2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application

A Survey on Failure Prediction in Large-scale Computing Systems

Fei Xia
State Grid Jiangsu Electric Power Co. Ltd. Information & Telecommunication Branch
Nanjing, China
13951852901@163.com

Hu Song
State Grid Jiangsu Electric Power Co. Ltd. Information & Telecommunication Branch
Nanjing, China
jim_song@126.com

Long-Chuan Yan
State Grid Information & Telecommunication Branch
Beijing, China
lchyan@sgcc.com.cn

Yan Li
State Grid Information & Telecommunication Branch
Beijing, China
liyanllyy@126.com

Li-Jun Wang
State Grid Electric Power Research Institute
Nanjing, China
wanglijun@sgepri.sgcc.com.cn

***Abstract*—With the development of many data-intensive applications, large-scale systems have been widely used to solve advanced computation problems. As the increasing growth of complexity and scale, these systems are more likely to confront failure. Since remedial measures for failure take huge cost and effort at today's large-scale systems, fault tolerance which aims to decrease the impacts of fault on systems has become a necessity instead of an option. As one of the key techniques of fault tolerance, failure prediction has made itself an increasing important issue to improve the resource efficiency and the availability of systems. Over the past few years, a multitude of innovative failure prediction approaches has emerged such as mathematical and statistical modeling, machine learning techniques and so forth. Unfortunately, they are currently so poorly classified that it is difficult to figure out the wide spectrum of methods concerning with this area. To this end, we provide an extensive and comprehensive survey of existing research work in the area of failure prediction via exploring and analyzing over 20 various approaches. Also, we develop our own taxonomy assisting in classifying methods, which makes it easier to understand and compare the pros and cons of these methods in respective category.**

***Index Terms*—failure prediction, large-scale systems, fault tolerance,fault diagnosis**

I. Introduction

Large-scale systems have been extensively applied in cloud computing, data centers and high-performance computing. With the ever-growing demand in science and engineering, cluster systems tend to consist of a large number of nodes. Such a scale along with the increasing system complexity makes them more prone to failures, which wastes a lot of resources and degrades the system efficiency. As Bianca et al. [1] have surveyed, the average failure rate of some computing systems designed for high-performance computing is comparatively high and the average restoration time is relatively long.

Although there have been various methods making effort in designing ultra-reliable components, scale and complexity of systems still prevail over their improvement, introducing challenges on failure management. Failure management which can simplify system management and improve system availability is mainly composed of two parts: failure prediction and failure avoidance. Failure prediction is used to predict whether the failure is going to happen while failure avoidance takes active action like checkpoint or job migration by making use of the predicted result of failure predictor. Being the prerequisite of failure tolerance, failure prediction is of vital importance.

Considerable research has been done on failure predication with so many techniques under different failure scenarios that there is difficulty in figuring out and classifying advanced research findings. Consequently, in this paper, we provide a comprehensive survey of failure prediction in large-scale systems and develop our own taxonomy aiding in method categorization based on [2].

Li-Jun Wang is the corresponding author.

II. Related Terminology

In general, failure prediction can usually be divided into two types in large-scale systems: offline failure prediction and online failure prediction. Offline failure prediction serves as an analysis tool to analyze the reliability of systems, delivering theoretical results or hypothetical scenarios. It is typically useful to predict average behaviour in the long run such as predicting the lifetime and failure rates and so forth, therefore also called reliability analysis. Differently, online failure prediction refers to prediction which focuses on whether or

978-1-6654-9457-1/21/$31.00 ©2021 IEEE

DOI 10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00303

not the failure will take place in the short run. It is made at runtime with regard to the current state of the system.

When it comes to online failure prediction, it is of significant importance to figure out some basic related concepts: error, fault, and failure. In this paper, we use the definition from [2] dating back to [3]. Failure is defined as misbehavior observed by either a human or a computing system. Error occurs when state of system is not consistent with normal status, which are classified into undetected error and detected error based on whether or not the error is caught by the error detector. Fault is the root cause of error. Their propagation relation is shown in Fig. 1 based on [2]. The activation of the fault might lead to error, which also causes an incorrect state of the system as a side effect, namely symptom. The efficient way to recognize symptom is to monitor system parameters and find the abnormality. Both undetected error and detected error might bring about failure, and the only difference is that detected error is recorded in the system logs while undetected error is not.

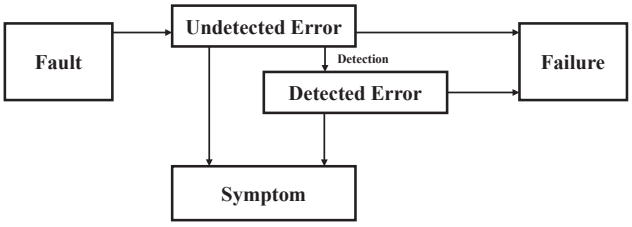

Fig. 1. Propagation Relation

In this paper, We have surveyed more than 20 failure prediction techniques and the taxonomy we propose is in Fig. 2. We generally divide all methods in a time horizon, in other words, online techniques applied at runtime and offline techniques serving as an analysis tool. In detail, for offline techniques, we divide them into 4 categories based on their focus: IC and chip, printed circuit board(PCB), interconnect failure and network, software reliability. For online techniques, we can step further from two aspects since failure can be predicted by symptoms or the logs in which detected error is recorded. In the consequence, we can observe the symptoms via monitoring key characteristics of the system or make full use of logs to predict failure.

III. Offline Failure Prediction

Offline failure prediction refers to the methods that seldom take the actual state of a system into account and are usually applied regarding with precautionary maintenance and reliability theory.

A. IC and chip failures

Vanessa Smet et al. [4] provide an experimental study on aging procedures, failure modes, and lifetime estimation of insulated-gate bipolar transistor (IGBT) power modules stressed under power-cycling conditions. They age 19 IGBT

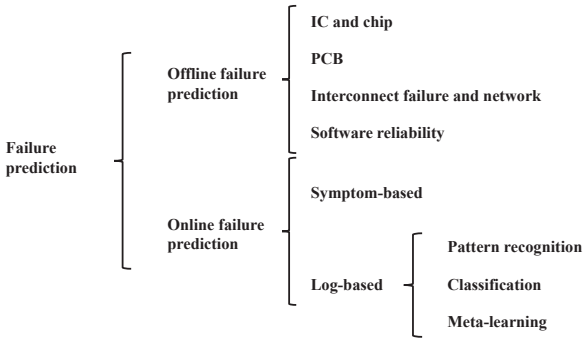

Fig. 2. Taxonomy

modules with five different power cycle protocols for testing. The related aging signals can be monitored frequently via the monitoring system during test. At the end of the test, they analyze acoustic scanning and perform scanning electron microscopy(SEM) imaging. If there is any damage in these modules, it will be listed. Experimental results show that the aging mechanism mainly involves wire bonding and emitter metallization, and the degree of aging depends on the severity of the protocol.

Due to the development of technology, the silicon processing on the chip has approached the atomic scale. It is predicted that the future chip design will include hundreds of billions of transistors, of which more than 10% are defective due to wear-out or diversity. Therefore, people must have the ability of designing reliable systems from unreliable components at that time. David Fick et al. [5] propose a network-on-chip called Vicis, which is resilient to the loss of network components caused by hard faults. Each router in Vicis has built-in self-in (BIST) to diagnose the location of hardware faults, and alleviate them by running various algorithms to make full use of error correcting code(ECC), port switching and crossbar bypass bus. Also, distributed algorithms in the network are run to protect the chip network from the single point failure of the router. Experiments show that Vicis is more reliable and has fewer overhead compared with solutions based on N-modular redundancy (NMR).

The eventual aim of failure physics evaluation is to assess the performance of product under specific working conditions given that there exists constraints in design, hence modeling the reliability of product is necessary. However, there is no existing approach having the ability to predict the failure rate or wear-out lifetime of semiconductor products nowadays. Therefore, Mark White et al. [6] introduce the failure physics method for microelectronic reliability modeling and evaluation. In addition, they summarize some complex reliability modeling tools and technologies of microelectronic equipment that takes root causes of failure and physical fault behavior into consideration.

The power consumption and power density of high-

performance microprocessor have been increased greatly due to system integration and performance requirements, which is one of the main challenging problems microprocessor designers are facing. This has brought about plenty of problems, such as the increasing cost of cooling and packaging design, high power cost of server, a decline in microprocessor reliability and so forth. Working on power consumption, David Brooks et al. [7] provide a comprehensive study. On the one hand, they explain related problems of power, thermal and reliability. On the other hand, they survey latest approaches of modeling and identify the promising directions in this area.

B. PCB failures

Electromigration (EM) has become a major problem affecting the reliability of very large scale integration(VLSI). Since the current density and temperature varies over time because of many different tasks running on the system, the existing EM models and evaluation techniques which don't take this time-varying change into account is no more practical at the system level.

Haibao Chen et al. [8] propose new EM model step by step from model considering time-changing temperature, model considering time-changing current density to model considering both of these time-varying features, which reflects practical operations of the chip realistically. What's more, they develop a fast approach to calculate stress for the nucleation phase after investigating how the changing current density and temperature profile impact a wire's EM-induced lifetime. Further, they offer new formulae to calculate the changes of resistance during growth phase according to time-varying changes in temperature and current density. The results of experiment demonstrate that their proposed approach can raise efficiency greatly while being consistent with numerical analysis.

C. Interconnect failure and network

To cope with time-varying stresses, Zhijian Lu et al. [9] first analyze the influence of time and space thermal gradients on the lifetime interconnects from the aspect of EM. Then, they discuss how the temporal characteristic and spatial characteristic of temperature distributions affect the solution of interconnect lifetime prediction individually. Also, formula is developed for increasing the accuracy of dependability estimation and allowing designers to design wisely.

Increasing interconnection between critical network infrastructures necessities the understanding of cascading actions that might trigger network failures. Therefore, Tony H et al. [10] study the topological complexities of network interconnections and evaluate the potential impacts of losing vital infrastructure elements via a proposed spatial optimization model. The model considers both the topo-logical characteristics of the network and the network characteristics like network traffic compared with former models.

Motivated by the obervation that faults can be propagated through interdependent network systems, Madhav V. Marathe et al. [11] concentrate on inter-dependent crucial infrastructures and analyze cascading failures to evaluate fault tolerance and reliability. They use a synchronous dynamical system (SyDS) and bottom-up dynamic programming to solve the configuration sequence completion problem (CSC), which can be generalized to some other decision problems for dynamical systems.

D. Software reliability

There are a lot of notable research in this area. Lars Grunske et al. [12] and Katerina Goševa-Popstojanova et al. [13] both use architecture modeling language to assess software reliability and predict failure behavior, which can help analyze properties like safety and dependability. Cheung et al. [14] develop a Markov model to evaluate the dependability of entire software system according to the dependability of each individual components. Muhammad Ali Babar et al. [15] provide an evaluation framework for comparing different evaluation methods of system architecture. Brosch et al. [16] take consideration of some important reliability-relevant factors and offer an approach called PCM-REL for software architecture-based reliability prediction. Uhle et al. [17] take full advantage of dependency graphs as well as qualitative and quantitative fault trees to model the dependability in micro-service architecture applications.

IV. Online Failure Prediction

For most large-scale systems, it is essential that failure prediction is able to be performed in time, desirably even in advance [2]. As a result, online failure prediction is more preferable in that it could anticipate failures ahead of their occurrence, enabling proactive counter-measures taken in advance. The foundation of online failure prediction is monitoring and evaluating the state of current system. As we have mentioned above, the methods of online failure prediction is divided into two aspects: symptom-based methods and log-based methods. The major difference between them is that log-based methods usually handle with event-driven input data while symptom-based methods operate on periodic system-monitoring parameters in most cases. What's more, system observations are often real numerical data while events in logs are discrete and classified data such as log IDs, event IDs and so forth.

A. Symptom-based Methods

Symptoms are side effects raised by errors and symptoms-based methods attempt to identify failure through indictive symptoms. In this section, we list out several classical methods.

Plenty of methods combine time series analysis with machine learning techniques. Lei Li et al. [18] collect time series data about parameters related to system resources via /proc virtual file system and monitoring tool named procmon. And they use ARMA to build a multivariate model for forecasting resource exhaustion. However, in contrast to the regular linear regression and extended linear regression models, the proposed solution induces additional computational overhead because Autocorrelation Function (ACF) and Partial Autocorrelation

Function (PACF) of collected data need to be calculated. In addition, the methods of data collection analysis are only aimed for the Apache server, which has not been experimented in other different systems to prove generality. Ioana Giurgiu et al. [19] advance the idea of predicting uncorrectable errors in DRAM by machine learning methods within specified time frames. They use the strategy called changepoint detection to pick out sensor metrics which are likely to correlate to DRAM uncorrectable errors among all the collected time series data. And they apply these predictors as inputs of machine learning techniques to predict whether the server will have a DRAM failure or not within a certain time period. Proposed method proves to have high prediction accuracy while suffering on the recall of machines at risk. Motivated by the observations that workload characteristics, temperature, power consumption are correlated with GPU errors, Bin Nie et al. [20] pick out useful features systematically via classifying features into spatial and temporal dimensions and exploit these features for GPU failure prediction based on machine learning methods such as Logistic Regression (LR), Gradient Boosting Decision Tree (GBDT) and so forth. The authors also examine the effectiveness of their approach under diverse scenarios from plenty of aspects consisting of its accuracy, overhead, robustness, overhead and so forth, which sheds light on GPU proactive fault management.

Nevertheless, these former methods only concentrate on individual components or think the system as a whole when predicting failure, taking no account of architectural dependencies of software, hence ignoring the propagation of failures. Instead, Teerat Pitakrat et al. [21] exploit a combination of both failure predictors and architectural models to develop a hierarchical online prediction method called HORA. The main principle behind HORA is that the propagation or the outcome of failures can be predicted by Bayesian networks if the failure of each individual component and the dependencies among components in the software system can be recognized. The results of experiment show that the prediction quality of HORA is remarkably higher than the monolithic method. On top of that, Thanyalak Chalermarrewong et al. [22] propose a framework for failure prediction with architectural consideration in a cluster, using self-adjusted ARMA and fault tree analysis. For one thing, a set of fault-correlated system parameters is intermittently monitored. For another, abnormalities of these values can be indicated by the ARMA model, and are lately converted into binary values to feed into the fault tree for prediction. Also, [24] [23] introduce an online fault localization approach based on architecture, which provides a new perspective for architecture-based failure prediction.

In addition, instead of using only the directly-observable data from outside executions or the architectural information of systems, Seer proposed in [25] is a lightweight method taking advantage of fast hardware performance counters to obtain internal execution data for failure prediction. Since the inside executions data is collected via hardware performance counters rather than additional measurement instruments, the quality of prediction can be improved without inducing much extra runtime overhead.

B. Log-based Methods

Systems logs can play an important part in failure prediction because the changing states of system are recorded in detail in logs so as to track system behaviors easily. Usually, the detection error is recorded in the log. Hence, log-based methods make endeavor in logs analysis to find valuable information for failure prediction. In general, the log-based methods can be classified into three categories and we will give a detail discussion on each of them in this section including pattern recognition, classification and meta-learning.

a) Pattern Recognition

Failure patterns can be derived from error event sequences. Pattern recognition methods are designed for examining causal relations of normal events to fatal events and recognizing patterns which can imply an impending failure.

Anwesha Das et al. [26] study on Cray system and offer an unprecedented insight that node failures can be predicted via analyzing log phrases carefully from the aspect of time series. They apply the algorithm called Topics over Time [27] to learn the top N topics over a period of time, propose time-based phrase (TBP) scheme to extract patterns and form sequences of phrases called chains indicative of failure in the past. If those chains with high similarity reappear in the latest data, there is possibility that the corresponding node will fail in the future.

Yukihiro Watanabe et al. [28] present a new failure prediction model which automatically learns message patterns regardless of their format. Firstly, the proposed model groups the input message according to the similarity of message texts. Secondly, the model identifies message patterns and calculates Bayesian probabilities for every input of message pattern or failure information. Finally, occurrence of failures is predicted by detecting the message pattern in the message pattern dictionary.

Ziming Zheng et al. [29] refine the widespread metrics for assessing the accuracy of prediction and combine the location and lead time information with failure prediction. They present a failure prediction mechanism based on Genetic Algorithm (GA) for IBM Blue Gene systems. Compared with common association rule-based method, there is high possibility that GA can converge rapidly to the rules because generating numerous candidates rules is no more necessary. In the training phase, they recognize a set of non-fatal events for each fatal event within specific time window and give a possible location for each rule. Then, weights of fitness function can be set to evolve the population after collecting an initial population from manually selecting potential rules and randomly choosing rules.

In [30], Yan Yu et al. present a method to predict failure through Self-Updating Cause-and-Effect Graph (CEG). The innovation is that the method can not only dig out the causality between log events of systems, but also set up and update CEG automatically within the life cycle of systems. Their failure

prediction based on CEG in real time is composed of four modules, namely log capturing module, log parsing module, event identification module, and prediction module. The latest log event generated by cluster systems will be captured by log capturing module and then be transmitted to log parsing module. Through the processing of the first two modules, the log event will get its tag. Finally, whether any failure is going to take place in the foreseeable future can be predicated according to the tag and CEG.

b) Classification

Classification refers to categorizing a new sample into a specific category among the existing known categories. The goal of classification is to allocate a label to an input data sample to predict whether there is going to be a failure or not within a certain amount of time.

In [31], Nithin Nakka et al. introduce an approach to predicting failure via combining failure logs with usage logs to extract failure instance. Every failure instance consists of preceding and subsequent failure information relative to its own timestamp, including time of usage, system idle time, time of unavailability, time since last failure and so forth. According to this data, they apply several widely used decision tree classifiers such as RepTree, RandomTree to predict whether a failure will take place within 1 hour.

In [32], non-negative matrix factorization is exploited on Oak Ridge National Laboratory's BG/P logs to build a model. Later, new testing data is provided to the system and the model makes prediction according to the similarity between the new data and the failure data. Also, Generalized Linear Discriminant Analysis is exploited to improve the classification into fault and non-fault data.

In order to improve resources utilization, Mbarka Soualhial et al. [33] use statistical models and historical information to predict task failure. First, they extract factors that make a direct impact on the results of task or job scheduling and identify the correlation. Then, they apply several algorithms such as Boost, GLM, CTree, Random Forest and Neural Network algorithms to make predictions about whether a scheduled task will fail or not.

c) Meta-learning

Known as ensemble-learning, meta-learning integrates multiple data mining techniques and combines different individual models to improve the quality of prediction.

Jiexing Gu et al. [34] provide a dynamic meta-learning prediction engine composed of the meta-learner, the predictor, and the reviser. The meta-learner takes pre-processed clean data as input to form diverse rules for online prediction. The reviser creates a valid ruleset from the rules generated by meta-learning for failure prediction and the ruleset is dynamically changed according to the accuracy of prediction and present state of system. The predictor triggers a warning when it finds out a match in the effective ruleset. Since the the training set is dynamically increased during system operation, the training phase doesn't require to be long.

Zhiling Lan et al. [35] study further to analyze the meta-learning method in detail. To get rid of redundant information in the log, they develop a categorizer offering an exact list of RAS types and a filter making temporal and spatial compression. As for the meta-learning step, they choose three methods as their base learners consisting of association rule-based method, statistical rule-based method, and probability distribution-based method. The meta-learning enables them to find out a diversity of failure patterns with no consideration of building complicated models of underlying system. Then, relearning is triggered periodically and effective rules are adaptively extracted via tracing prediction accuracy.

V. Conclusion

With the evolution of large-scale systems, providing fault-tolerance strategies remains to be one of the most critical research topics. The ever-growing scale of applications and complex heterogeneity even bring about more challenges for large-scale systems. Under this background, failure prediction has made itself as a major area of interest and has resulted in a broad range of techniques.

There are a lot of methods related to failure prediction. Former studies on failure occurrence find relationship both in time and space, enabling basic prediction methodology like calculating probability and correlation scores. As time goes by, more complicated approaches have emerged for failure prediction like elaborated correlation analysis and tree-based methods. Then hidden markov models, neural networks and other methods have also been explored with exciting prediction outcomes. In this paper, we provide an extensive survey over 20 different methods to introduce failure prediction techniques and a taxonomy is provided to structure these methods.

Acknowledgment

This work was supported by the Science and Technology Project of State Grid Corporation of China (Research and Application on Multi-Datacenters Cooperation & Intelligent Operation and Maintenance, No.5700–202018194A-0-0-00).

References

[1] Schroeder B, Gibson G A. A large-scale study of failures in high-performance computing systems[J]. IEEE transactions on Dependable and Secure Computing, 2009, 7(4): 337-350.

[2] Salfner F, Lenk M, Malek M. A survey of online failure prediction methods[J]. ACM Computing Surveys (CSUR), 2010, 42(3): 1-42.

[3] Avizienis A, Laprie J C, Randell B, et al. Basic concepts and taxonomy of dependable and secure computing[J]. IEEE transactions on dependable and secure computing, 2004, 1(1): 11-33.

[4] Smet V, Forest F, Huselstein J J, et al. Ageing and failure modes of IGBT modules in high-temperature power cycling[J]. IEEE transactions on industrial electronics, 2011, 58(10): 4931-4941.

[5] Fick D, DeOrio A, Hu J, et al. Vicis: A reliable network for unreliable silicon[C]//Proceedings of the 46th Annual Design Automation Conference. 2009: 812-817.

[6] White M. Microelectronics reliability: Physics-of-failure based modeling and lifetime evaluation[R]. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2008., 2008.

[7] Brooks D, Dick R P, Joseph R, et al. Power, thermal, and reliability modeling in nanometer-scale microprocessors[J]. Ieee Micro, 2007, 27(3): 49-62.

[8] Chen H B, Tan S X D, Huang X, et al. New electromigration modeling and analysis considering time-varying temperature and current densities[C]//The 20th Asia and South Pacific Design Automation Conference. IEEE, 2015: 352-357.

[9] Lu Z, Huang W, Stan M R, et al. Interconnect lifetime prediction for reliability-aware systems[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2007, 15(2): 159-172.

[10] Grubesic T H, Murray A T. Vital nodes, interconnected infrastructures, and the geographies of network survivability[J]. Annals of the Association of American Geographers, 2006, 96(1): 64-83.

[11] Marathe M V, Ravi S S, Rosenkrantz D J, et al. Computational aspects of fault location and resilience problems for interdependent infrastructure networks[C]//International Conference on Complex Networks and their Applications. Springer, Cham, 2018: 879-890.

[12] Grunske L, Han J. A comparative study into architecture-based safety evaluation methodologies using AADL's error annex and failure propagation models[C]//2008 11th IEEE High Assurance Systems Engineering Symposium. IEEE, 2008: 283-292.

[13] Goševa-Popstojanova K, Trivedi K S. Architecture-based approach to reliability assessment of software systems[J]. Performance Evaluation, 2001, 45(2-3): 179-204.

[14] Cheung R C. A user-oriented software reliability model[J]. IEEE transactions on Software Engineering, 1980 (2): 118-125.

[15] Babar M A, Gorton I. Comparison of scenario-based software architecture evaluation methods[C]//11th Asia-Pacific Software Engineering Conference. IEEE, 2004: 600-607.

[16] Brosch F. Integrated software architecture-based reliability prediction for IT systems[M]. KIT Scientific Publishing, 2012.

[17] Uhle J, Tröger P. On dependability modeling in a deployed microservice architecture[J]. Operating Systems and Middleware Group, potsdam, 2014.

[18] Li L, Vaidyanathan K, Trivedi K S. An approach for estimation of software aging in a web server[C]//Proceedings International Symposium on Empirical Software Engineering. IEEE, 2002: 91-100.

[19] Giurgiu I, Szabo J, Wiesmann D, et al. Predicting DRAM reliability in the field with machine learning[C]//Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Industrial Track. 2017: 15-21.

[20] Nie B, Xue J, Gupta S, et al. Machine learning models for GPU error prediction in a large scale HPC system[C]//2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, 2018: 95-106.

[21] Pitakrat T, Okanovic D, Van Hoorn A, et al. An architecture-aware approach to hierarchical online failure prediction[C]//2016 12th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA). IEEE, 2016: 60-69.

[22] Chalermarrewong T, Achalakul T, See S C W. Failure prediction of data centers using time series and fault tree analysis[C]//2012 IEEE 18th International Conference on Parallel and Distributed Systems. IEEE, 2012: 794-799.

[23] Casanova P, Garlan D, Schmerl B, et al. Diagnosing architectural run-time failures[C]//2013 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). IEEE, 2013: 103-112.

[24] Casanova P, Schmerl B, Garlan D, et al. Architecture-based run-time fault diagnosis[C]//European Conference on Software Architecture. Springer, Berlin, Heidelberg, 2011: 261-277.

[25] Ozcelik B, Yilmaz C. Seer: a lightweight online failure prediction approach[J]. IEEE Transactions on Software Engineering, 2015, 42(1): 26-46.

[26] Das A, Mueller F, Hargrove P, et al. Doomsday: Predicting which node will fail when on supercomputers[C]//SC18: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2018: 108-121.

[27] Wang X, McCallum A. Topics over time: a non-markov continuous-time model of topical trends[C]//Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006: 424-433.

[28] Watanabe Y, Otsuka H, Sonoda M, et al. Online failure prediction in cloud datacenters by real-time message pattern learning[C]//4th IEEE International Conference on Cloud Computing Technology and Science Proceedings. IEEE, 2012: 504-511.

[29] Zheng Z, Lan Z, Gupta R, et al. A practical failure prediction with location and lead time for blue gene/p[C]//2010 International Conference on Dependable Systems and Networks Workshops (DSN-W). IEEE, 2010: 15-22.

[30] Yu Y, Chen H. An approach to failure prediction in cluster by self-updating cause-and-effect graph[C]//International Conference on Cloud Computing. Springer, Cham, 2019: 114-129.

[31] Nakka N, Agrawal A, Choudhary A. Predicting node failure in high performance computing systems from failure and usage logs[C]//2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum. IEEE, 2011: 1557-1566.

[32] Thompson J, Dreisigmeyer D W, Jones T, et al. Accurate fault prediction of BlueGene/P RAS logs via geometric reduction[C]//2010 International Conference on Dependable Systems and Networks Workshops (DSN-W). IEEE, 2010: 8-14.

[33] Soualhia M, Khomh F, Tahar S. Predicting scheduling failures in the cloud: A case study with google clusters and hadoop on amazon EMR[C]//2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems. IEEE, 2015: 58-65.

[34] Gu J, Zheng Z, Lan Z, et al. Dynamic meta-learning for failure prediction in large-scale systems: A case study[C]//2008 37th International Conference on Parallel Processing. IEEE, 2008: 157-164.

[35] Lan Z, Gu J, Zheng Z, et al. A study of dynamic meta-learning for failure prediction in large-scale systems[J]. Journal of Parallel and Distributed Computing, 2010, 70(6): 630-643.