Nguyen, Tri; Truong, Linh

Demonstration Paper: Monitoring Machine Learning Contracts with QoA4ML

DOI:
10.1145/3447545.3451172

Published: 19/04/2021

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Nguyen, T., & Truong, L. (2021). *Demonstration Paper: Monitoring Machine Learning Contracts with QoA4ML*. 169-170. Poster session presented at ACM/SPEC International Conference on Performance Engineering, Rennes, France. https://doi.org/10.1145/3447545.3451172

Demonstration Paper: Monitoring Machine Learning Contracts with QoA4ML

Minh-Tri Nguyen
tri.m.nguyen@aalto.fi
Dept. of Computer Science, Aalto University
Espoo, Finland

Hong-Linh Truong
linh.truong@aalto.fi
Dept. of Computer Science, Aalto University
Espoo, Finland

ABSTRACT

Using machine learning (ML) services, both service customers and providers need to monitor complex contractual constraints of ML service that are strongly related to ML models and data. Therefore, establishing and monitoring comprehensive ML contracts are crucial in ML serving. This paper demonstrates a set of features and utilities of the QoA4ML framework for ML contracts.

CCS CONCEPTS

• **Software and its engineering → Software as a service orchestration system**;

KEYWORDS

Service contract, ML Serving, SLO/SLA, System monitoring

ACM Reference Format:
Minh-Tri Nguyen and Hong-Linh Truong. 2021. Demonstration Paper: Monitoring Machine Learning Contracts with QoA4ML. In *Companion of the 2021 ACM/SPEC International Conference on Performance Engineering (ICPE '21 Companion), April 19–23, 2021, Virtual Event, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3447545.3451172

1 MOTIVATION

Today, Machine Learning as a Service (MLaaS) has become a popular business model [1]. There is a strong demand for flexible ways to establish and monitor ML service contracts/agreements among multiple stakeholders, such as the ML customer, the ML provider, and the infrastructure provider. However, supports for ML service contracts have not been well researched. Commonly, ML service providers only allow customers to choose pre-defined contractual plans with a certain level of support and service quantity/quality [5] (e.g., connections/host, CPUs/instance, bandwidth, and security). The lack of tools for implementing and managing flexible ML contracts is one of the main challenges in ML serving. To fill the gap, we have developed the QoA4ML [5], a framework that supports the ML service contract specification and monitoring.

In this paper, we show that the developer, consumers, and providers can use QoA4ML features to establish a service contract comprising various ML-specific attributes without much effort. Given numerous monitoring probes, we will demonstrate ML contracts of two real ML services in real-time. QoA4ML can simplify ML contract specifications, detect ML contract violations, and support ML service compliance as well as elastic ML service management.

ICPE '21 Companion, April 19–23, 2021, Virtual Event, France

2 SPECIFYING & MONITORING CONTRACT

Figure 1 provides an overview of QoA4ML in an ML serving pipeline. Within QoA4ML, *probes*, *contracts*, *policies*, and *Observability Service* are key components. QoA4ML allows us to specify all the requirements and the service constraints in a service contract.

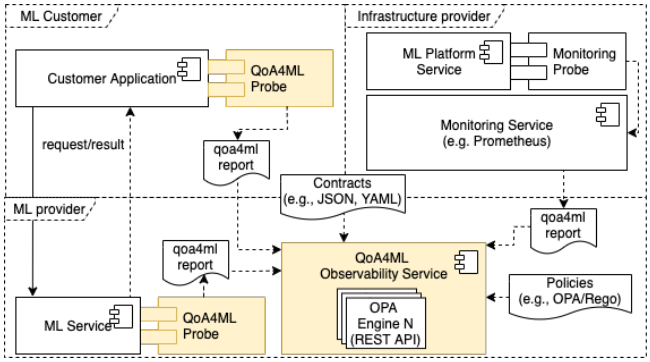

Figure 1: Integrating QoA4ML into an ML serving pipeline.

Specifying ML service contracts: Based on the requirements of both customers and providers, developers can define common service attributes as well as specify the ML-specific attributes, such as data quality and inference accuracy, using QoA4ML contract specifications. As a result, the ML service contract in JSON format will be created. Contractual constraints will be in different categories based on terms defined. For example, as depicted in Listing 1, a contract is defined with the following attributes: *response time* (maximum 1.0 seconds), *inference accuracy* (minimum 95%), *input data accuracy* (minimum 90%), and some other attributes related to resource usage, security, and cost. Generally, constraints can be changed on the fly, such as when deployed on a small resource device, the required accuracy can be adjusted to 95%. The constraints will be monitored by a set of policies that are written in Rego [2] format and are coupled with the contract. Contracts and policies will be submitted to the QoA4ML Observability Service for monitoring.

Implementing monitoring probes and monitoring ML contracts: The QoA4ML Observability Service collects the common metrics from the existing monitoring service provided by the ML service as well as the infrastructure provider and the ML-specific attributes from the QoA4ML probes. The QoA4ML offers various probes that need to be integrated into both the user application and the ML pipeline to produce the QoA4ML reports. The probes are packaged in a lightweight library so that they can be instrumented with a few

lines of code, without a significant impact on the application performance. According to the service contract, the ML provider has to employ several monitoring probes within their serving pipeline to expose the desired metrics. To measure the quality of services, such as the response time and accuracy, customers can also deploy some probes on their application. We classify the ML-specific attributes into several categories (e.g., *Quality of Data, Security & Privacy* [5]) for better management and development. During runtime, monitoring probes constantly send the QoA4ML report to the Observability Service, which evaluates contract conditions and reports all contract violations in real-time.

Listing 1: An excerpt from the BTS contract (simplified) for dynamic inference in the edge

```
"resources": {
   "mlmodels":[
     {"id": "ml_inference", "mlinfrastructures": "tensorflow", "machinetypes":
           ["small", "normal"], "inferencemodes": "dynamic"}
   ]
 },
"quality": {
   "services": {
       "Responsetime": {"operators": "max", "unit": "s", "value": 1.0}
   },
   "data":{
        "Accuracy": {"operators": "min", "unit": "percentage", "value": 90}
   },
   "mlmodels": {
       "Accuracy": {"operators": "min", "unit": "percentage", "value": 95, "
            machinetypes": ["small"]}
   }
}, ...
```

Observing contract violations: To evaluate the ML service contract, the ML provider has to employ a QoA4ML Observability Service, which is currently implemented based on Open Policy Agent [2]. At first, all the contracts and policies must be submitted to the Observability Service, which stores service contracts and policies in different resources, thus enabling the contract supports for multiple ML services. Once a QoA4ML report is submitted, it triggers an engine to evaluating this report based on the service contract and policies taken from the requested path. The service contracts and policies could be extracted and updated at runtime via REST APIs.

3 ILLUSTRATED EXAMPLES

3.1 ML services under tests

The first application is the base transceiver stations (BTS) predictive maintenance, in which the ML service predicts equipment failures. The ML service predicts the next possible alarms of equipment in BTS[1] (e.g., high/low AC voltage and high/low moisture/temperature). The second application is object classification in Building Information Modeling (BIM)[4]. In BIM, the customers send the design to the ML service of which ML models will identify possible objects. The ML service could be run on a third-party cloud platform such as AWS. For both applications, the *inference accuracy probes* are integrated into both the consumer application and the ML serving pipeline to measure the accuracy. One probe is deployed on the customer application to measure the service response time, and another probe within the ML service measures the input data quality. Prometheus [3] is used to monitor resources and to collect ML-specific attributes. Thus, we can visualize them in real-time on Prometheus or connect them to other visualization tools. An engine running on the QoA4ML Observability Service takes these metrics to produce service evaluations. The quality of service is logged, and the logs could be used for debugging.

[1]https://github.com/rdsea/IoTCloudSamples/tree/master/MLUnits/BTSPrediction.

3.2 Monitoring examples

In BTS, we train and test the ML service with a dataset provided by a Vietnamese company. By employing the *data quality probes* into the ML serving pipelines, the ML provider can detect contract violations caused by incorrect input data. For example, the customers send improper data, or data is modified during transmission. As shown in Figure 2, we observe several violations within a few periods. The *Data-Accuracy* violation pattern observed from the ML Provider is the same as the *Model-Accuracy* violations on the customer applications (low-quality data may influence the prediction accuracy). Besides, other probes also reveal some violations in terms of *Responsetime*. With QoA4ML supports, the customer can trust the service as all contracts are monitored in real-time. Moreover, the ML provider can trace the root cause of violations by building the dependency graph among components in the serving pipeline (not support yet).

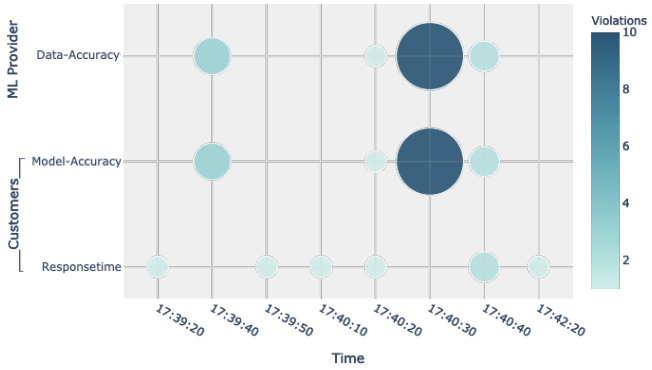

Figure 2: Violation reported in the BTS case.

In BIM, the cost of computation resource for object classification could be expensive, so that the stakeholders have to consent to the trade-offs among accuracy, response time, and cloud resource. Here, QoA4ML can allow them to define and update the contract elastically in runtime without much effort. Moreover, the input data sending to BIM ML service could be structured or unstructured, thus the data quality probes become an important part of the serving pipelines as low-quality input might affect the accuracy significantly and cause contract violations. Since there are more and more objects with new designs coming, monitoring the whole service would also help the ML provider know when the provider needs to update, retrain, or replace ML models.

REFERENCES

[1] Ricardo Bianchini, Marcus Fontoura, Eli Cortez, Anand Bonde, Alexandre Muzio, Ana-Maria Constantin, Thomas Moscibroda, Gabriel Magalhaes, Girish Bablani, and Mark Russinovich. 2020. Toward ML-Centric Cloud Platforms. *Commun. ACM* 63, 2 (Jan. 2020), 50âĂŞ59.

[2] OPA. 2021. Open Policy Agent. (2021). https://www.openpolicyagent.org

[3] Prometheus. 2021. Prometheus. (2021). https://prometheus.io/

[4] Minjung Ryu, Linh Truong, and Matti Kannala. 2021. Understanding quality of analytics trade-offs in an end-to-end machine learning-based classification system for building information modeling. *Journal of Big Data* 8 (2021).

[5] Hong-Linh Truong and Minh-Tri Nguyen. 2020. QoA4ML – A Framework for Supporting Contracts in Machine Learning Services. Under submission. (2020).