Practical Online Failure Prediction for Blue Gene/P: Period-based vs Event-driven

Li Yu, Ziming Zheng, Zhiling Lan
Department of Computer Science
Illinois Institute of Technology
{lyu17, zzheng11, lan}@iit.edu

Susan Coghlan
Leadership Computing Facility
Argonne National Laboratory
smc@alcf.anl.gov

***Abstract*—To facilitate proactive fault management in large-scale systems such as IBM Blue Gene/P, online failure prediction is of paramount importance. While many techniques have been presented for online failure prediction, questions arise regarding two commonly used approaches: period-based and event-driven. Which one has better accuracy? What is the best *observation window* (i.e., the time interval used to collect evidence before making a prediction)? How does the *lead time* (i.e., the time interval from the prediction to the failure occurrence) impact prediction arruracy? To answer these questions, we analyze and compare period-based and event-driven prediction approaches via a Bayesian prediction model. We evaluate these prediction approaches, under a variety of testing parameters, by means of RAS logs collected from a production supercomputer at Argonne National Laboratory. Experimental results show that the period-based Bayesian model and the event-driven Bayesian model can achieve up to 65.0% and 83.8% prediction accuracy, respectively. Furthermore, our sensitivity study indicates that the event-driven approach seems more suitable for proactive fault management in large-scale systems like Blue Gene/P.**

I. Introduction

A. Motivation

Proactive fault management has been studied to meet the increasing demands of reliability and availability in large-scale systems. The process of proactive fault management usually consists of four steps: online failure prediction, further diagnosis, action scheduling and execution of actions [15]. It is widely acknowledged that online failure prediction is crucial for proactive fault management. The accuracy of failure prediction can greatly impact the effectiveness of fault management. On one hand, a fault tolerant action, as a response to a failure warning, becomes useless if the prediction itself is a false alarm. Consequently, in case of too many false alarms, a high management overhead may be introduced due to a large amount of unnecessary fault management actions. On the other hand, if too many failures are missed by the predictor, the effectiveness of fault management is questionable. Li et al. have shown that run-time fault management can be effective only when the prediction can achieve an acceptable accuracy.

Generally speaking, online failure prediction methods can be classified into two groups: the period-based approach and the event-driven approach, differing in the trigger mechanism [15].

1) Period-based approach: Typically, a prediction cycle of a period-based method consists of three parts as shown in Figure 1: an *observation window* W_{obs}, a *lead time* W_{lt} and a *prediction window* W_{pdt}. W_{obs} is usually composed of a set of consecutive time intervals $I = \{I_1, I_2, ..., I_n\}$, where each interval has the same size as W_{pdt}, so W_{obs} is n times longer than W_{pdt}. In a prediction cycle, the *observation window* W_{obs} is used to collect evidence that determines whether a failure will occur within the *prediction window* W_{pdt}. Lead time is the time interval preceding the the time of failure occurrence. To be practical, lead time is supposed to be long enough to perform a desired proactive fault prevention.

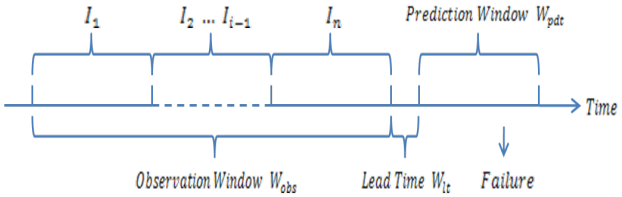


Fig. 1. Period-based approach

2) Event-driven approach: In an event-driven method, the triggering of a failure alarm is determined by events. Strictly speaking, the predictor needs to continuously keep track of every event occurrence until a failure alarm. However, in practice, there still exists an *observation window* W_{obs} for event-driven approach. There are two reasons for doing so. First, it is impractical to keep track of every event occurring before a failure due to the potential amount of events that could happen in a large-scale system. Second, many studies have shown that the events occurred too far away from a failure are less likely correlated to the failure. Hence, in an event-driven method, the predictor keeps on moving W_{obs} forward and the events outside of W_{obs} are not considered. Figure 2 illustrates the main components of a prediction cycle in the event-driven approach: W_{obs}, W_{lt} and $Failure$. Unlike the period-based approach, a predictor using the event-driven approach predicts whether a failure will occur or not right after W_{lt}.

B. Main Contributions

Both event-driven and period based approaches have great potential for fault management in large-scale systems. In this paper, we analyze and compare the impact of *observation window* and *lead time* on both period-based and event-driven

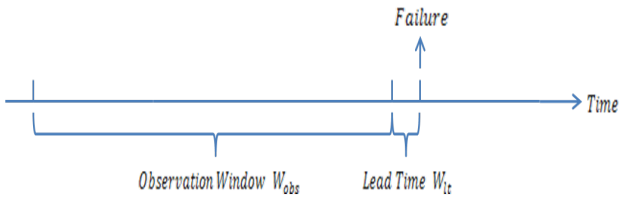


Fig. 2. Event-driven approach

prediction approaches by means of real system logs collected from a production supercomputer. The objective is two-folded: one is to show which prediction approach provides better accuracy and is more suitable for practical use in reality, and the other is to provide some guidance in terms of the design of proactive failure management. More specifically, this paper makes the following major contributions:

- We develop an online Bayesian-based failure prediction method and implement it via both period-based and event-driven approaches.
- We evaluate these prediction approaches under a variety of testing parameters. In particular, we examine the sensitivity of observation window and lead time on both prediction mechanisms. To the best of our knowledge, this paper is the first to study the time characteristics of these two commonly used prediction approaches in large-scale systems.

C. Organization

The organization of this paper is as follows. Section I-I briefly describes the production Blue Gene/P system at Argonne National Laboratory and the RAS log collected from this machine. Section III presents the details of our methodology. Section IV presents the experimental results, followed by a discussion. Section V discusses related work and Section VI draws a conclusion.

II. Background

The RAS log used for this study was collected from the production Blue Gene/P system called *Intrepid* at Argonne National Laboratory. Intrepid is a 40-rack Blue Gene/P system, in which the 40 racks are laid in 5 rows (i.e., R0 to R4). It consists of 40,960 compute nodes with a total of 163,840 cores, offering a peak performance of 556 TFlops. It ranks the #9 on the latest TOP500 supercomputer list (June 2010) [21]. In Intrepid, a Core Monitoring and Control System (CMCS) monitors the hardware components such as compute nodes, I/O nodes and networks and reports the monitored information as RAS event messages that are stored in back-end DB2 databases. An example of event record from Intrepid is shown in TABLE I. The meanings of major entries in the example record are as below.

- **RECID:** the sequence number of the event record.
- **MSG_ID:** the source of the message.
- **COMPONENT:** the software component detecting and reporting the event, including MMCS, KERNEL, CARD, BAREMETAL, MC, DIAGS and APPLICATION.

RECID	13718190
MSG_ID	CARD_0411
COMPONENT	CARD
SUBCOMPONENT	PALOMINO_S
ERRCODE	DetectedClockCardErrors
SEVERITY	FATAL
EVENT_TIME	2008-04-14-15.08.12.285324
FLAGS	DefaultControlEventListener
LOCATION	R-04-M0-S
SERIANUMBER	44V4173YL11K8021017
MESSAGE	An error(s) was detectedby the Clock card : Error=Loss of reference input

TABLE I
AN EXAMPLE OF EVENT FROM BLUE GENE/P RAS LOG.

- **SUBCOMPONENT:** the functional area that generates the message for each component.
- **ERRCODE:** the fine-gained event type information.
- **SEVERITY:** it can be either DEBUG, TRACE, INFO, WARNING, ERROR, or FATAL.
- **EVEVT_TIME:** the start time of the event.
- **LOCATION:** the location where the event occurs
- **MESSAGE:** a brief overview of the event condition.

The failure(s) we try to predict actually refer to the event record(s) with FATAL severity, which is represented as FATAL event(s) in the following of this paper. A FATAL interval refers to a time interval in the RAS log, during which at least one FATAL event occurs. The terms NON-FATAL event(s) and NON-FATAL interval(s) are defined in a similar way.

The RAS log contains health related events occurred from 2008-03-11 00:03:27 to 2008-08-28 10:11:59. We first filtered and cleaned the log using an iterative approach [5][2]. To generate the dataset, we adopt different methods in two approaches. In the period-based approach, the sampling begins at the first entry in the log, each time we move the sampling window forward by one W_{obs}. The size of the dataset depends on the size of W_{obs} and W_{lt} used for that test. In the event-driven approach, all FATAL events are extracted first, then NON-FATAL events that are five times number of FATAL events are selected from the rest of the log randomly. Based on the occurring time of these extracted events, we generate the dataset by searching within W_{obs}. Unlike the period-based approach, the event-driven approach has datasets with constant size.

III. Methodology

A. Bayesian Network Classifier

A Bayesian network B is a probabilistic graphical model that represents a set of random variables $x = \{x_1, x_2, ..., x_n\}$ and their conditional dependencies via a directed acyclic graph (DAG). In Bayesian networks, nodes represent random variables and edges represent conditional dependencies; nodes which are not connected indicate conditionally independent variables. Each node is associated with a probability function that takes a set of values of its parent variables as input and gives the probability of the variable represented by the node [7].

In this study, we build a Bayesian network to model the causality between current observations and failures that appear later. That is, given a set of attribute variables (observations) $o = \{o_1, o_2...o_n\}$ and a class variable (failure) f, a Bayesian network classifier maps an instance of o to a value of f. And to make a failure prediction, the classifier simply computes $\arg\min_y P(f|o)$ using the distribution $P(x)$ represented by B.

$$P(f|o) = \frac{P(x)}{P(o)} \propto P(x) \tag{1}$$

$$P(x) = \prod_{c \epsilon x} P(c|pa(c)) \tag{2}$$

Note that in Equation (1) , $P(x) = P(o \cup f)$, given an instance of observation, $f(o)$ is known, so the classification can be made using only Equation (2).

B. Random Variables

All random variables used in our Bayesian network are binary, and are classified into three categories: (1) *correlation attribute:* a total number of 68 attribute variables representing the occurrence of 68 NON-FATAL events within W_{obs}; (2) *statistic attribute:* 4 attribute variables used to capture statistical features during the W_{obs} window and; (3) *Failure State:* a single class variable named $Failure$ with either *TRUE* or *FALSE* state, representing whether the FATAL events (intervals) appear or not. We give the detailed description of the first two categories as below.

1) Correlation attribute: The idea of using the occurrence of NON-FATAL events as attribute variables derives from a common method used for failure prediction: association rule based techniques [6][18][17]. Intuitively, the direction of causality in our Byesian network model should be: [events with lower severity] $\rightarrow$ [events with higher severity]. In other words, we assume that events with lower severity usually occur in advance and indicate a trend that events with higher severity will appear later. In the RAS log used for our experiment, there are a total number of 243 kinds of NON-FATAL events with distinct ERRCODEs, each of which is a candidate as a random variable. However, not all of them have causal relationships with future FATAL events. According to the association rules extracted by [8], we select 68 out of these 243 kinds of NONFATAL events to be random variables. Each variable is named after its ERRCODE of the corresponding NON-FATAL event and has two states: *YES* and *NO*. Within W_{obs}, if an event belonging to one of the 68 kinds occurs, the state of the corresponding variable will be set to *YES*. Table II summarizes these 68 correlation variables.

2) Statistic attribute: Four variables are used in our Bayesian network to capture statistical characteristics in W_{obs}. The statistical characteristics such as the frequency of failure occurrence and time between failures have proven to be very useful for failure prediction [16][20][3]. Each attribute variable in this category has two states: *NORMAL* and *ABNORMAL*. The state of the variable depends on whether the statistical feature it represents exceeds a predefined threshold. If yes, the state will be *ABNORMAL*, otherwise, it will be *NORMAL*. These four statistic attributes are:

COMPONENT	SEVERITY		
	INFO	WARN	ERROR
KERNEL	1	19	1
CARD	4	2	3
BAREMETAL	10	0	0
DIAGS	2	6	4
MMCS	4	1	7
MC	4	0	0

TABLE II
COMPONENTS AND SEVERITY LEVELS OF THE 68 CORRELATION VARIABLES

- *Event Deviation:* it describes the deviation of the number of events in W_{obs} from the average number, defined as $\frac{Num\_Events - Avg\_Events}{Avg\_Events}$, where Num_Events is the number of events (all severity levels) occurring within W_{obs} and Avg_Events is the average number.
- *Fatal Rate in W_{obs}:* the percentage of FATAL events in W_{obs}.
- *Fatal Rate in I_n:* the percentage of FATAL events in I_n.
- *Time from Last Failure:* the time span from the last failure to the beginning of W_{pdt}.

Note that the variable *Fatal Rate* I_n is only applicable for the period-based approach because, in the event-driven approach, W_{pdt} is not divided into time intervals. Similarly, the beginning of W_{pdt} refers to the definition in the period-based approach, for event-driven, it should be the time of $Failure$. (see Figure 2). The variable *Event Deviation* describes how far the number of events in W_{obs} differs from the average number. The variables *Fatal Rate in* W_{obs} and *Fatal Rate in* I_n compute the ratio of FATAL events to all events occurring in the time window, indicating a commonly used assumption that the ratio of FATAL events will be higher than usual if another failure is approaching. In other words, if multiple failures occur within a short period of time, it is highly possible another failure will come soon. The variable *Time from Last_Failure* captures the distribution of time between failures and typically the longer the time is from the last failure, the higher possibility it is to see another failure.

IV. EXPERIMENT

We conduct two sets of experiments. In the first set, we analyze the impact of W_{obs} on both period-based and event-driven approaches and focus on the best performance they can achieve respectively without considering W_{lt}. In the second set, we apply different W_{obs} and W_{lt} in both approaches and

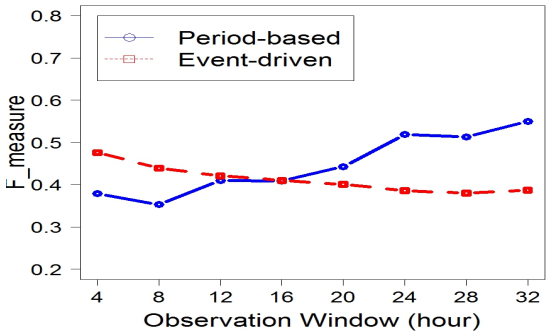
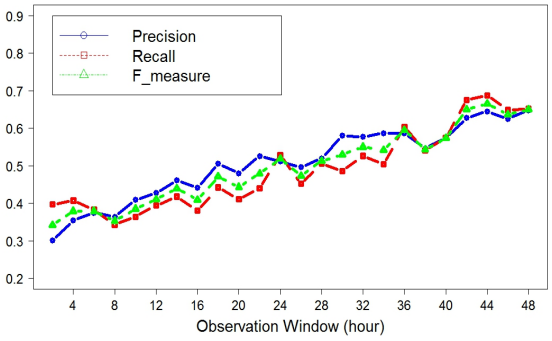
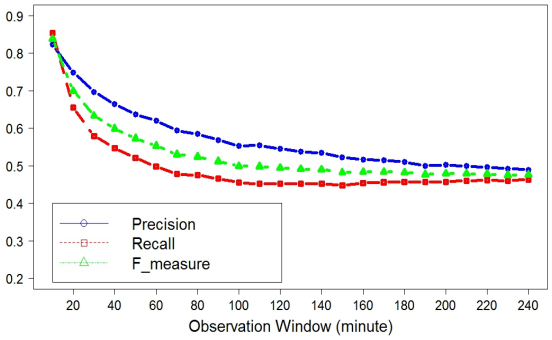


Fig. 3. The interaction of two approaches



(a) Period-based



(b) Event-driven

Fig. 4. Evaluation on both approaches using different W_{obs}

evaluate the effects of lead time on their accuracy. All the experimental results given in this section are based on 10-fold cross validation.

A. Evaluation Metrics

Three metrics are used to evaluate prediction accuracy. Using the symbols shown in TABLE III, the definitions of these metrics are as below.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F\_measure = \frac{2TP}{2TP + FP + FN}$$

Since predictors with higher *precision* usually have lower *recall*, *F_measure* provides a balanced measure of the goodness of prediction methods.

Predicted Class		FATAL	NON-FATAL
True Class	FATAL	TP	FN
	NON-FATAL	FP	TN

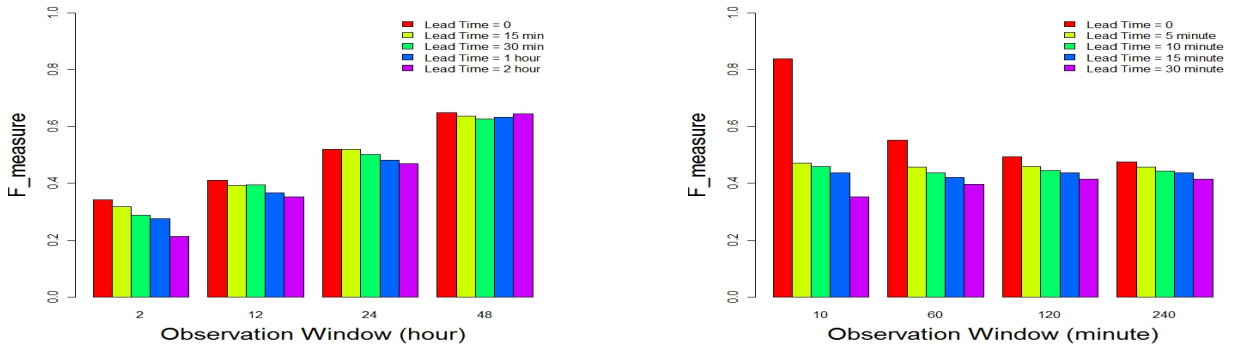
TABLE III
THE CONFUSION TABLE

B. Results

In the first set of experiments, we examine the impact of W_{obs} on both period-based and event-driven approaches, with lead time being set to zero. We vary W_{obs} from 10 minutes to 48 hours. Figure 3 shows the trends of *F_measure* results for both approaches, where we only list the results of W_{obs} setting between 4 hours and 32 hours. It is a little bit surprising that the event-driven approach outperforms the period-based one till W_{obs} gets close to 16 hours.

As we can see from Figure 3, the accuracy achieved by the period-based approach is increasing with the growth of W_{obs}, whereas it is decreasing in case of applying the event-driven approach. To better illustrate the detailed difference between these two approaches, Figure 4 presents the *precision*, *recall* and *F_measure* results for these approaches in separate plots. In our experiment, the period-based approach achieves its best performance (i.e., 64.8% $precision$ and 65.2% $recall$) when W_{obs} is set to 48 hours, while the event-driven one reaches its peak (i.e., 82.3% $precision$ and 85.4% $recall$) when W_{obs} is set to 10 minutes. Without considering the difference of W_{obs}, the event-driven approach outperforms the period-based approach significantly.

Further, as shown in Figure 4, the evaluation results of these two approaches show different characteristics. First, the optimal $W_{obs}(s)$ to achieve the highest accuracy are significantly different (i.e., 48 hours vs 10 minutes). As a result, the event-driven approach seems suitable for minute-level prediction, which facilities fast and low-overhead fault management actions, whereas the period-based approach is more fit for long-term prediction (e.g., in the scope of dozens of hours). Second, the event-driven approach is more sensitive to W_{obs} than the period-based one. The period-based approach gets about 30% accuracy enhancement from the 2-hour W_{obs} to the 48-hour W_{obs}, while the event-driven approach lose its accuracy up to 35% within about 200 minutes. Therefore it is more critical to identify the optimal W_{obs} for event-driven approach.



(a) Period-based

(b) Event-driven

Fig. 5. The effects of lead time on both approaches

In the second set of experiments, we investigate the impact of W_{lt} on both period-based and event-driven approaches. In previous studies [19][11], W_{lt} varies according to W_{obs}, and the increasing of W_{lt} leads to lower accuracy. Since most fault management operations in the Blue Gene/P system require at least 5 minutes and can be completed within 2 hours [9][12], we use $W_{lt} \in \{15min, 30min, 1hour, 2hour\}$ for the period-based approach and $W_{lt} \in \{5min, 10min, 15min, 30min\}$ for the event-driven approach in the experiment.

The results are shown in Figure 5. Obviously, the period-based approach is less sensitive to W_{lt} than the event-driven one. On one hand, as shown in Figure 5(a), the 2-hour W_{lt} only leads to about 10% loss of accuracy for the 2-hour W_{obs}, and the impact of W_{lt} decreases as W_{obs} grows. When W_{obs} is set to 48 hours, the impact of W_{lt} is negligible. The introduction of the 2-hour W_{lt} can lead to an even higher accuracy due to less noisy information. On the other hand, as shown in Figure 5(b), there is a dramatic drop of accuracy (i.e., about 40%) for the 10-minute W_{obs} when a 5-minute W_{lt} is added.

There are several explanations on the observations from Figure 5. First, although $W_{lt}(s)$ used in the event-driven approach is much smaller than that used in the period-based approach, their ratios to $W_{obs}(s)$ are larger. In other words, a higher ratio of W_{lt}/W_{obs} substantially introduces more side effects. Second, the event-driven approach is very sensitive to NON-FATAL events occurring right preceding the failure. That is the reason why the event-driven predictor can achieve its best performance with W_{obs} around 10 minutes, and long lead time tends to reduce prediction accuracy significantly. On the contrary, the period-based approach takes more benefits from statistic attributes, which is more robust to the impact of lead time. Third, since the period-based approach is made for the time interval W_{pdt} rather than a single event, there is a time interval between the beginning point of W_{pdt} and the failure occurrence time. As a result, a natural lead time exists in the period-based approach, and the side effects introduced by W_{lt} are trivial.

Based on the observations above, we further study the advantages and disadvantages of both approaches. First, although the event-driven approach is sensitive to W_{obs}, we can not simply declaim that it is only suitable for minute-level prediction. This is because it outperforms period-based till W_{obs} gets close to 16 hours, which can not be considered as short-term. Second, although the period-based approach can achieve even higher accuracy if we keep on increasing W_{obs} in the experiment, it is not practical for real applications. There are three reasons: (1) few fault management operations need that long-term prediction (e.g., 48-hour W_{obs} or longer); (2) since one or more failures can occur at any time points within W_{pdt}, a large W_{pdt} indicates less certainty about failures' occurring time; and (3) the more failures may occur within W_{pdt}, the more difficulties to give an accurate diagnosis for their locations, which is crucial to action scheduling in fault management. Third, although the event-driven approach is more sensitive to W_{lt}, its accuracy with W_{lt} added is still comparable to that of period-based approach (see Figure 5). Finally, Table IV summarizes our findings from this study, where the two approaches are separated by the 16-hour W_{obs}.

V. Related Work

Recognizing the importance of proactive fault management, considerable research has been conducted on failure prediction. Salfner et al. give a comprehensive survey of existing online failure prediction technologies in [15]. Based on the monitoring and trigger mechanism, existing methods can be broadly classified as event-driven approach or period-based approach [15]. In large-scale systems, a majority of online failure predictors are based on the event-driven approach. For example, Sahoo et. al. uses association rules for failure prediction in a 350-node IBM cluster [14]. In [10], several statistical based techniques are studied to capture the event causal correlations for failure forecasting in a Blue Gene/L system. In our previous study [8], we investigate a dynamic meta-learning prediction engine by adaptively combining the merits of various data mining techniques. While event-driven approach has been studied extensively, research on period-based approach for large-scale systems is limited. As a

	Event-driven			Period-based	
$\mathbf{W_{obs}}$	10 min	10 min $\sim$ 60 min	1 hour $\sim$ 16 hour	16 hour $\sim$ 48 hour	48 hour
F_measure	Best accuracy 83.8%	Drops to 55.3% dramatically	Drops to 41.0% smoothly	Increase from 40.9% smoothly in a roundabout manner	Best accuracy 65.0%
Sensitivity to W_{lt}	High	Medium	Low	Low	Low
Best fit	Short-term prediction without W_{lt}	Medium-term prediction with W_{lt}	Not recommend	Not recommend	Long-term prediction

TABLE IV
SUMMARY OF EXPERIMENTAL RESULTS

complement to existing work, this study compares these two approaches in terms of prediction accuracy and practical use in reality. To the best of our knowledge, we are not aware of any such comparison for failure prediction in large-scale systems like Blue Gene series.

Our study in the period-based approach is inspired by the Liang's work in [20], which periodically explores three different classifiers and evaluates them with Blue Gene/L RAS logs. Our work differs from [20] in two key aspects. First, while their study only uses the statistical characteristics as the feature for prediction, we collect both correlation attributes and statistic attributes to capture system-wide symptoms and improve the prediction accuracy. Second, the lead time is explicitly considered in our predictor, and we study the impact of lead time on prediction accuracy in our experiments.

Bayesian methods have been widely used for anomaly prediction. For example, Hamerly and Elkan present both supervised and unsupervised methods based on naive Bayes classifier to predict disk failures [4]. Pizza et. al. propose a Bayesian method to distinguish transient faults from permanent faults [13]. In this paper, we design a general Bayesian classifier for the prediction, which is adopted for both period-based and event-driven approaches.

VI. CONCLUSION

In this paper, we have presented a comparison of event-driven and period-based failure predication approaches for high performance computing systems. The proposed Bayesian-based predictor has proven to be effective for both period-based and event-driven approaches, achieving up to 65.0% and 83.8% prediction accuracy respectively. Experimental results show that the event-driven approach outperforms the period-based one significantly without considering W_{lt}. Although the period-based approach has the advantage of less sensitivity to both W_{obs} and W_{lt} and is suitable for long-term prediction (i.e., 48-hour W_{obs} or longer), considering the practical time consumed by commonly adopted fault management strategies, the event-driven approach is preferred in most cases.

We are planning to study more cases with a variety of HPC systems, such as the Cray XT5 at ORNL. This research will be integrated with the FENCE [1] project and deployed on real systems for improving the overall fault management.

ACKNOWLEDGMENT

The work at Illinois Institute of Technology is supported in part by US National Science Foundation grants CNS-0834514, CNS-0720549, and CCF-0702737. This work was supported in part by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

REFERENCES

[1] http://www.cs.iit.edu/ zlan/fence.html.
[2] M. Buckley and D. Siewiorek. Comparative analysis of event tupling schemes. *Proc. of Fault-Tolerant Computing*, 1996.
[3] S. Fu and C. Xu. Exploring event correlation for failure prediction in coalitions of clusters. *Proc. of Supercomputing*, 2007.
[4] G. Hamerly and C. Elkan. Bayesian approaches to failure prediction for disk drives. *Proc. of ICML*, 2001.
[5] J. Hansen and D. Siewiorek. Models for time coalescence in event logs. *Proc. of Fault-Tolerant Computing*, 1992.
[6] K. Hatonen, M. Klemettinen, H. Mannila, P., and H. Toivonen. Tasa: Telecommunication alarm sequence analyzer or: How to enjoy faults in your network. *IEEE/IFIP Network Operations and Management Symposium*, 1996.
[7] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
[8] Z. Lan, J. Gu, Z. Zheng, R. Thakur, and S. Coghlan. A study of dynamic meta-learning for failure prediction in large-scale systems. *J. Parallel Distrib. Comput.*, 70:630–643, June 2010.
[9] Y. Li, Z. Lan, P. Gujrati, and X. Sun. Fault-aware runtime strategies for high-performance computing. *IEEE Trans. Parallel Distrib. Syst.*, 20:460–473, April 2009.
[10] Y. Liang, Y. Zhang, M. Jette, A. Sivasubramaniam, and R. Sahoo. BlueGene/L failure analysis and prediction models. *Proc. of DSN*, 2006.
[11] H. Meng, Di H., and Y. Chen. A rough wavelet network model with genetic algorithm and its application to aging forecasting of application server. *International Conference on Machine Learning and Cybernetics*, 2007.
[12] H. Naik, R. Gupta, and P. Beckman. Analyzing checkpointing trends for applications on the ibm blue gene/p system. *Workshop on P2S2 in conjunction with ICPP*, 2009.
[13] M. Pizza, L. Strigini, A. Bondavalli, and F. Gi. Optimal discrimination between transient and permanent faults. *IEEE International High-Assurance Systems Engineering Symposium*, 1998.
[14] R. Sahoo, A. Oliner, I. Rish, M. Gupta, E. Moreira, S. Ma, R. Vilalta, and A. Sivasubramaniam. Critical event prediction for proactive management in large-scale computer clusters. *Proc. of ACM SIGKDD*, 2003.
[15] F. Salfner, M. Lenk, and M. Malek. A survey of online failure prediction methods. *ACM Comput. Surv.*, 42:10:1–10:42, March 2010.
[16] B. Schroeder and G. Gibson. A large-scale study of failures in high-performance computing systems. *Proc. of DSN*, 2006.
[17] R. Vilalta and S. Ma. Predicting rare events in temporal domains. *Proc. of ICDM*, 2002.
[18] G. Weiss. Timeweaver: a genetic algorithm for identifying predictive patterns in sequences of events. *Proc. of the Genetic and Evolutionary Computation Conference*, 1999.
[19] G. Weiss. Predicting telecommunication equipment failures from sequences of network alarms. *Handbook of Data Mining and Knowledge Discovery*, pages 891–896, 2001.
[20] Y. Zhang and A. Sivasubramaniam. Failure prediction in IBM BlueGene/L event logs. *Proc. of IPDPS*, 2008.
[21] Z. Zheng, L. Yu, W. Tang, Z. Lan, R. Gupta, N. Desai, S. Coghlan, and D. Buettner. Co-analysis of ras log and job log on blue gene/p. *Proc. of IEEE International Parallel & Distributed Processing Symposium*, 2011.