AOCMS: An Adaptive and Scalable Monitoring System for Large-Scale Clusters", In *Proc. of IEEE Asia-Pacific conference on services computing*, Dec. 2006.

[2] K. Vaidyanathan, R. E. Harper, S. W. Hunter, and K. S. Trivedi, "Analysis and Implementation of Software Rejuvenation in Cluster Systems", In *Proc. of ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, 2001.

[3] B. Rasndell, "On Failures and Faults", In *Formal Methods (FME)*, vol. 2805 of LNCS, 2003, pp. 18-39.

[4] B. Schroeder, G. Gibson, "A Large Scale Study of Failures in High-performance Computing Systems", In *proc. Of International Conference on Dependable Systems and Networks* , June 2006.

[5] The raw data is available at the following two URLs: http://www.pdl.cmu.edu/FailureData/ and http://www.lanl.gov/projects/computerscience/data/, 2006.

[6] T. Heath, R. P.Martin, and T. D. Nguyen. "Improving Cluster Availability Using Workstation Validation", In *Proc. of ACM SIGMETRICS*, 2002.

[7] D. Nurmi, J. Brevik, and R. Wolski, "Modeling Machine Availability in Enterprise and Wide-area Distributed Computing Environments", In *proc. of EuroPar'05*, Aug. 2005.

[8] R. Sahoo, A. Oliner, I. Rish, M. Gupta, J. Moreira, S. Ma, R. Vilalta, A. Sivasubramaniam, "Critical Event Prediction for Proactive Management in Large-scale Computer Clusters", In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2003.

[9] F. Salfner, S. Tschirpke, and M. Malek, "Comprehensive Logfiles for Autonomic Systems", In *Proc. of 9th IEEE Workshop on Fault-Tolerant Parallel Distributed and Network-Centric Systems*, 2004

[10] Y. Liang, Y. Zhang, A. Sivasubramaniam, R. Sahoo, and J. Moreira, "BlueGene/L Failure Analysis and Prediction Models", In *Proc. of the 2006 International Conference on Dependable Systems and Networks* , June 2006.

[11] Y. Liang, Y. Zhang, A. Sivasubramaniam, R. Sahoo, J. Moreira, and M. Gupta, "Filtering Failure Logs for a BlueGene/L Prototype", In *Proc. of the International Conference on Dependable Systems and Networks*, June 2005.

[12] F. A. Nassar, D. M. Andrews, "A Methodology for Analysis of Failure Prediction Data", *CRC Technical Report No. 85-20*, Stanford University, Polo Alto, California, 1985.

[13] W. Yang, Q. Zhu, P. Li, and P. Qian, "Server Load Prediction Based on Time Series", *Journal of Computer Engineering*, vol. 32, no. 19, 2006, pp. 143-148.

[14] B. Zou, Q. Liu, "ARMA-Based Traffic Prediction and Overload Detection of Network", *Journal of Computer Research and Development*, vol. 39, no. 12, 2004, pp. 1645-1652.

[15] E. Castillo, J. M. Guti´errez, and A. S. Hadi, "Expert Systems and Probabilistic Network Models", *Springer-Verlag*, New York, 1997.

[16] F. Salfner, M. Schieschke, and M. Malek, "Predicting Failures of Computer Systems: A Case Study for a Telecommunication System", In *Proc. of the 20th International Parallel and Distributed Processing Symposium*, April 2006.

[17] G. A. Hoffmann, F. Salfner, and M. Malek, "Advanced Failure Prediction in Complex Software Systems", In *Symposium on Reliable Distributed. Systems*, Oct. 2004.

[18] K. Trivedi and K. Vaidyanathan, "A Measurement-based Model for Estimation of Resource Exhaustion in Operational Software Systems", In *Proc. of the 10th International Symposium on Software Reliability Engineering*, Nov. 1999.

[19] X. Ren, S. Lee, R. Eigenmann, and S. Bagchi, "Resource Failure Prediction in Fine-Grained CycleSharing Systems", In *Proc. of the 15th IEEE International Symposium on High Performance Distributed Computing*, June 2006.