
Gajendra Sharma

Department of Computer Science and Engineering, Kathmandu University, Dhulikhel-8, Kavre, Nepal
E-mail: gajendra.sharma@ku.edu.np

Abstract. Fault tolerance is an important issue in the field of cloud computing which is concerned with the techniques or mechanism needed to enable a system to tolerate the faults that may encounter during its functioning. Fault tolerance policy can be categorized into three categories viz. proactive, reactive and adaptive. Providing a systematic solution the loss can be minimized and guarantee the availability and reliability of the critical services. The purpose and scope of this study is to recommend Support Vector Machine, a supervised machine learning algorithm to proactively monitor the fault so as to increase the availability and reliability by combining the strength of machine learning algorithm with cloud computing.

Keywords: Cloud computing, fault tolerance, proactive, support vector machine.

1 Introduction

Cloud computing is a widely adopted technology by the industry. It is a style of computing where it’s end used are provided with a services through the internet using different models and layers of abstraction as pay-per-use basis. The services provided by the cloud computing has been divided as:

(a) Software-as-a-Service (SaaS)
(b) Platform-as-a-service (PaaS)
(c) Infrastructure-as-a-Service (IaaS)

With the increasing maturity of cloud computing there is also increase in the research regarding the issues like fault tolerance, workflow scheduling, workflow management, security. Fault tolerance is one of the key factors that may encounter in several communication and computer networks [4, 5]. It is related to entire techniques required to enable a system to tolerate software faults. A fault is a defect or flaw that occurs in some hardware or software component. In the traditional software fault classification is dividing fault into hardware fault and software fault. In cloud computing system the hardware fault and software fault are further classified. General hierarchy of the fault classification is shown in the Figure 1.

The main concern of the fault tolerance is to assure reliability and availability of sensitive services and application execution by reducing the failure influence on the system as well as application execution [6, 7]. The fault should be anticipated and carefully handled. As such fault tolerance technique is to predict failures and take suitable action before failures occur.

Fault tolerance techniques

Depending upon when to apply the fault tolerance techniques Fault tolerance has been classified as [8]:

(a) Proactive fault tolerance:
This is an important technique which predicts the fault and eliminate recovery from faults and failure by substituting the alleged component. It is able to detect the problem before it arises. This is a perception preventing compute
node failures from running parallel application by pre-emptively migration parts of an application.

(b) Reactive fault tolerance:
This mechanism enable to reduce the effort of failures, when the failure takes place. Reactive fault tolerance technique facilitates system more robust or on-demand fault tolerance.

(c) Adaptive fault tolerance:
The fault-tolerance of an application depending on its existing position and the range of control inputs can be applied efficiently. Therefore in adaptive fault tolerance entire procedures are performed automatically as per the condition. It can guarantee reliability of critical modules, under resources constraints and temporal by allocating redundancy to less critical modules. It can be affordable minimizing resource requirement.

Objectives
This paper aims to develop a systematic solution to improve the predictability of the fault tolerant system in cloud computing environment using the machine learning approach which has been proved to be a well-adapted model on different domains.

Statement of the problem
Using the reactive Fault tolerant technique increases the clock execution time of the system. While rule based fault detection may underfit the problem and the model based system are generally complex and computationally intensive which requires a large amount of skilled work to develop a model for particular system.

A systematic solution is required to improve the observability of the system, machine learning algorithm like Support vector Machine may be well efficient solution to solve this particular issue.

The paper is organized as follows: Section 2 presents the related works where the various existing techniques have been pointed as well as a comparison chart is provided to give an analytical view; Section 3 presents the proposed Proactive Fault tolerance Framework; Section 4 shows a detailed illustration of how proposed technique works; Section 5 concludes the paper along with the future works.

2 Related Works
Various fault tolerance are currently prevalent in clouds [2, 8, 9].

Check pointing: In check pointing after doing a change in system a checkpoint is created. Whenever a task fails the job is restarted from the recently added checkpoint.

Job Migration: If the job cannot be executed on a particular machine then it is migrated to another machine where it can be continued.

Replication: It permits several copies of tasks to run different resources for the effective implementation and for receiving the expected result.

Self-Healing: Different instances of an application are allowed to run virtual machines and failure of instances which are handled repeatedly.

Safety-bag checks: The command does not meet the safety properties and likely to vulnerable are blocked.

S-Guard: This is less turbulent to normal stream processing and is based on rollback recovery.

Retry: This retires the failed task on the identical resource to be implemented repeatedly.

Task Resubmission: The task is resubmitted either to the similar or different resource for execution whenever a failed task is detected

Timing Check: This technique with critical function can be performed by watch dog.

Rescue workflow: It facilitates the workflow to continue until it becomes unimaginable to move forward without catering the failed task.

Software Rejuvenation: It allows the frequent reboots of the system. It assists the system to clean start and fresh start.

Preemptive Migration: This is regularly monitored and analyzed using feedback-loop control appliance.

Masking: A new state is recognized as transformed state after employment of error recovery. If this process applied thoroughly in the absence of effective error provide the user error masking.

Reconfiguration: Faulty element form the system is removed.

Resource Co-allocation: Resource is allocated for execution of task.

User specific exception handling: User defines the treatment for a task on its failure.

Based upon these techniques a number of models are implemented. Table 1 summarizes different models based on protection against fault and procedure.

3 Framework
A general framework for a fault tolerant system is given below that consist of different modules with different special tasks:
Table 1. Comparison among various models based on protection against the type of fault, and procedure [3].

<table>
<thead>
<tr>
<th>Model no</th>
<th>Model name</th>
<th>Protection against type of fault</th>
<th>Applied procedure for tolerate the fault</th>
</tr>
</thead>
</table>
| M1       | AFTRC      | Reliability                     | 1. Delete node depending on their reliability  
2. Back word recovery with the help of checkpointing |
| M2       | LLFT       | Crash-cost, trimming fault      | Replication                               |
| M3       | FTWS       | Dead line of work flow          | Replication and resubmission of jobs       |
| M4       | FTM        | Reliability, availability, on demand service | Replication users application and in the case of replica failure use algorithm like gossip based protocol. |
| M5       | CANDY      | Availability                    | 1. It assembles the model components generated from IBD and STM according to allocation notation.  
2. Then activity SNR is synchronized to system SRN by identifying the relationship between action in activity SNR and state transition in system SRN. |
| M6       | VEGA-WARDEN| Usability, security, scaling    | 1. Two layer authentication and standard technical solution for the application. |
| M7       | FT-CLOUD   | Reliability, crash and value fault | 1. Significant component is determined based on the ranking.  
2. Optimal ft technique is determined. |
| M8       | MAGI-CUBE  | Performance, reliability, low storage cost | 1. Source file is encoded in then splits to save as a cluster.  
2. File recovery procedure is triggered if the original file is lost. |

Node monitoring module

It is equipped with the lm-sensors which is used to evaluate the system affecting parameters that is used for forecast as an attributes along with the prediction model that have been purposed in this study.

Failure predictor

Failure predictor module is run in each node. It uses the model trained using support vector machine algorithm to predict to failure. Model is trained by using the log captured in the past and with some seed values. The parameters captured by the node monitoring modules are used as input by the model which are used for predictions [10, 11].

Proactive fault tolerance policy

The objectives of this module is to decrease the influence of failure of the execution. The policies that will be implemented by proposed architecture are:
(a) Detect the addition node.
(b) Leave the unhealthy node
(c) Set the alarm to inform the administrator to take an action.

The log is maintained by noting the incident of happening of fault.

**Controller module**

Controller module is the one that implements the policies listed above. In every node controller module is installed. This node is responsible for the action to be performed by the node that is about to fail [12]. Once the fault in the system is predicted by the model the controller module takes an action based upon the policies and record the incident in the log table.

### 4 Method Development

The proposed system acts on following 3 steps:

(a) Capturing data

Data sensed through the lm-sensor are captured. These data are further used as an input.

(b) Monitoring system

Monitoring task is performed by the failure predictor module. This module is built by using Support Vector Machine

#### Support vector machine

Support vector machine is the one of the well-known supervised approaches. This is usually used for the classification purpose. Support Vector Machine divides the data provided by the hyper plane. Using Support vector model involves following two phases:

#### Training phase

As being the supervised learning Support Vector Machine first need to be trained. Different instances are captured from the log table and using seed value and are labelled in to two classes:

(a) Normal
(b) Fault

These instances are further used for training the model. Once the model is trained it is tested using different testing methods like cross-validation test. Optimal Hyper Plane separating Fault and Normal

**Deployment phase**

The trained and tested model is implemented and used for the purpose of prediction. Model takes the data captured in step a) and then predict the class for the data. If the predicted data class is found to be Fault the alarm is set.

**Handling the predicted fault**

Once the occurrence of fault is predicted the controller module takes an appropriate action based upon the policies set and maintain the log.

### 5 Conclusion

Fault tolerance is popular research domain in security and networking including cloud computing. Fault tolerance is significant issue that required to maintain the stability of the system. Early detection of the fault helps to minimize the risk associated with the fault. Machine learning algorithm has been found to be milestone for the purpose of prediction. The power of machine learning algorithm like Support Vector Machine can be implemented in the cloud. Here in this paper a conceptual view to implement support vector machine for the purpose of proactive detection of fault has been discussed. This task can be further improved by performing the experiment and comparing with other machine learning algorithms like Naïve Bayes, Logistic Regression. It can also be tweaked by using different parameters.

### References


