Proposal Defense

VM Failure Prediction using Log Analysis for Proactive Fault Tolerance

Pratheek Senevirathne
19001622

Supervisor: Dr. Dinuni Fernando, Senior Lecturer (UCSC)
Co-supervisor: Dr. Dinal Herath, Security Expert (Obsidian Security, CA, USA)

Content

01. Introduction

Introduction - VMs and Failures

- A Virtual Machine (VM) is a software emulation of a physical computer that allows us to run several Operating Systems (OSs) independently on the same machine.
- Usually used to run an end-user application, or a service.
- VMs and all the other related software and hardware components are failure-prone.
- VM failure → User application failure
- VMs will fail if there is a failure in,
    - Hardware
    - Software
    - Network

Introduction - VM Fault Tolerance

- **Fault Tolerance:** The ability of a system to continue operating without any issues even in failure of its sub-components.
- VM Fault Tolerance approaches,
  - Reactive Fault Tolerance
  - Proactive Fault Tolerance
- Reactive approaches,
  - Reactive Migration
  - Checkpoint and Restart
  - Replication
- Proactive approaches,
  - Preemptive Migration
  - System Rejuvenation
  - Self-healing
- VM migration is one of the main FT approaches, and Google performs over 1 Million VM migrations per month!

Introduction - VM Migration

- Two main VM migration techniques based on “Lively-ness”,
  - Non-Live VM Migration
  - Live VM Migration
- **Non-Live Migration:** Migrate the VM by turning off the VM
- **Live Migration:** Migrate the VM while it is turned on and the applications are running.

VM Live Migration Animation

Introduction - Failures and Logs

- Focus is on VM failures due to hardware and software failures.
    - **Hardware:** Failures/faults related to Memory, CPU, or Secondary Storage.
    - **Software:** Failures/faults related to Host/Guest OS, Hypervisor, or User Applications.
- All of these failures generate logs (system log, application log, kernel log, etc.)
- These logs contain information about the system state and events, which can be used to predict VM failure.
- If we can predict the VM failure, we can use the migration technique to save the VM from failing by proactively moving it to a healthy server.

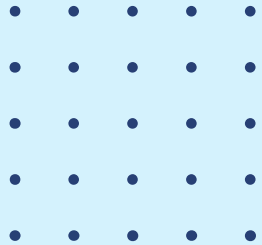
02. Motivation

Motivation - Significant research gap

- Researchers have attempted to achieve high VM availability by using node failure prediction and proactive migration.
- Only using physical server resource usage history data to train ML models for the failure prediction.
- Most of them have left out the most crucial part of any digital system that keeps track of the system state and the events, **the logs**.

Motivation - The Problem

- VM Live migration takes time, mainly depending on the VM RAM size and the available network bandwidth.
- VM failure should be predicted before the time it takes to migrate it.
- Most of the existing papers in this area have looked over this basic fact.
- Even though the failure prediction may be accurate, the VM may fail during migration due to late failure prediction.



03.
Related Work

Related Work

Virtual Machine Failure Prediction using Log Analysis (2021)

Nam, Hong, Yoo, et al. [6]

- Failure prediction of VNFs by analyzing logs.
- NLP technique which uses Word2Vec and a CNN model on processed log data.
- Failure prediction before 5 min to failure. Overall F1 Score: 0.67.

VM Failure Prediction with Log Analysis using BERT-CNN Model (2022)

Nam, Hong, Yoo, et al. [7]

- Failure prediction of VNFs by analyzing logs.
- Using Google BERT with a CNN model on processed log data.
- Failure prediction before 30 min to failure. Overall F1 Score: 0.74.

Related Work

Proactive Live Migration for VNFs using Machine Learning (2021)

Jeong, Van Tu, Yoo, et al. [8]

- "Paging-failure" prediction of vEPC by using VM resource usage info and log data using an LSTM model.
- Count of specific log lines instead of the actual logs.
- Successfully prevents long-term vEPC failures

MING
Microsoft Research (2018)

Lin, Hsieh, Dang, et al. [5]

- Failure prediction of CDC physical servers
- LSTM and Random forest models
- Server ranking system to rank all the servers by their failure-proneness
- First-ever production deployment

04.
Research Gap

Research Gap

- Primarily focused on utilizing physical machine resource usage history to predict failures.
- Overlooked the potential insights provided by VM and PM logs.
- Limited to studying specific VMs, such as VNFs.
- Implemented techniques are reactive.
- Migration time when predicting failures has been disregarded.

05. Research Questions

Research Questions

01. How to effectively utilize VM and PM logs for machine learning-based VM failure prediction to predict the failures ahead of time?

02. How to develop a generalized VM failure prediction approach using log analysis, enabling its applicability to a wide range of generic Linux-based VMs?

03. How can the timing of VM and PM failure prediction be optimized to ensure successful VM migration to a healthy PM, considering the total time required for the VM migration?

06. Objectives

Objectives

- To develop a **generalized ML-based** prediction approach that leverages key events and indicators present in VM and PM logs to **predict failures ahead of time** in a variety of Linux-based VMs.
- To make the prediction **time-aware** so that it considers the time required for migration to ensure successful VM migration to another physical machine.
- To **evaluate** the proposed prediction approach and compare its performance against existing techniques.

07.
Scope

Scope

In Scope

- VM failure prediction ahead of time by analyzing logs using a ML-based approach
- Online failure prediction
- VM Live Migration in QEMU-KVM on Ubuntu host OS
- LAN based VM migrations
- Implementation of a working prototype

Out of Scope

- Handling unexpected VM failure situations - No pre-failure logs
- Network-related failures - Migration is not effective

08.
Research Approach

Research Approach

- **Data collection:** No real-world Linux VM failure log data is available. Log data will be collected using simulated failures by fault injection.
- **Testbed setup:** Two physical servers interconnected with Gigabit ethernet.
- **Data processing:** The collected log corpus should be preprocessed in order to use them in a ML model.
- **ML Model(s) Development:** Each model can be individually developed and evaluated to select the best model.
- **Evaluation of ML model:** Evaluated using appropriate evaluation metrics. The models will also be compared with multiple datasets and multiple models.

Research Approach

- **Timing and Migration Analysis:** The timing aspect of the prediction in relation to VM migration should also be thoroughly analyzed to check for timely prediction of failures.
- **Prototype Implementation:** A working prototype will be implemented to integrate the developed VM failure prediction model with QEMU live migration.
- **Whole system evaluation:** The implemented prototype system will be evaluated through experiments conducted on the test-bed to assess the efficiency, and reliability in simulated failure scenarios.

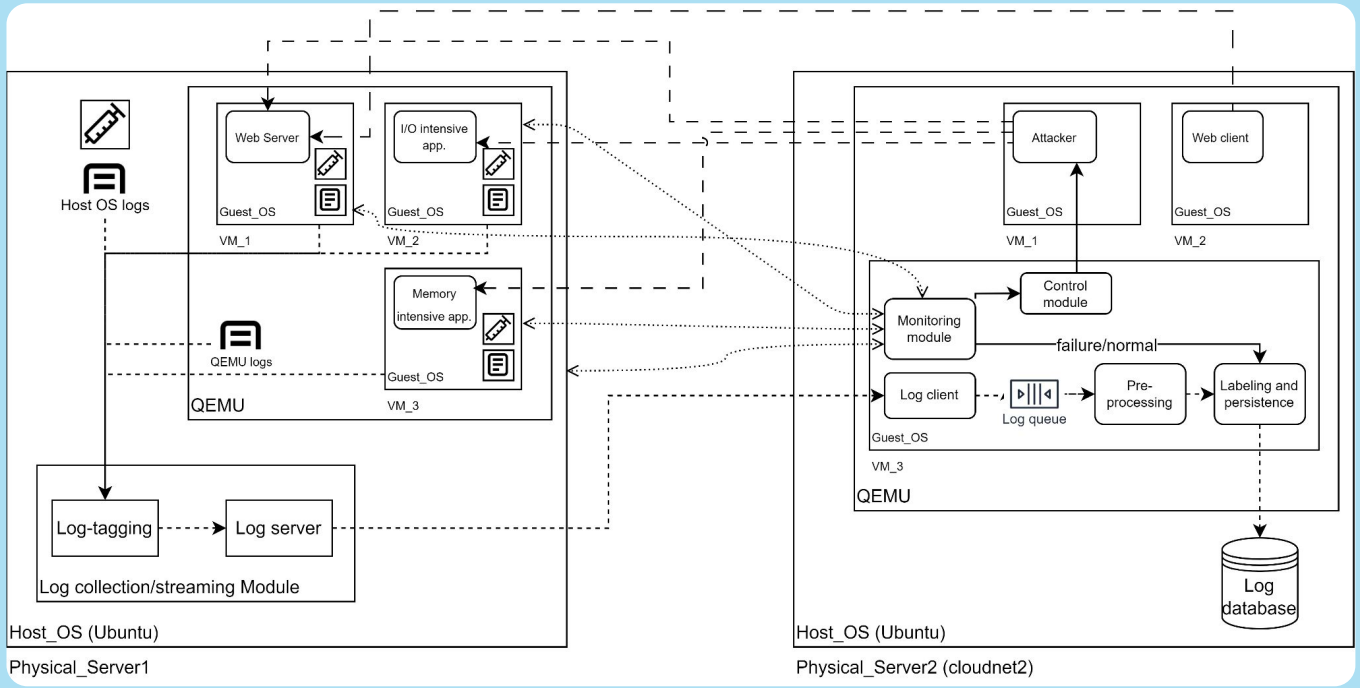


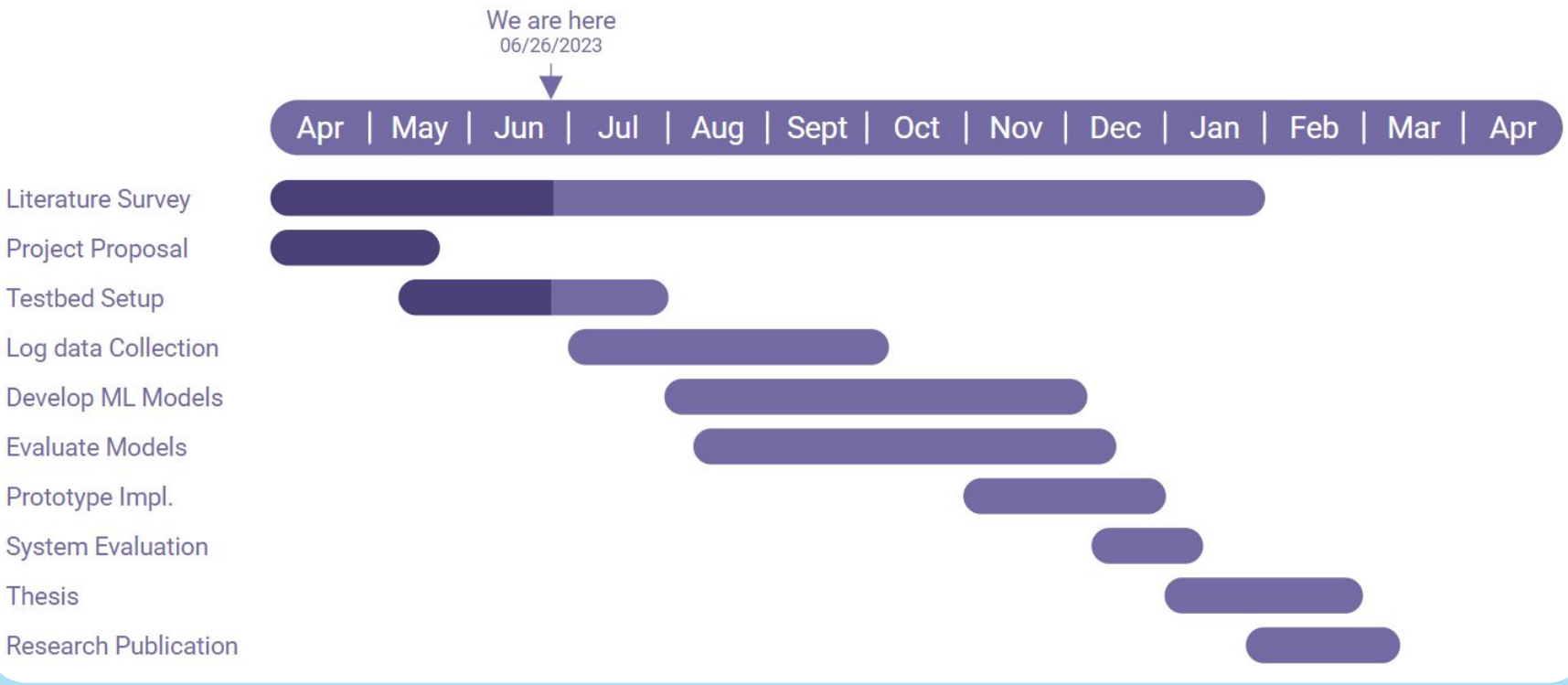
Figure 1: High-level architecture of the testbed

Research Method

- Planned to follow **Design science** research methodology.
- Creation of an innovative, purposeful artifact for a special problem domain.
- Novel research contribution:
  - Solve a problem that has not yet been solved
  - Provides a more effective solution
- Proper evaluation of the produced artifact
- Present the research effectively to both technology-oriented and management-oriented audiences.

09. Progress

Timeline


We are here
06/26/2023
Apr | May | Jun | Jul | Aug | Sept | Oct | Nov | Dec | Jan | Feb | Mar | Apr
Literature Survey
Project Proposal
Testbed Setup
Log data Collection
Develop ML Models
Evaluate Models
Prototype Impl.
System Evaluation
Thesis
Research Publication

Progress so far

- Setting up the testbed to collect log data.
- Got access to the log data set and the source code of the paper “Virtual Machine Failure Prediction using Log Analysis”
- Learning ML techniques like Neural networks, Deep learning, and PyTorch.

References

[1] R. Birke, I. Giurgiu, L. Y. Chen, D. Wiesmann, and T. Engbersen, "Failure analysis of virtual and physical machines: Patterns, causes and characteristics," in 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2014, pp. 1–12. doi: 10.1109/DSN.2014.18.

[2] C. Engelmann, G. R. Vallee, T. Naughton, and S. L. Scott, "Proactive fault tolerance using preemptive migration," in 2009 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing 2009, pp. 252–257. doi: 10.1109/PDP.2009.31.

[3] A. Polze, P. Tr ̈oger, and F. Salfner, "Timely virtual machine migration for proactive fault tolerance," in 2011 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, 2011, pp. 234–243. doi: 10.1109/ISORCW.2011.42.

[4] Nvidia. "Datacenter virtual gpu data migration animation." (2023), [Online]. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/solutions/vgpu-migration/data-center-virtual-gpu-data-migration-animation-625-ud.gif (visited on06/10/2023).

[5] Q. Lin, K. Hsieh, Y. Dang, et al., "Predicting node failure in cloud service systems," in Proceedings of the 2018 26th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering, 2018, pp. 480–490.

[6] S. Nam, J. Hong, J.-H. Yoo, and J. W.-K. Hong, "Virtual machine failure prediction using log analysis," in 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE, 2021, pp. 279–284.

[7] S. Nam, J.-H. Yoo, and J. W.-K. Hong, "Vm failure prediction with log analysis using bert-cnn model," in 2022 18th International Conference on Network and Service Management (CNSM), IEEE, 2022, pp. 331–337.

[8] S. Jeong, N. Van Tu, J.-H. Yoo, and J. W.-K. Hong, "Proactive live migration for virtual network functions using machine learning," in 2021 17th International Conference on Network and Service Management (CNSM), IEEE, 2021, pp. 335–339

[9] A. Ruprecht, D. Jones, D. Shiraev, et al., "Vm live migration at scale," in Proceedings of the 14th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, ser. VEE '18, Williamsburg, VA, USA: Association for Computing Machinery, 2018, pp. 45–56, isbn: 9781450355797. doi: 10.1145/3186411.3186415. [Online]. Available: https://doi.org/10.1145/3186411.3186415.

Thanks!

Appendix

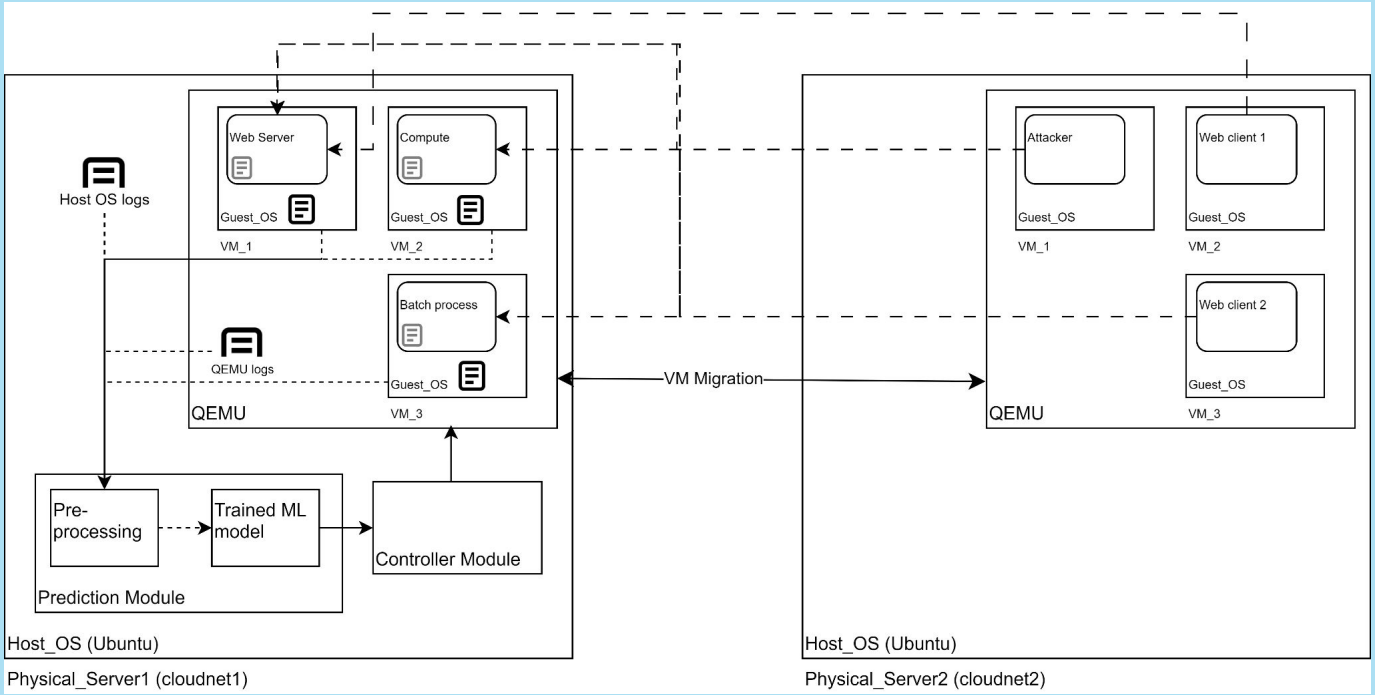
- Log data files:
  - syslog - /var/log/syslog
    - Generic system activity logs
  - bootlog - /var/log/boot.log
    - Booting related information and messages logged during system startup
  - dmesg logs - /var/log/dmesg
    - Captures the device status, hardware errors and other generic messages.
  - kernel logs - /var/log/kern.log
    - Information logged by the kernel
  - QEMU logs- /var/log/libvirt/qemu
    - Logs generated by QEMU

Appendix

- Fault injection:
  - Stress-NG - Overloading CPU, Memory
  - Apache bench - HTTP load testing
  - dmsetup - Inject HDD errors
  - scsi_debug - Linux HDD fault injection suite
  - mce-inject - Linux CPU fault injection suite
  - Chaos Mesh - Fault simulation and orchestrate fault scenarios
- Log collection and streaming:
  - R-syslog
  - Collectd

Appendix

- High-level architecture of final system [Tentative]:



Appendix

- ML model evaluation metrics:
  - Accuracy - Correct predictions / All predictions
  - Precision - True positives / (True positives + False positives)
  - Recall - True positives / (True positives + False negatives)
  - F1 score - 2 * (Precision * Recall) / (Precision + Recall)
- Log collection and streaming:
  - R-syslog
  - Collectd