International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

PREDICTION OF TASK FAILURE IN CLOUD DATA CENTERS USING DEEP LEARNING

[1] Dr. B. Sateesh Kumar Sir, [2] P.Alekhya

[1]Professor (Ph.d from the JNTUH) Department Of CSE

[2]M. Tech, Department Of CSE, (21JJ1D5804)

Sateeshbkumar@jntuh.ac.in, alekhyapolsani562@gmail.com

Abstract: A large-scale cloud data center needs to provide high service reliability and availability with low failure occurrence probability. However, current large-scale cloud data centers still face high failure rates due to many reasons such as hardware and software failures, which often result in task and job failures. Such failures can severely reduce the reliability of cloud services and also occupy huge amount of resources to recover the service from failures. Therefore, it is important to predict task or job failures before occurrence with high accuracy to avoid unexpected wastage. Many machine learning and deep learning based methods have been proposed for the task or job failure prediction by analyzing past system message logs and identifying the relationship between the data and the failures. In order to further improve the failure prediction accuracy of the previous machine learning and deep learning based methods, in this paper, we propose a failure prediction algorithm based on multi-layer Bidirectional Long Short Term Memory (Bi-LSTM) to identify task and job failures in the cloud. The goal of Bi-LSTM prediction algorithm is to predict whether the tasks and jobs are failed or completed. The trace-driven experiments show that our algorithm outperforms other state-of-art prediction methods with 93% accuracy and 87% for task failure and job failures respectively.

Index Terms: Cloud data centers and deep learning.

INTRODUCTION

Nowadays, cloud computing service has been wildly used because it provides high reliability, resource saving and also on-demand services. The cloud data centers include processors, memory units, disk drives, networking devices, and various types of sensors that support many applications (i.e., jobs) from users. The users can send requests such as store data and run applications to the cloud. Each cloud data center is composed with physical machines (PMs) and each PM can support a set of virtual machines (VMs). The tasks that are sent from users are processed in each VM. Such a large scale cloud data center can host hundreds of thousands of servers which often run tons of applications and receive work requests every second from users all over the world. A cloud data center with such heterogeneity and intensive workloads may sometimes be vulnerable to different types of failures (e.g., hardware, software, disk failures).

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

Take software failures as an example, Ya-hoo Inc. and Microsoft's search engine, Bing, crashed for 20 mins in January 2015, which cost about $9000 per minute to reboot the system. Previous research found that hardware failure, especially disk failure, is a major contributing factor to the outages of cloud services. These many different types of failures will lead to the application running failures. Thus, accurate prediction for the occurrence of application failures beforehand can improve the efficiency of recovering the failure and application running.

A job is comprised of one or more tasks, each of which is accompanied by a set of resource requirements. A job fails when one of its tasks fails. The previous works [3], [7]–[10] use statistical and machine learning approaches such as Hidden Semi-markov Model (HSMM) and Support Vector Machine (SVM) to predict the task and job failures in cloud data centers. They use CPU usage and memory usage, unmapped page cache, mean disk I/O time and disk usage as inputs and the task failure or job failure as the output. However, HSMM and SVM assume that all their inputs are stationary and independent of each other which are not true in the cloud data centers. Thus, they cannot handle the sequence data or high dimensional data, in which data in time points or different features may be dependent to each other. In the cloud data centers, the input features and noisy data are diverse in nature and have dependencies on the past events. Thus HSMM and SVM can't handle the failure prediction in cloud data centers.

LITERATURE REVIEW

Mina Sedaghat et. al: In large scale data centers, a single fault can lead to correlated failures of several physical machines and the tasks running on them, simultaneously. Such correlated failures can severely damage the reliability of a service or a job. This paper models the impact of stochastic and correlated failures on job reliability in a data center. We focus on correlated failures caused by power outages or failures of network components, on jobs running multiple replicas of identical tasks [2]. We present a statistical reliability model and an approximation technique for computing a job's reliability in the presence of correlated failures. In addition, we address the problem of scheduling a job with reliability constraints. We formulate the scheduling problem as an optimization problem, with the aim being to achieve the desired reliability with the minimum number of extra tasks. We present a scheduling algorithm that approximates the minimum number of required tasks and a placement to achieve a desired job reliability. We study the efficiency of our algorithm using an analytical approach and by simulating a cluster with different failure sources and reliabilities. The results show that the algorithm can effectively approximate the minimum number of extra tasks required to achieve the job's reliability.

Thanyalak Chalermarrewong et. al [3]: This paper proposes a framework for online failure prediction of data centers. A data center often has a high failure rate as it features a number of servers and components. Moreover, long running applications and intensive workloads are common in such facilities. Performance of the system depends on the availability of the machines, which can be easily compromised if failure cannot be handled gracefully. The main idea of this paper is to create an effective prediction model focusing on hardware failure. Accurate prediction may enhance the overall system performance. In this work, we employ two methods, namely, ARMA (Auto Regressive Moving Average) and Fault Tree Analysis. Experiments were then performed on a simulated cluster built based on Simi's platform. The results show prediction accuracy of 97%, which is very high. We thus believe that our framework is practical and can be adapted to use in data centers in the future.

Subrata Mitra et. al: With the explosion of data in applications all around us, erasure coded storage has emerged as an attractive alternative to replication because even with significantly lower storage overhead, they provide better reliability against data loss. Reed-Solomon code is the most widely used erasure code because it provides maximum reliability for a given storage overhead and is flexible in the choice of coding parameters that determine the achievable reliability. However, reconstruction time for unavailable data becomes prohibitively long mainly because of network bottlenecks. Some proposed solutions either use additional storage or limit the coding parameters that can be used. In this paper [4], we propose a novel distributed reconstruction technique, called Partial Parallel Repair (PPR), which divides the reconstruction operation to small partial operations and schedules them on multiple nodes already involved in the data reconstruction. Then a distributed protocol progressively combines these partial results to reconstruct the unavailable data blocks and this technique reduces the network pressure. Theoretically, our technique can complete the network transfer in $\lceil(\log 2(k + 1))\rceil$ time, compared to k time needed for a (k, m) Reed-Solomon code. Our experiments show that PPR reduces repair time and degraded read time significantly. Moreover, our technique is compatible with existing erasure codes and does not require any additional storage overhead. We demonstrate this by overlaying PPR on top of two prior schemes, Local Reconstruction Code and Rotated Reed-Solomon code, to gain additional savings in reconstruction time.

Haoyu Wang et. al [5]: In a modern cloud datacenter, a cascading failure will cause many Service Level Objective (SLO) violations. In a cascading failure, when a set of physical machines (PMs) in a failure domain are failed, their workloads are transferred to the PMs in another failure domain to continue. However, the new domain receiving additional workloads may become overloaded due to the resource oversubscription feature in the cloud, which easily leads to domain failures and subsequent workload transfer to other domains. This process repeats and a cascading failure is created finally. However, few previous methods can effectively handle the cascading failures. To handle this problem, we propose a Cascading Failure Resilience System (CFRS), which incorporates three methods: Overload-Avoidance VM Reassignment (OAVR), VM backup set placement (VMset) and Dynamic Oversubscription Ratio Adjustment (DOA). The experiments in trace-driven simulation show that CFRS outperforms other comparison methods in terms of the number of domain failures, the number of failed PMs and the number of SLO violations.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
IJARST
ISSN: 2457-0362

Haiying Shen et. al: With the rapid development of web applications in datacenters, network latency becomes more important to user experience. The network latency will be greatly increased by incast congestion, in which a huge number of requests arrive at the front-end server simultaneously. Previous incast problem solutions usually handle the data transmission between the data servers and the front-end server directly, and they are not sufficiently effective in proactively avoiding incast congestion. To further improve the effectiveness, in this paper [12], we propose a Proactive Incast Congestion Control system (PICC). Since each connection has bandwidth limit, PICC novelly limits the number of data servers concurrently connected to the front-end server to avoid the incast congestion through data placement. Specifically, the front-end server gathers popular data objects (i.e., frequently requested data objects) into as few data servers as possible, but without overloading them. It also re-allocates the data objects that are likely to be concurrently or sequentially requested into the same server. As a result, PICC reduces the number of data servers concurrently connected to the front-end server (which avoids the incast congestion), and also the number of connection establishments (which reduces the network latency). Since the selected data servers tend to have long queues to send out data, to reduce the queuing latency, PICC incorporates a queuing delay reduction algorithm that assigns higher transmission priorities to data objects with smaller sizes and longer queuing times. The experimental results on simulation and a real cluster based on a benchmark show the superior performance of PICC over previous incast congestion problem solutions.

ALGORITHM

In this we used algorithms like –

Random Forest: Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Decision Tree: Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable.

KNN: K Nearest Neighbour is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure. It is mostly used to classifies a data point based on how its neighbours are classified.

Support Vector Machine: Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

Voting Classifier: A voting classifier is a machine learning estimator that trains various base models or estimators and predicts on the basis of aggregating the findings of each base estimator. The aggregating criteria can be combined decision of voting for each estimator output.

CNN: A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data. There are other types of neural networks in deep learning, but for identifying and recognizing objects, CNNs are the network architecture of choice.

LSTM: Long short-term memory (LSTM) is an artificial neural network used in the fields of artificial intelligence and deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. Such a recurrent neural network (RNN) can process not only single data points (such as images), but also entire sequences of data (such as speech or video).

BiLSTM: Bidirectional Long Short-Term Memory (BiLSTM) In general time series processing, LSTM often ignores future information. BiLSTM uses two separate hidden. layers to process series data in forward and reverse directions on the basis of LSTM, connecting the two hidden.

RNN: A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes can create a cycle, allowing output from some nodes to affect subsequent input to the same nodes. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. Recurrent neural networks are theoretically Turing complete and can run arbitrary programs to process arbitrary sequences of inputs.

ARCHITECTURE

The system architecture proposed in the image consists of three main components: data collection, model training, and failure prediction. Data is collected from various nodes in a WDM MESH network and processed using double exponential smoothing to identify anomalies. The processed data is then used to train an SVM model to distinguish between normal and failure states. Once trained, the SVM model is deployed to predict the failure status of the network nodes in real-time. The overall system aims to provide risk-aware protection for the network by proactively identifying and mitigating potential failures.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

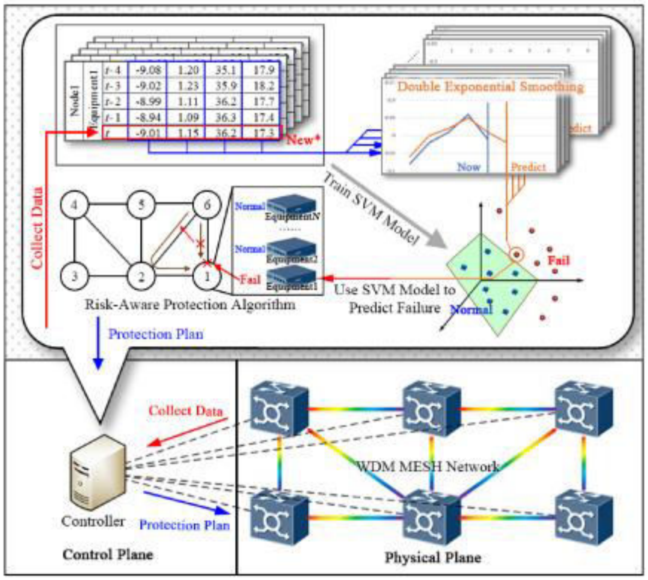

Fig.1: Model architecture

COMPARISON TABLE

Table.1: A summary of Disease Classification Model Based on Multi-Modal Feature Fusion

S.No	Title	Author/ Reference	Method/Algorithm implemented	Disadvantage	Advantage
1	DieHard: Reliable Scheduling to Survive Correlated Failures in Cloud Data Centers	Mina Sedaghat et. al., [2]	The algorithm estimates the minimum number of extra tasks needed by optimizing task scheduling and placement to achieve desired job reliability in the presence of correlated failures.	The algorithm may not handle highly complex failure correlations or dynamically changing failure patterns efficiently, potentially leading to suboptimal reliability and increased computational	It effectively approximates the minimum number of additional tasks required, optimizing job reliability under correlated failures, and balances task scheduling and placement to maintain service integrity.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

				overhead in large-scale data centers.	
2	Failure prediction of data centers using time series and fault tree analysis	Thanyalak Chalermarrewong et. al., [3]	ARMA for time-series prediction of hardware failures using historical data, then integrate Fault Tree Analysis to identify potential failure causes. Simulate on Simi's platform to validate prediction accuracy.	ARMA may struggle with non-linear failure patterns and dynamic workloads, while Fault Tree Analysis can become complex with many failure modes, potentially reducing scalability and adaptability in diverse data center environments.	Achieves high prediction accuracy of 97% by combining ARMA's time-series analysis with Fault Tree Analysis. This integrated approach effectively anticipates hardware failures, enhancing system reliability and performance in data centers.
3	Partial-parallel-repair (ppr): a distributed technique for repairing erasure coded storage	Subrata Mitra et. al., [4]	Divide reconstruction into small tasks, distribute them across nodes, and combine results using a distributed protocol. This reduces network load and repair time compared to traditional methods.	Increased complexity in coordination and protocol implementation may require more sophisticated management and synchronization, potentially complicating system maintenance.	Significantly reduces network pressure and repair time, compatible with existing erasure codes, and requires no additional storage overhead, enhancing efficiency and reliability in data reconstruction.

4	Approaches for resilience against cascading failures in cloud datacenters	H. Wang et. al., [5]	CFRS with OAVR for proactive VM reassignment, VMset for strategic backup placement, and DOA for dynamic oversubscription adjustment. Simulate to balance load, reduce domain failures, and mitigate SLO violations.	CFRS's complexity may increase operational overhead and require significant computational resources. Additionally, dynamic adjustments could introduce latency and impact performance if not finely tuned or monitored closely.	CFRS effectively minimizes cascading failures by distributing loads intelligently, optimizing resource usage, and adapting to overload scenarios. This leads to reduced domain failures and fewer SLO violations compared to traditional methods.
5	Proactive incast congestion control in a datacenter serving web applications	H. Shen et. al., [6]	PICC gathers popular data objects into fewer servers, reallocates objects with similar request patterns to the same server, limits concurrent connections, and prioritizes smaller, older data objects to reduce latency.	PICC may create imbalances in server load and potentially overload selected servers, leading to uneven performance and increased latency for less popular or infrequently accessed data objects.	PICC effectively reduces incast congestion and connection establishments, minimizing network latency and improving overall user experience by proactively managing data placement and queuing priorities for popular data objects.

SUMMARY

Several techniques for improving reliability and performance in data centers have been proposed. One method optimizes task scheduling to enhance job reliability despite correlated failures, though it may struggle with complex or dynamic failure patterns. Another approach combines ARMA time-series prediction with Fault Tree Analysis to achieve high accuracy in forecasting hardware failures, but it may not scale well with complex failure modes. A distributed repair technique reduces network load and repair time by dividing tasks and using distributed protocols, though it introduces coordination complexity. Additionally, a proactive method addresses cascading failures through intelligent load distribution and resource optimization, while another technique controls incast congestion by managing data placement and connection priorities, reducing latency but potentially leading to server imbalances.

CONCLUSION

In cloud data centers, high service reliability and availability are crucial to application QoS. In this paper, we proposed a failure prediction model multi-layer Bidirectional LSTM (called Bi-LSTM). Bi-LSTM can more accurately predict the termination statuses of tasks and jobs using Google cluster trace compared with previous methods. In our method, we first input the data into forward state and backward state in order to adjust the weight of both closer and further input features. We then find that the further input features is essential to achieving high prediction accuracy. Secondly, in the experiments, we compare Bi-LSTM with other comparison methods including statistical, machine learning and deep learning based methods and evaluate the performance with three metrics: accuracy and F1 score, receiver operating characteristic and time cost overhead. The results show that we achieved 93% accuracy in task failure prediction and 87% accuracy in job failure prediction. We also achieved 92% F1 score in task failure prediction and 86% F1 score in job failure prediction. Our prediction method Bi-LSTM also have low FPR which can also indicate the proactive failure management based on prediction results become more effective. We also observe that the time cost overhead for Bi-LSTM is almost the same compared with RNN and LSTM, which means Bi-LSTM can achieve higher prediction performance with no further time cost.

REFERENCES

[1] https://techcrunch.com/2015/01/02/following-bing-coms-brief-outagesearch-yahoo-com-goes-down-too/, [Accessed in APR 2019].

[2] M. Sedaghat, E. Wadbro, J. Wilkes, S. De Luna, O. Seleznjev, and E. Elmroth, "Diehard: reliable scheduling to survive correlated failures in cloud data centers," in 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2016.

[3] T. Chalermarrewong, T. Achalakul, and S. See, "Failure prediction of data centers using time series and fault tree analysis," in 2012 IEEE 18th International Conference on Parallel and Distributed Systems, 2012.

[4] S. Mitra, M. Ra, and S. Bagchi, "Partial-parallel-repair (ppr): a distributed technique for repairing erasure coded storage," in Proceedings of the eleventh European conference on computer systems, 2016.

[5] H. Wang, H. Shen, and Z. Li, "Approaches for resilience against cascading failures in cloud datacenters," in Proc. of ICDCS, 2018.

[6] H. Wang and H. Shen, "Proactive incast congestion control in a datacenter serving web applications," in Proc. of INFOCOM, 2018.

[7] R. Baldoni, L. Montanari, and M. Rizzuto, "On-line failure prediction in safety-critical systems," Future Generation Computer Systems, 2015.

[8] Y. Zhao, X. Liu, S. Gan, and W. Zheng, "Predicting disk failures with hmm-and hsmm-based approaches," in Industrial Conference on Data Mining, 2010.

[9] J. Murray, G. Hughes, and K. Kreutz-Delgado, "Machine learning methods for predicting failures in hard drives: A multiple-instance application," Journal of Machine Learning Research, 2005.

[10] I. Fronza, A. Sillitti, G. Succi, M. Terho, and J. Vlasenko, "Failure prediction based on log files using random indexing and support vector machines," Journal of Systems and Software, 2013.

[11] Clickstream Data Prediction Using Random Forestclassifier, IJSR, ISBN No.09762876, pp.29-32, March, 2018

[12] Comparison of Methods of Storing and Protecting Information in the Cloud, IJARCSSE, ISBN No.2277128X, Vol No.7, Issue No.1, pp.117-125, January, 2017