January 2023

Automatically Identifying Regression Detection Conditions for System Performance Metrics

Shubham Sharma

Varun Puri

Automatically Identifying Regression Detection Conditions for System Performance Metrics

ABSTRACT

System performance in various computing systems is measured using various benchmarks. A benchmark allows users to observe a set of performance metrics of the system as a function of time and workload, and to determine if a performance metric has deviated or regressed. However, different regression analyzers are suitable for different metrics and finding accurate analyzers often requires substantial manual effort that needs to be repeated whenever a variable that impacts a performance metric changes. This disclosure describes techniques that obtain historical data (sourced from stable workloads) about the pattern of a performance metric and use the data to train a machine learning algorithm to analyze a performance metric and determine a suitable analyzer. The analyzer configuration is selected based upon classification of the metric as noisy or not noisy, and on what is suitable for the particular metric.

KEYWORDS

- Regression analysis
- Performance metric
- Cloud performance
- Metric regression
- Statistical analyzer
- Immediate analyzer
- Metric pattern

BACKGROUND

System performance in various computing systems, e.g., for virtual machines provided by a cloud service provider, is measured using various benchmarks. A benchmark allows users to observe a set of performance metrics of the system as a function of time and workload. In scenarios where a performance metric deviates from an expected value (or range of values), e.g., drops below a threshold, the metric is said to be regressed.

Determining the regression condition of a performance metric is based on multiple factors. For example, such factors include enforcement of a threshold by business (e.g., a customer of the cloud service provider), the nature of the metric (whether it is noisy or not), phase of the workload, state of the system under observation, etc. Further, a number of choices exist for regression analyzers. Statistical analyzers determine the condition of the system on the basis of history of the performance metric over different runs. Immediate analyzers are based on pre-defined thresholds for a certain number of datapoints.

Users that utilize benchmarks (e.g., to ensure system performance) may waste a significant amount of time determining the nature of a metric (e.g., whether it is noisy or not) and finding accurate analyzers by fine tuning the analyzer configuration manually. Inaccuracies in fine tuning a metric may allow a performance issue to creep in. This process is not a one-time activity; rather, it needs to be repeated whenever any variable that has an impact on a performance metric changes.

DESCRIPTION

This disclosure describes techniques that obtain historical data (sourced from stable workloads) about the pattern of a performance metric and use the data to train a machine learning algorithm to analyze a performance metric and determine a suitable analyzer. The analyzer configuration is selected based upon classification of the metric as noisy or not noisy, and on what is suitable for the particular metric.

A cloud service provider that provides virtual machines requires a number of benchmarks with varied performance metrics to qualify the performance of virtual machines. Benchmarks can fail (be unreliable) due to inaccuracies in performance metric analyzers. The pass percentage of benchmarks can be improved by identifying and fine tuning.

Per techniques of this disclosure, this process is automated as described below. First, training data is generated for various performance metrics. Performance metrics are sourced from a number of stable workloads hosting metrics with different monitoring or snapshotting tools. Data is ingested about the analyzers - alerting and regression detection policies that use such monitoring data.

Next, the training data is utilized to train a machine learning algorithm to generate a mapping of metric patterns to appropriate regression detection policies/ analyzers. For example, a non-threshold based analyzer implies that the metric is identified as being noisy.

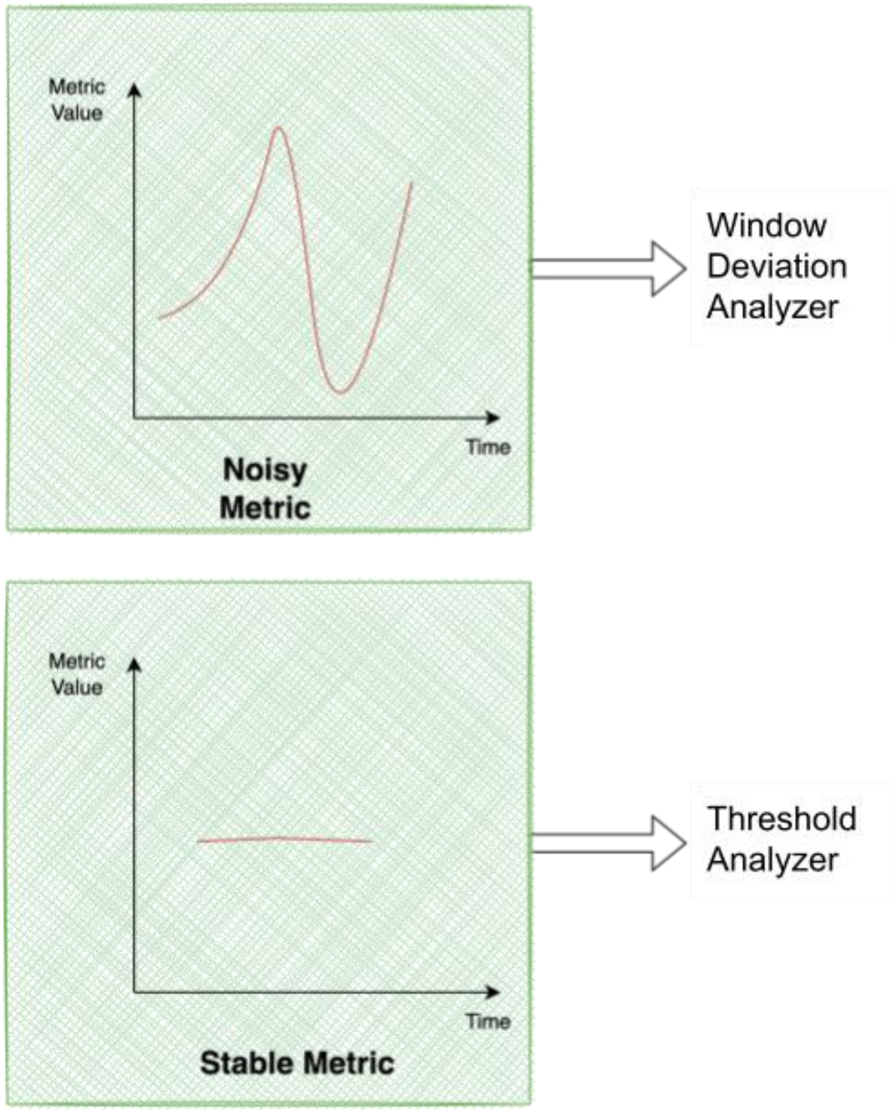

Fig. 1: Using machine learning to determine analyzer configuration based on historical data

Based on the training data and the trained machine learning algorithm, an analyzer configuration is determined. Different strategies are used for different analyzers. For example, thresholds for a threshold analyzer can be identified based on the historical minimum and maximum values of a metric. The configuration for a complex analyzer (e.g., a window deviation analyzer) can be determined using machine learning techniques. For multiple sizes of a historical window, historical metric data can be iterated to check various thresholds, e.g., on mean, medium, percentile, etc. The final surviving configurations can be adopted.

Fig. 1 shows examples of using machine learning to determine analyzer configuration. In the top portion of Fig. 1, a noisy metric (value varying over time) is shown. A window deviation analyzer is determined for a noisy metric such as the one in the top portion of Fig. 1. In the bottom portion of Fig. 1, a stable metric is shown. A threshold analyzer is selected for such a stable metric.

The iteration and selection can itself be performed using machine learning techniques. Once the configuration for an analyzer is determined, the analyzer is simulated over the history of the metric and analyzer results are obtained. This can help build user confidence.

CONCLUSION

This disclosure describes techniques that obtain historical data (sourced from stable workloads) about the pattern of a performance metric and use the data to train a machine learning algorithm to analyze a performance metric and determine a suitable analyzer. The analyzer configuration is selected based upon classification of the metric as noisy or not noisy, and on what is suitable for the particular metric. The described techniques can reduce or eliminate manual efforts to analyze a metric and select a regression analyzer.