Section A-Research paper
ISSN 2063-5346

Task Failure Prediction in Cloud Data Centers Using Deep Learning

Mrs.R.G.Vyshnavi
Assistant Professor ,Department of IT
Sridevi Women's Engineering College,Telangana
swecrgv@gmail.com

Basam Meghana
BTech Student ,Department of IT
Sridevi Women's Engineering College,Telangana
basammeghana06@gmail.com

Bussa Upassana
BTech Student ,Department of IT
Sridevi Women's Engineering College,Telangana
upassana.b03@gmail.com

Bollu Akshitha
BTech Student ,Department of IT
Sridevi Women's Engineering College,Telangana
bolluakshitha@gmail.com

ABSTRACT: A large-scale cloud data centre must deliver high service dependability and availability while minimising failure incidence. However, modern large-scale cloud data centres continue to have significant failure rates owing to a variety of factors, including hardware and software faults, which often result in task and job failures. Such failures may substantially degrade the dependability of cloud services while also using a large amount of resources to restore the service. To reduce unexpected waste, it is critical to forecast task or job failures with high accuracy before they occur. Many machine learning and deep learning-based approaches for task or job failure prediction have been presented, which include examining previous system message logs and detecting the link between the data and the failures. In this research, we present a failure prediction technique based on multi-layer Bidirectional Long Short Term Memory (Bi-LSTM) to detect task and job failures in the cloud, in order to enhance the failure prediction accuracy of prior machine learning and deep learning-based approaches. The purpose of the Bi-LSTM prediction algorithm is to anticipate whether tasks and jobs will be completed or unsuccessful. Our approach beats existing state-of-the-art prediction algorithms in trace-driven tests, with 93% accuracy for task failure and 87% accuracy for job failures, respectively.

Keywords *– Cloud datacenters and deep learning.*

1. INTRODUCTION

Cloud computing services are becoming widely utilised because they offer high dependability, resource savings, and on-demand services. Cloud data centres include processors, memory units, disc drives, networking devices, and different kinds of sensors that serve a wide range of user applications (i.e., jobs). Users may submit requests to the cloud, such as storing data and running apps. Each cloud data centre is made up of physical machines (PMs), each of which may serve a group of virtual machines (VMs). The tasks supplied by users are handled in each VM. A large-scale cloud data centre may house hundreds of thousands of computers, which often operate hundreds of apps and get work requests from people all over the globe

every second. A cloud data centre with such heterogeneity and intense workloads may be subject to several forms of failures at times (e.g., hardware, software, disc failures). Take software failures as an example: in January 2015, Yahoo Inc. and Microsoft's search engine, Bing, collapsed for 20 minutes, costing nearly $9000 per minute to restart the system. Previous studies found that hardware failure, particularly disc failure, is a primary cause of cloud service disruptions. These many forms of problems will result in application execution failures. Thus, effective prediction of application failures ahead of time may enhance the efficiency of recovering the failure and keeping the programme operational.

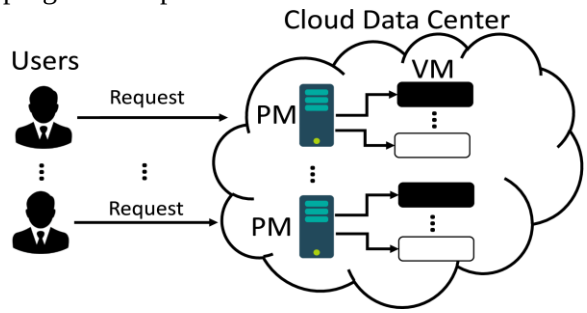

Fig.1: Example figure

A job is made up of one or more tasks, each with its own set of resource needs. When one of a job's tasks fails, the job fails. Previous research [3, 7–13] forecast task and job failures in cloud data centres using statistical and machine learning methodologies such as Hidden Semi-markov Model (HSMM) and Support Vector Machine (SVM). They take CPU and memory consumption, unmapped page cache, mean disc I/O time, and disc usage as inputs and output task or job failure. HSMM and SVM, on the other hand, presume that all of their inputs are fixed and independent of one another, which is not the case in cloud data centres. As a result, they are unable to handle sequence data or high-dimensional data in which data in time points or distinct attributes may be reliant on one another. The input characteristics and noisy data in cloud data centres are varied in type and are dependent on previous occurrences. As a result, HSMM and SVM cannot handle failure prediction in cloud data centres.

2. LITERATURE REVIEW

Diehard: reliable scheduling to survive correlated failures in cloud data centers:

A single malfunction in a big data centre might cause linked failures of numerous physical machines and the processes operating on them at the same time. Such connected failures may seriously undermine the dependability of a service or activity. The influence of random and linked failures on job dependability in a data centre is modelled in this research. We concentrate on linked failures induced by power outages or network component failures, and on jobs that perform numerous clones of identical tasks. In the case of correlated failures, we provide a statistical reliability model and an approximation approach for calculating work reliability. We also examine the issue of scheduling a work with dependability restrictions. We structure the scheduling issue as an optimization problem, with the goal of achieving the specified dependability with the fewest additional jobs. We provide a scheduling technique that approximates the lowest number of tasks needed and a placement to achieve the specified job dependability. We investigate our algorithm's efficiency using an analytical method and by simulating a cluster with various failure causes and reliabilities. The findings demonstrate that the algorithm can successfully estimate the lowest number of additional jobs necessary to ensure work dependability.

Failure prediction of data centers using time series and fault tree analysis

This research presents a methodology for online data centre failure prediction. Because of the amount of servers and components, a data centre often has a high failure rate. Long-running programmes and heavy workloads are also

prevalent in such facilities. The system's performance is dependent on the availability of the machines, which may be easily jeopardised if failure is not handled graciously. The primary goal of this article is to develop an effective hardware failure prediction model. Prediction accuracy may improve overall system performance. We use two approaches in this paper: ARMA (Auto Regressive Moving Average) and Fault Tree Analysis. The experiments were then carried out on a virtual cluster built using Simi's platform. The findings reveal a very high prediction accuracy of 97%. As a result, we feel that our architecture is realistic and that it may be extended for future usage in data centres.

Partial-parallel-repair (ppr): a distributed technique for repairing erasure coded storage

With the growth of data in applications all around us, erasure coded storage has arisen as an appealing alternative to replication because, although having a substantially smaller storage overhead, it provides higher data loss resilience. The Reed-Solomon code is the most extensively used erasure code because it delivers the most reliability for a given storage overhead and is versatile in terms of the coding parameters that affect the attainable reliability. However, due to network constraints, the reconstruction time for inaccessible data becomes unreasonably lengthy. Some suggested methods either utilise more storage or restrict the number of coding parameters that may be employed. In this research, we present Partial Parallel Repair (PPR), a new distributed reconstruction approach that breaks the reconstruction task into discrete partial operations and schedules them on many nodes already participating in the data reconstruction. Then, a distributed protocol gradually integrates these partial findings to rebuild the missing data blocks, reducing network congestion. In theory, our approach may finish the network transfer in (log2(k + 1)) time, as opposed to the k time required by a (k, m) Reed-Solomon code. Our results reveal that PPR considerably decreases repair time and impaired read time. Furthermore, our method is compatible with current erasure codes and requires no extra storage overhead. We show this by superimposing PPR on top of two previous techniques, Local Reconstruction Code and Rotated Reed-Solomon code, to get further time savings during reconstruction.

Approaches for resilience against cascading failures in cloud datacenters

A cascading failure in a contemporary cloud datacenter will result in several Service Level Objective (SLO) breaches. When a group of physical machines (PMs) in one failure domain fail, their workloads are shifted to PMs in another failure domain to continue. However, owing to the cloud's resource oversubscription capability, the new domain receiving extra workloads may become overloaded, resulting in domain failures and subsequent workload transfer to other domains. This practise is repeated until a cascade failure occurs. However, few prior techniques have been shown to efficiently manage cascading failures. To address this issue, we suggest a Cascading Failure Resilience System (CFRS) that integrates three methods: overload-avoidance, overload-resilience, and overload-resilience. Dynamic Oversubscription Ratio Adjustment, VM Reassignment (OAVR), and VM Backup Set Placement (VMset) (DOA). The results of the trace-driven simulation trials reveal that CFRS surpasses alternative comparison approaches in terms of the amount of domain failures, failed PMs, and SLO violations.

Proactive incast congestion control in a datacenter serving web applications

Due to the fast growth of web applications in datacenters, network latency is becoming more critical to user experience. Incast congestion, which occurs when a large number of requests

arrive at the front-end server at the same time, will significantly increase network latency. Previous incast issue solutions often handled data transfer directly between data servers and front-end servers, and therefore were ineffective in proactively avoiding incast congestion. In this study, we present a Proactive Incast Congestion Control method to boost efficacy even more (PICC). Because each connection has a bandwidth restriction, PICC restricts the amount of data servers connected to the front-end server simultaneously to reduce incast congestion via data placement. The front-end server, in particular, aggregates popular data items (i.e., often requested data objects) into as few data servers as feasible while without overloading them. It also re-allocates data items that are expected to be queried simultaneously or sequentially in the same server. As a consequence, PICC minimises the number of data servers linked to the front-end server simultaneously (avoiding incast congestion) as well as the number of connection installations (which reduces the network latency). Because the chosen data servers often have large queues to transfer data, PICC includes a queuing delay reduction algorithm that allocates greater transmission priority to data items with smaller sizes and longer waiting durations. The experimental findings on simulation and a real cluster using a benchmark indicate that PICC outperforms earlier incast congestion issue solutions.

3. METHODOLOGY

However, the LSTM-based prediction algorithms have a few flaws. First, as input features, the approaches only evaluate CPU utilisation, memory usage, cache memory usage, mean disc I/O time, and disc usage. More input features may improve prediction accuracy even more. Second, the LSTM-based prediction model employed single-layer LSTM construction, which is incapable of handling various input characteristics as well as multi-layer construction. Third, input characteristics such as CPU consumption and memory usage are significantly connected over time in the cloud data centre. For a particular prediction time, the LSTM-based prediction model always assigns larger weights to data closer to the time and lower weights to data farther away from the time, with the premise that data further away from the time has less effect on the forecast. However, such settings cannot truly show the degree of influence since other data may still have a greater impact on the failure (e.g., failures in long term jobs). The performance may improve if the weights of data items are computed using the actual data trace. To achieve improved prediction accuracy, a new prediction model must be built to apply failure prediction in the cloud data centre.

Disadvantages:

1. However, existing large-scale cloud data centres continue to have significant failure rates owing to a variety of factors, including hardware and software faults, which often result in task and job failures.
2. Such failures may substantially affect the dependability of cloud services while also using a large amount of resources to restore the service.

To address these problems, we present in this study a failure prediction model called Bi-LSTM that is built on multi-layer Bidirectional LSTM. To begin with, Bi-LSTM includes more input characteristics than earlier techniques, such as job prioritisation, task resubmissions, and scheduling delay. Second, Bi-LSTM has a multi-layer structure that allows it to handle numerous input characteristics with greater precision. Multi-layer structure may minimise the number of parameters in computing functions while maintaining the same number of neurons, reducing calculation time. Third, rather than

simply assigning greater weights to data items closer to the specified time for prediction than data items farther away from the time, Bi-LSTM may calculate the weights of data items based on their genuine influence on the failure. We conduct a trace-driven failure prediction research utilising Google cluster trace and compare the performance of Bi-LSTM to various cutting-edge prediction techniques.

Advantages:

1. Our method identifies task and job failures with great precision.
2. We also notice that the time cost overhead for Bi-LSTM is about the same as for RNN and LSTM, implying that Bi-LSTM may achieve improved prediction performance with no additional time cost.

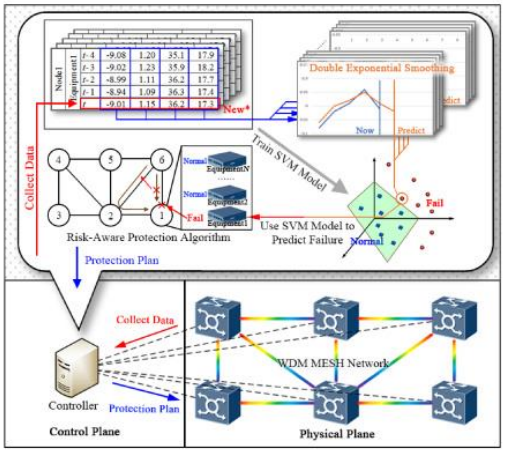

Fig.2: System architecture

MODULES:

To carry out the aforementioned project, we created the modules listed below.

- Data exploration: we will put data into the system using this module.
- Processing: we will read data for processing using this module.
- Splitting data into train & test: using this module data will be divided into train & test
- Model generation: Random Forest - Decision Tree - KMM - Support Vector Machine - Voting Classifier - CNN - CNN+LSTM - LSTM - BiLSTM - RNN - CNN with KFoldVaildation.
- User registration and login: Using this module will result in registration and login.
- User input: Using this module will provide input for prediction
- Prediction: the final projected value will be presented

4. IMPLEMENTATION

ALGORITHMS:

Random Forest: A Supervised Machine Learning Algorithm that is commonly utilised in Classification and Regression applications. It constructs decision trees from several samples and uses their majority vote for classification and average for regression.

Decision Tree: Decision trees use numerous methods to determine whether or not to divide a node into two or more sub-nodes. The development of sub-nodes promotes the homogeneity of the sub-nodes that arise. In other words, the purity of the node rises in relation to the target variable.

KNN: KNN is a basic algorithm that maintains all existing examples and classifies incoming data or cases based on a similarity metric. It is often used to classify a data point based on the classification of its neighbours.

SVM: Support Vector Machine (SVM) is a supervised machine learning technique that may be used for both classification and regression. Though we call them regression issues, they are best suited for categorization. The SVM algorithm's goal is to identify a hyperplane in an N-dimensional space that clearly classifies the input points.

Voting classifier: A voting classifier is a machine learning estimator that trains numerous base models or estimators and predicts based on the results of each base estimator. Aggregating

ISSN 2063-5346

criteria may be coupled voting decisions for each estimator output.

CNN: A CNN is a network architecture for deep learning algorithms that is primarily utilised for image recognition and pixel data processing applications. There are different forms of neural networks in deep learning, but CNNs are the network design of choice for identifying and recognising things.

LSTM: Long short-term memory (LSTM) is a kind of artificial neural network used in artificial intelligence and deep learning. Unlike traditional feedforward neural networks, LSTM has feedback connections. A recurrent neural network (RNN) of this kind may analyse not just single data points (such as photos), but also complete data sequences (such as speech or video).

BiLSTM: BiLSTM stands for Bidirectional Long Short-Term Memory (BiLSTM) In general, LSTM ignores future information in time series processing. BiLSTM processes series data in forward and reverse directions on the basis of LSTM, linking the two hidden layers.

RNN: A recurrent neural network (RNN) is a kind of artificial neural network in which node connections may form a cycle, enabling output from one node to influence future input to the same node. This enables it to display temporal dynamic behaviour. RNNs, which are derived from feedforward neural networks, can handle variable length sequences of inputs using their internal state (memory). As a result, they may be used for tasks like unsegmented, linked handwriting recognition or voice recognition. Recurrent neural networks are Turing complete in theory and may execute arbitrary algorithms to handle arbitrary sequences of inputs.

5. EXPERIMENTAL RESULTS

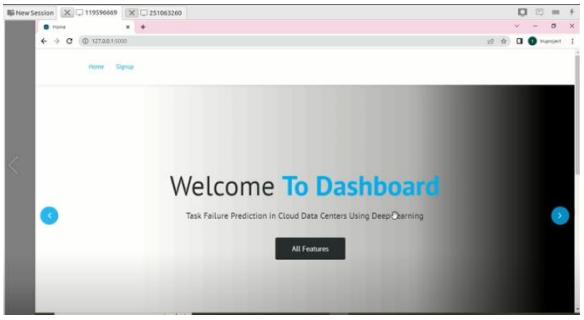

Fig.3: Home screen

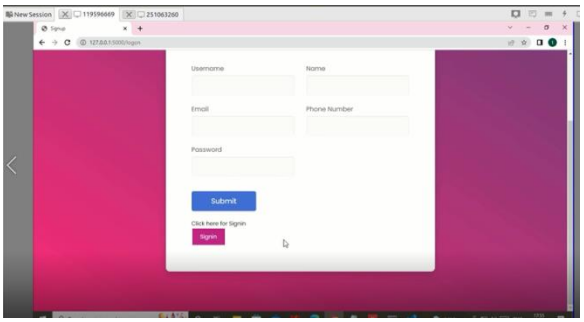

Fig.4: User registration

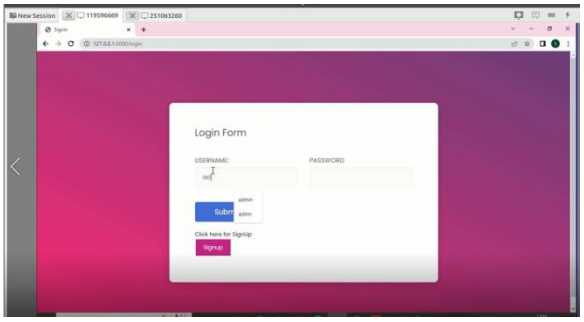

Fig.5: user login

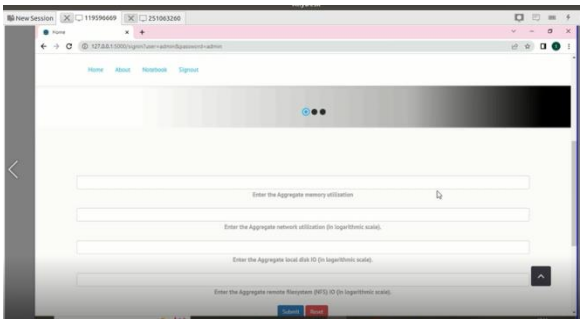

Fig.6: Main screen

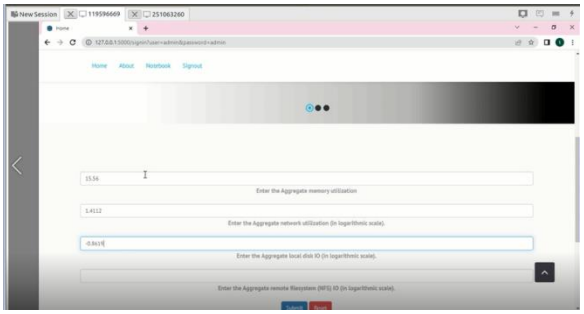

Fig.7: User input

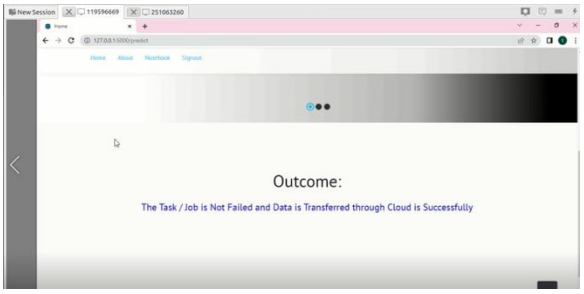

Fig.8: Prediction result

6. CONCLUSION

High service dependability and availability are critical to application QoS in cloud data centres. In this research, we introduced a multi-layer Bidirectional LSTM failure prediction model (called Bi-LSTM). When compared to prior approaches, Bi-LSTM can more reliably predict the termination states of tasks and jobs using Google cluster trace. In order to modify the weight of both closer and farther input characteristics, we first input the data into forward and backward states in our approach. We then discover that additional input characteristics are critical to getting high prediction accuracy. Second, in the tests, we compare Bi-LSTM to various comparison approaches, such as statistical, machine learning, and deep learning-based methods, and assess performance using three metrics: accuracy and F1 score, receiver operating characteristic, and time cost overhead. The findings reveal that we predicted task failure with 93% accuracy and job failure with 87% accuracy. We also got a 92% F1 in task failure prediction and an 86% F1 in job failure prediction. Our prediction approach Bi-LSTM has a low FPR, indicating that proactive failure management based on prediction findings is becoming increasingly successful. We also notice that the time cost overhead for Bi-LSTM is about the same as for RNN and LSTM, implying that Bi-LSTM may achieve improved prediction performance with no additional 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] Q. Guan, Z. Zhang, and S. Fu, "Ensemble of bayesian predictors and decision trees for proactive failure management in cloud computing systems," Journal of Communications, 2012.
[12] T. Pitakrat, D. Okanovic, A. van Hoorn, and L. Grunske, "Hora: ´ Architecture-aware online failure prediction," Journal of Systems and Software, 2018.
[13] S. Zhang, Y. Liu, Z. Meng, W.and Luo, J. Bu, P. Yang, S.and Liang, D. Pei, J. Xu, and Y. Zhang, "Prefix: Switch failure prediction in datacenter networks," Proceedings of the ACM on Measurement and Analysis of Computing Systems, 2018.
[14] C. Xu, G. Wang, X. Liu, D. Guo, and T. Liu, "Health status assessment and failure prediction for hard drives with recurrent neural networks," IEEE Transactions on Computers, 2016.
[15] Y. Cheng, H. Zhu, J. Wu, and X. Shao, "Machine health monitoring using adaptive kernel spectral clustering and deep long short-term memory recurrent neural networks," IEEE Transactions on Industrial Informatics, 2019.