Challenges and Experiences with MLOps for Performance Diagnostics in Hybrid-Cloud Enterprise Software Deployments

Amitabha Banerjee, Chien-Chia Chen, Chien-Chun Hung, Xiaobo Huang, Yifan Wang, Razvan Chevesaran
VMware Inc., Palo Alto, CA, USA

Abstract

This paper presents how VMware addressed the following challenges in operationalizing our ML-based performance diagnostics solution in enterprise hybrid-cloud environments: data governance, model serving and deployment, dealing with system performance drifts, selecting model features, centralized model training pipeline, setting the appropriate alarm threshold, and explainability. We also share the lessons and experiences we learned over the past four years in deploying ML operations at scale for enterprise customers.

1. Introduction

Operationalizing ML solutions is challenging across the industry as described in [1-2]. In addition, VMware faces several unique challenges to productize our ML-based performance diagnostics solution. As one of the top hybrid-cloud providers, VMware has the most large-enterprise customers who deploy our Software-Defined Datacenter (SDDC) stack in both their on-premises datacenters as well as VMware-managed public clouds. This unique position gives VMware an opportunity to collect telemetry data from a wide variety of hardware/software combinations all over the world. However, it also results in two challenges: **(1) Data governance**. VMware must comply with local data regulations where the SDDC deployments reside, private cloud or public cloud. **(2) Model deployment**. Enterprise customers often have a long and inconsistent cycle to update the software, typically once every year or two. It is impractical to couple ML model deployment with customer's inconsistent lengthy update cycles.

Moreover, developing ML-based solutions for performance diagnostics needs to address the following issues: **(1) Handling Performance drifts**. System performance changes across the time due to software/hardware updates or normal hardware degradation. **(2) Feature engineering**. The system performance depends on numerous factors, which constitutes a multivariate ML problem. Consequently, the selection of the appropriate features plays an important role in model performance, and often requires human intervention. **(3) Model training**. Our experience has found that models perform better when being trained with global data from all deployments sharing a common hardware type. Thus, a centralized ML pipeline that complies with data regulations is required. **(4) Sensitivity to alarm threshold:** Flooded false alarms lead to either alarm disabling or overwhelming product support. On the other hand, conservative alarm threshold may miss performance issues and lead to silent failures and hence costly manual investigation. It is therefore critical to identify the appropriate alarm threshold. **(5) Explainability**. Once an issue is detected and an alarm is fired, it is important to provide insights and recommendations as to how to remediate the issue. Otherwise, it can still lead to unnecessary product support overhead.

2. Solution Design and Deployment

We have been deploying ML solutions to detect and root-cause performance issues in two offerings: (1) VMware vSAN Performance Diagnostics, a product feature allowing enterprise customers to self-diagnose performance issues in their deployments [3], and (2) VMware Support Insight, a support facing feature that allows VMware support teams to proactively monitor performance issues across all product deployments in a dashboard [4]. Figure 1 describes the ML/Ops lifecycle by which these solutions work from an operational perspective.

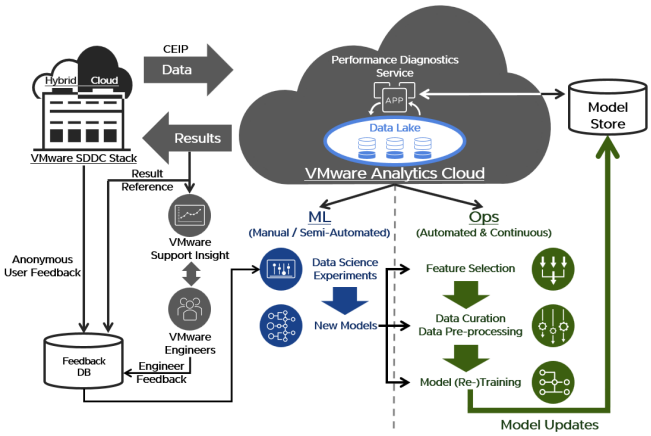

Figure 1. ML/Ops Lifecycle of Our Solution

2.1 ML/Ops Lifecycle

Data collection is done via VMware vCenter [5], the appliance managing a cluster of servers. For those customers who opted-in VMware Customer Experience Improvement Program (CEIP) [6], vCenter periodically transmits performance counters hourly to the VMware's centralized data platform, VMware Analytics Cloud (VAC) [7]. VAC stores and governs all the collected telemetry data and performs necessary placement, anonymization and encryption to comply with local data regulations. VAC also provides a run-time platform to deploy and run our ML-based performance diagnostics services. The results of the performance diagnostics are stored

in a key-value database in VAC from which they can be consumed either by the source vCenter, where the analysis in the form of charts, alarms, or recommendations are presented to the customers, or by support tools such as VMware Support Insight, where the support engineers monitor the results of multiple deployments simultaneously. Since the service runs within VMware's cloud infrastructure, its updates are decoupled from the customer's software update cycles, which allows a continuous deployment of new models [8]. VAC deployment addresses our challenges of data governance and model deployment.

2.2 ML/Ops Pipeline

We develop an ML/Ops pipeline to tackle the five challenges in our performance diagnostics use case. We develop a fully automated pipeline to continuously train new models based on the newly arrived batch of data. These models are maintained and version-controlled in the model store. The pipeline evaluates a large number of feature sets combining with various data pre-processing techniques and different ensembles of ML algorithms. The continuous training and deployment ensure our production model always learns the latest performance behaviors. Different models are trained for different hardware configurations. As an example, the model for a server with all-flash NVMe storage is different from that of a server with mixed magnetic disks and flash storage.

When deployed for performance diagnostics, the newest stable model specific to the hardware configuration of the system is used. If a rollback is desired, performance diagnostic would revert to the last-known-good model or any earlier model specified with certain version name.

The feature set candidates are first defined by product experts based on the specific performance issues to detect. Note that these feature sets are merely means to guide the pipeline to explore the feature space more efficiently, since there are easily thousands of thousands of performance metrics in a production SDDC. The pipeline also exploits several standard feature selection techniques such as PCA and autoencoder for each given set in addition to the human chosen ones, to achieve the best results [9-10].

Before feeding data to ML algorithms, the pipeline applies various data pre-processing techniques such as Z-transformation and Box-Cox transformation. We learned that ML algorithms might perform very differently with different data pre-processing techniques. For example, certain models prefer all dimensions to be in normal distribution, so Z-transformation works the best for this type of models. To provide explainability on top of the models' results, our pipeline employs a rule-based root-cause analysis engine to pinpoint the source of performance issue.

The rest of the pipeline includes the common ML training steps such as optimizing against a training set, applying boosting algorithms, and validating against a validation set to find the best performing model ensemble. Once a new model ensemble is trained and deployed in all three solutions outlined in Section 2.1, product performance experts, site reliability engineers, and solution engineers start consuming the output. These experts continuously monitor the latest model performance, review feedback given by service consumers, and examine new performance issues. This requires several manual or at best semi-automated steps to understand scenarios where the ML models are under-performing. To make this process more efficient, we developed a framework that enables performance perturbations in carefully orchestrated experiments and examine the behaviors from our models. We also leverage generic monitoring solution such as VMware Wavefront to monitor the performance of deployed models in conjunction with the feedback provided by the users [11]. As an example, we identified that our initial models were having high false positives in situations where very large IO size requests were being handled by vSAN because the ML models were overfitted towards the more common scenarios of small IO size requests. In such situations, we push new datasets to the Ops pipeline and evaluate if the changes improve the performance of the diagnostic service.

3. Lessons Learned

This paper describes our efforts spanning over four years in operationalizing an ML solution to detect and diagnose performance issues in our production deployed enterprise solutions. We learned several valuable lessons as follows:

(1) Continuous training and serving. Prior to building a pipeline, majority of the development time was spent on manual steps that are error-prone and time consuming. The ML/Ops pipeline automates our workflow and helps us churn models in hours, steps which earlier took us close to six months.

(2) Importance of a monitoring solution. Having a monitoring solution with great visualization helps us immensely in understanding why performance anomalies were being identified by an ML model and getting an intuitive understanding of how ML models work.

(3) Importance of automation: Continuous delivery of new models, noting improvements, and rollbacks when necessary are immensely valuable for production use cases. Delivering an ML framework for performance diagnostics in production necessitates a robust alarm threshold in a dynamically changing environment. It is hard to achieve the same without an efficient ML Ops lifecycle.

(4) Orchestrated experiment environment for model behavior validation. Having a framework that enables injection of the perturbed configuration allows us to quickly examine and verify whether the models react appropriately to a specific scenario. This mitigates the overhead with investigating the model issues in production environment.

4. References

[1] Z. Li, Y. Dang, “AIOps: Challenges and Experiences in Azure”, OpML 2019

[2] MLOps: Continuous delivery and automation pipelines in machine learning. https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning.

[3] vSAN Performance Diagnostics. https://kb.vmware.com/s/article/2148770.

[4] VMware Support Insight. https://storagehub.vmware.com/t/vsan-support-insight/.

[5] VMware vCenter. https://www.vmware.com/products/vcenter-server.html.

[6] VMware Customer Experience Improvement Program (CEIP). https://www.vmware.com/solutions/trustvmware/ceip.html.

[7] VMware Analytics Cloud (VAC). https://blogs.vmware.com/vsphere/2019/01/understanding-vsphere-health.html.’

[8] A. Banerjee and A. Srivastava, “A Cloud Performance Analytics Framework to Support Online Performance Diagnosis and Monitoring Tools”, in Proc. of the 2019 ACM/SPEC International Conference on Performance Engineering, pp. 151–158, Mumbai, India, April 2019.

[9] I.T. Jolliffe, "Principal Component Analysis", Second Edition, Springer (2002).

[10] D. H. Ballard, "Modular learning in neural networks," Proceedings of the sixth National conference on Artificial intelligence, July, 1987, Seattle, WA, U.S.A.

[11] VMware Wavefront. https://cloud.vmware.com/wavefront.