Online Failure Prediction in Cloud Datacenters

● Yukihiro Watanabe ● Yasuhide Matsumoto

Once failures occur in a cloud datacenter accommodating a large number of virtual resources, they tend to spread rapidly and widely, impacting many cloud services and their users. One of the best ways to prevent a failure from spreading in the system is to identify signs of a failure before its occurrence and deal with it proactively before it causes serious problems. Although several approaches have been proposed to predict failures by analyzing past logs of system messages and identifying the relationship between the messages and the failures, it is still difficult to automatically predict the failure for several reasons such as variation of log message formats and frequent changes in their configurations. Based on this understanding, we propose a new failure prediction method that Fujitsu Laboratories has developed. The method automatically learns message patterns as signs of failure by classifying messages by their similarity regardless of their format and re-learning the message patterns in frequently changed configurations. We evaluated our method in an actual cloud datacenter. The experimental results showed that our approach predicted failures with 80% precision and 90% recall in the best case.

1. Introduction

As cloud computing has recently become widespread, users have come to be able to procure the necessary computer resources only for the periods of time required without needing to have their own servers. While cloud computing brings significant convenience to users, new issues have emerged in operations and management that supports the cloud. In a cloud datacenter, many users share computer resources on a virtualization platform. Once failures occur in such an environment, they tend to spread rapidly and widely. In addition, it takes a lot of time before the failures can be contained because hardware is hidden by virtualization technology. Accordingly, it is important to promptly detect failures and respond to them before they become serious. In order to realize reliable, low-cost operations, the existing after-the-fact approach of responding after the occurrence of a failure must be replaced with a proactive one in which a failure is dealt with in advance.

Fujitsu Laboratories has developed a method in which message patterns are created and learned in real time so as to identify signs of failure in advance and respond promptly. This paper presents the failure prediction technology based on message pattern learning and the results of an evaluation of its performance obtained by experimental failure prediction through online acquisition of messages in an actual cloud datacenter.

2. Issues with failure prediction

One way to promptly detect failures is to try to predict any failure that may affect services based on the behavior of the devices that constitute a system. Various approaches have been proposed in this field, many of which are based on analyzing message logs output by devices constituting a system and identifying message patterns related with failures. Salfner et al. used a hidden semi-Markov model (HSMM) to analyze the order of messages in a log and identified message sequences related to failures.[1] However, applying these methods to a large system such as a cloud datacenter poses some issues, as described below.

1) Various message formats

In large cloud datacenters, systems tend to consist of different device models from different vendors. The formats of message output from various components greatly vary and are not uniform, unlike logs in a high-performance computing (HPC) environment which were the subject of past research. This makes message classification difficult.

2) Failure to strictly guarantee order of messages

Operating a large system involves collecting messages from a large number of devices that constitute the system. Due to time lags between devices and differences in network delays during message collection, the order of messages collected and recorded may not be the same as the order in which the messages were actually output. For this reason, the existing method of considering the order of messages is incapable of adequately learning the signs of failures.

3) Obsolescence of results of learning

In a cloud datacenter, some of the devices contained there may be changed at any time because of the need to replace them or upgrade software, and this renders the results of analysis for failure prediction obsolete in short periods of time. In order to keep analysis results up to date in such an environment, it is important to analyze signs of failures in real time and promptly reflect the results of the analysis in failure prediction systems so as to remain updated.

3. Online failure prediction

To resolve the issues mentioned in the previous section, we have developed a method in which message patterns are created and learned in real time by following the procedure described below to identify signs of failures. The terms used in this paper are based on a reference,[2)] but they have been altered slightly to suit the present method (**Figure 1**).

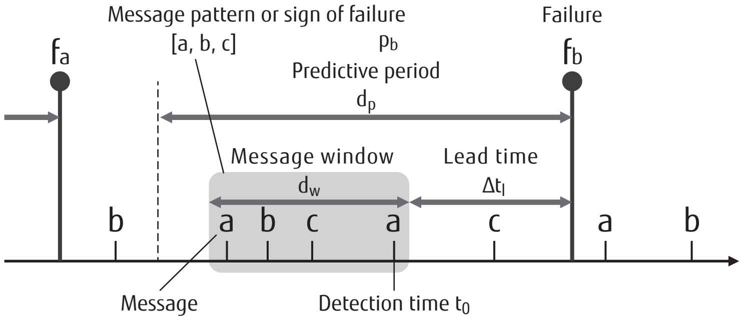
1) Message classification

First, as shown in **Figure 2**, the acquired messages are split into words and compared with each entry in the message dictionary. Each message is classified into the entry with the largest number of word matches with the words included in the message.

2) Message pattern learning

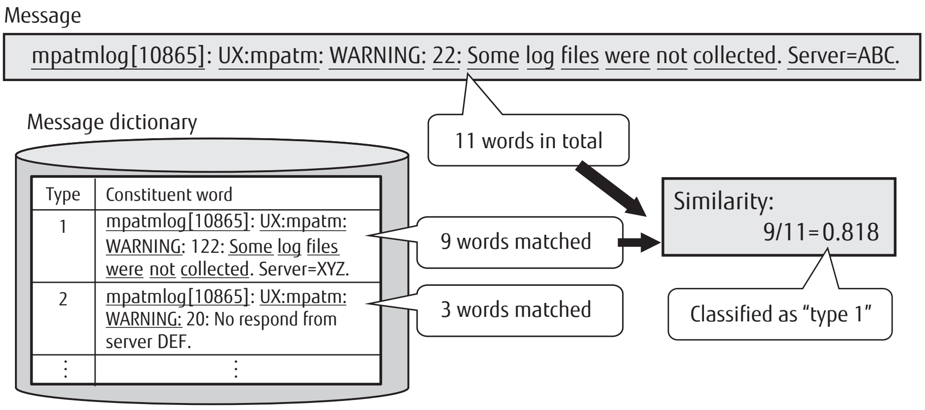
Then, with a set of types of messages in the last few minutes as of a certain time defined as a message pattern, the relationship between message patterns and failures are learned by using Bayesian inference [**Figure 3 (a)**]. The probability of occurrence of failure *T* in a certain period after occurrence of message pattern *P* is determined by using the formula:

$$\text{(Probability of occurrence of failure } T\text{) } = \frac{\text{No. of instances of } P \text{ observed in predictive period of } T}{\text{No. of instances of } P \text{ observed in entire period}}$$



Failure: Event affecting service
Lead time: Grace period between detection of sign and occurrence of failure
Message: Event composed of time and text, output by device
Message window: A scope in message logs limited to a certain duration
Message pattern: List of types of messages within message window
Sign of failure: Message pattern with strong co-occurrence relationship with failure
Predictive period: Period before occurrence of failure regarded to contain sign of failure

Figure 1
Definitions.

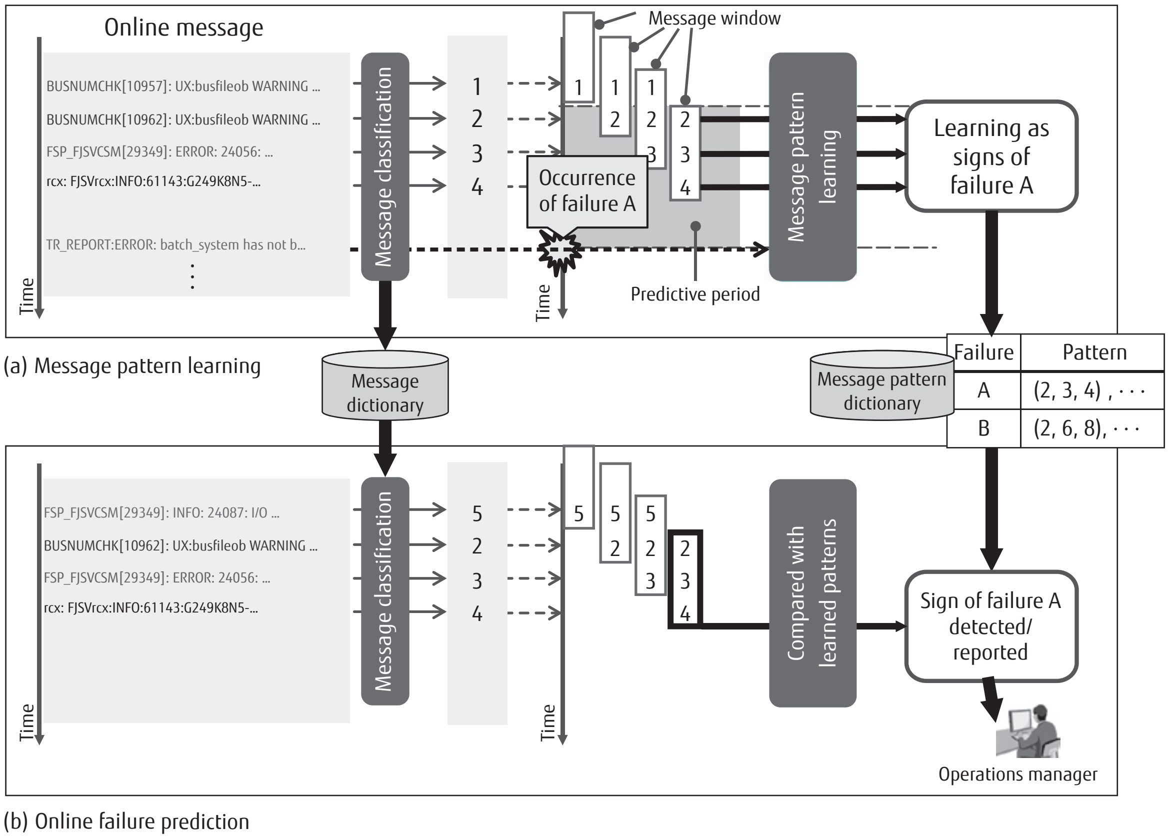


mpatmlog[10865]: UX:mpatm: WARNING: 22: Some log files were not collected. Server=ABC.

Type	Constituent word
1	mpatmlog[10865]: UX:mpatm: WARNING: 122: Some log files were not collected. Server=XYZ.
2	mpatmlog[10865]: UX:mpatm: WARNING: 20: No respond from server DEF.
⋮	⋮

Message dictionary: dictionary containing messages classified in past times and results of classification

Figure 2
Message classification.



BUSNUMCHK[10957]: UX:busfileob WARNING ...
BUSNUMCHK[10962]: UX:busfileob WARNING ...
FSP_FJSVCSM[29349]: ERROR: 24056: ...
rcx: FJSVrcx:INFO:61143:G249K8N5-...
TR_REPORT:ERROR: batch_system has not b...

Failure	Pattern
A	(2, 3, 4) , · · ·
B	(2, 6, 8), · · ·

FSP_FJSVCSM[29349]: INFO: 24087: I/O ...
BUSNUMCHK[10962]: UX:busfileob WARNING ...
FSP_FJSVCSM[29349]: ERROR: 24056: ...
rcx: FJSVrcx:INFO:61143:G249K8N5-...

(a) Message pattern learning

(b) Online failure prediction

Figure 3
Learning and detecting signs.

The probability found is recorded in the message pattern dictionary and, at the same time, the time difference (lead time) between message pattern P and failure T is determined and recorded.

3) Online failure prediction

Messages output from the system are classified to create patterns, which are compared with the results of learning, thereby evaluating the probability of occurrence of various failures in real time. Any probability of failure occurrence higher than the defined threshold is regarded as a sign of a failure and reported to the operations manager [**Figure 3 (b)**].

This method has three features:

1) Message classification independent of format

A cloud environment contains a mixture of messages of various formats. With the present method, messages are classified based on the number of matches of words constituting them, and this allows for a uniform handling of messages of different formats. In addition, messages are automatically classified without interpreting their meanings, which eliminates the need for human intervention for defining the message dictionary.

2) Message pattern creation independent of order

In a cloud environment, the order of messages is not always guaranteed. With the present method, the order of messages is not taken into account but message patterns are created by handling sets of message types and any little change in the order of messages does not affect the result of learning.

3) Real-time message pattern learning

With the present method, input messages are classified in real time to create message patterns and signs of failures are learned and detected. Unlike general pattern learning by batch processing, this method allows the message pattern dictionary to be updated immediately to accommodate any configuration change in the system. In this way, the latest result of learning can be used to detect signs of failures.

4. Evaluation in cloud environment

In order to evaluate the performance of the present method, we used a commercial cloud datacenter for online evaluation.

1) Target system

This system was composed of several hundred physical servers and provided more than 10 000 virtual machines (VMs). In this environment, we collected a message log for 90 days to try reporting signs of failures. During this period, approximately 9.45 million messages were output, and they were classified into 509 types. One hundred and twelve failures also occurred in the period, and they were classified into 20 types. **Table 1** shows examples of the failures that occurred.

Table 1
Examples of failures generated.

Failure type	Description	No. of occurrences
a	Batch system failure #1	21
b	Process operating rate error	10
c	Threshold error	10
d	Storage node stop	21
e	Batch system failure #2	7
f	Batch system failure #3	6
g	Unexpected node restart	5
h	Disc copy failure	7
Other (12 type)		25
Total		112

2) Implementation

We prototyped the online failure prediction system and installed it on a VM in the management area of the actual commercial cloud datacenter (**Figure 4**). For prototyping, we used Java and MySQL. The performance of the VM corresponds to Xeon 2.0 GHz in terms of CPU and 3.4 GB in terms of memory.

3) Metrics

As the metrics for evaluation, we chose the following three, which are often used in research in the field of failure prediction:

- Precision: ratio of correctly identified failures to the number of all predicted failures
- Recall: ratio of correctly predicted failures to the number of true failures
- F-measure: harmonic mean of precision and recall

Generally, there tends to be a trade-off between precision and recall. Attempting to avoid missing of signs of failures causes many false detections. Frequent occurrence of false detections in operations and management of a system increases the working hours of the manager and the cost. Accordingly, performance of failure prediction must be controlled within an allowable range for actual operations.

4) Results

For the respective failure types listed in Table 1, the failure prediction performance with a threshold of 0.99 is shown in **Figure 5**. To take an example offering the best figures, the results of prediction for failure type A were 80% (24/30) in precision, 90% (19/21) in recall and 0.85 in F-measure.

5) Discussion (difference in failure prediction

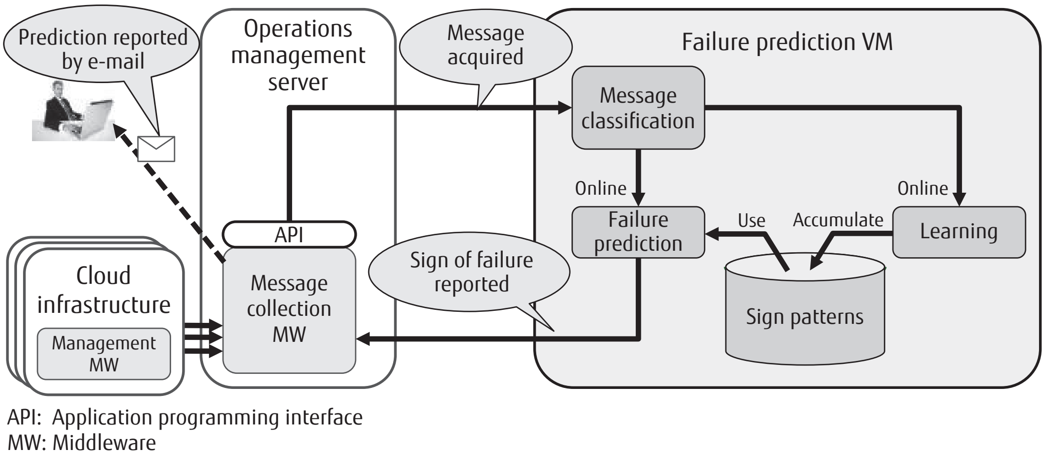


Figure 4
Trial of failure prediction.

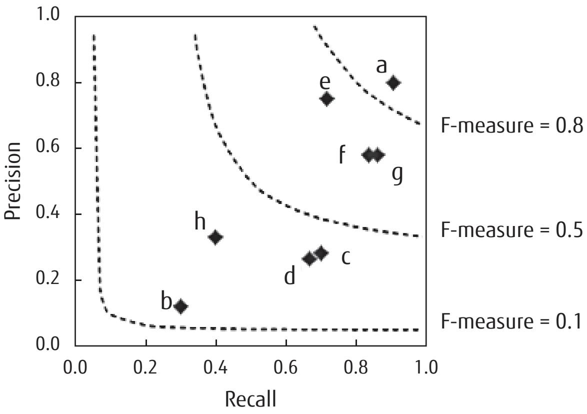


Figure 5
Failure prediction performance for respective failures.

performance caused by nature of failure)

The evaluation has shown that precision and recall may largely vary depending on the failure type. In order to probe into the cause, we carried out an analysis with the focus on the relationship between the lead time of message patterns learned and performance of failure prediction. As a result of the analysis, we have found out that failures can be classified into three categories:

- Gradual decrease

Precision decreases as the lead time becomes longer. Failures showing this tendency were mostly those assumed to occur as a result of accumulation of minor errors such as process hangs. These failures require the operator to consider the balance between the time required for responding to them and precision. To take the process hang failure mentioned above as an example, precision was 77% for a lead time of 0 to 10 minutes, 52% for 10 to 20 minutes and 17% for 20 to 30 minutes. The time needed to make a response for avoiding this failure (process restart) is about 10 minutes. In this case, the performance of failure prediction is low with a lead time of 20 minutes or longer and a lead time of shorter than 10 minutes is too short for response. Accordingly, in actual operations, reporting only signs of failures with a lead time of 10 minutes or longer and shorter than 20 minutes allows operators to be notified of only "signs of failures that are likely to be accurate and can be handled before it occurs."

- Long term

There are a certain number of signs of failures that are accurate even with a long lead time. Many of the failures showing this tendency were those accompanied by hardware errors such as storage device stop. For these failures, signs are detected one hour or longer before the occurrence, although with a low precision of around 50%. Accordingly, of these, failures that are critical and take time to take avoidance action can be dealt with by proactively reporting the signs, and this should reduce the number of critical failures.

- Immediately prior

There are signs that are accurate only when the lead time is very short at less than 10 minutes. Failures showing this tendency are likely to occur immediately after a certain operation of a device by human intervention or a program, such as failure in VM migration. For these failures, measures including prevention of the occurrence of a failure cause by techniques such as pre-verification of operation procedures, rather than failure prediction, seem effective.

To apply failure prediction to the operations of actual cloud datacenters, it is necessary to implement a function that can suppress the reporting of signs depending on the characteristics of failures in order to avoid false prediction of failures.

5. Conclusion

This paper has presented the failure prediction method developed by Fujitsu Laboratories for responding before serious failures occur, and the results of an online evaluation in an actual cloud datacenter. The results of evaluating this method, which calculates relationships between sets of messages and failures, showed that it can detect signs of failures in a large-scale cloud computing environment where it is difficult to apply conventional techniques. In addition, it has been suggested that failure prediction can be efficiently dealt with by classifying the natures of failures and characteristics of precision of failure prediction into three categories and taking those characteristics into consideration.

This method does not delve into the mechanism between message patterns and failures but extracts relationships in a statistical manner. Our idea is that operations and management of infrastructure of cloud datacenters can be improved by integrating this method into troubleshooting processes for cloud infrastructure and using it in combination with other analytical methods, configuration information and incident records.

References

1) F. Salfner et al.: Using Hidden Semi-Markov Models for Effective Online Failure Prediction. In Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems. Washington, DC, USA, IEEE Computer Society, pp. 161–174, 2007.

2) F. Salfner et al.: A survey of Online Failure Prediction Methods. *ACM Comput. Surv*. Vol. 42, No. 3, pp. 10:1–10:42 (2010).

Yukihiro Watanabe
Fujitsu Laboratories Ltd.
Mr. Watanabe is currently engaged in research on operations and management technology for cloud computing environment.

Yasuhide Matsumoto
Fujitsu Laboratories Ltd.
Mr. Matsumoto is currently engaged in research and development for operation and maintenance and standardization of cloud-related technologies.