December 2022

Categorizing Software Regression Test Results

Varun Puri

Shubham Sharma

Categorizing Software Regression Test Results

ABSTRACT

Customer complaints in cloud computing can originate from components of the cloud infrastructure platform or from components of third-party software. Further, cloud infrastructure components causing issues can affect customers generally or only customers under certain computing environments or using certain third-party software. Pinpointing the origin of a given customer complaint using targeted testing of components in isolation is computationally infeasible. This disclosure describes techniques that correlate test signals across multiple sources to reliably categorize issues in cloud computing to identify the origin of bad rollouts in a timely and cost-efficient manner. An issue that affects a plurality of workloads can cause test signals generated by the workloads to become correlated. By discovering correlations between signals emitted by distinct workloads, determination can be made of the workloads, customer subsets, computing environments, and third-party software impacted by the issue.

KEYWORDS

- Regression testing
- Test failure
- Canary testing
- Testing region
- Cloud computing
- Cloud infrastructure
- Software rollout
- Bad rollout

BACKGROUND

In the context of cloud computing, customer complaints can be considered as broadly originating from:

A. bad changes rolled out in components of the cloud infrastructure platform, the impact of which is felt generally by all customers;
B. bad changes rolled out by a third-party software provider, the impact of which is felt by customers who use the third-party software on the cloud platform; or
C. bad changes rolled out in components of the cloud infrastructure, the impact of which is limited to customers of certain third-party software and/or under certain environments on the cloud platform (example: users of highly optimized virtual machines running a particular database).

Pinpointing the origin of a given customer complaint using targeted testing of components in isolation (e.g., end-to-end tests, probers, functional tests, etc.) is difficult, since covering all the unknown unknowns upfront in the test coverage plan is computationally infeasible.

DESCRIPTION

This disclosure describes techniques that correlate test signals across multiple sources to categorize regressions (e.g., customer complaints or other issues) reliably into one of the three above-mentioned types (A, B, or C) to identify bad rollouts in a timely and cost-efficient manner. In this context, test signals are signals generated while measuring performance or system characteristics, performing end-to-end tests, etc., in the presence of running workloads.

An issue that affects multiple workloads can cause test signals generated by the workloads to become correlated. By discovering correlations between signals emitted by distinct workloads, determination can be made of the workloads, customer subsets, computing environments, and third-party software impacted by the issue.

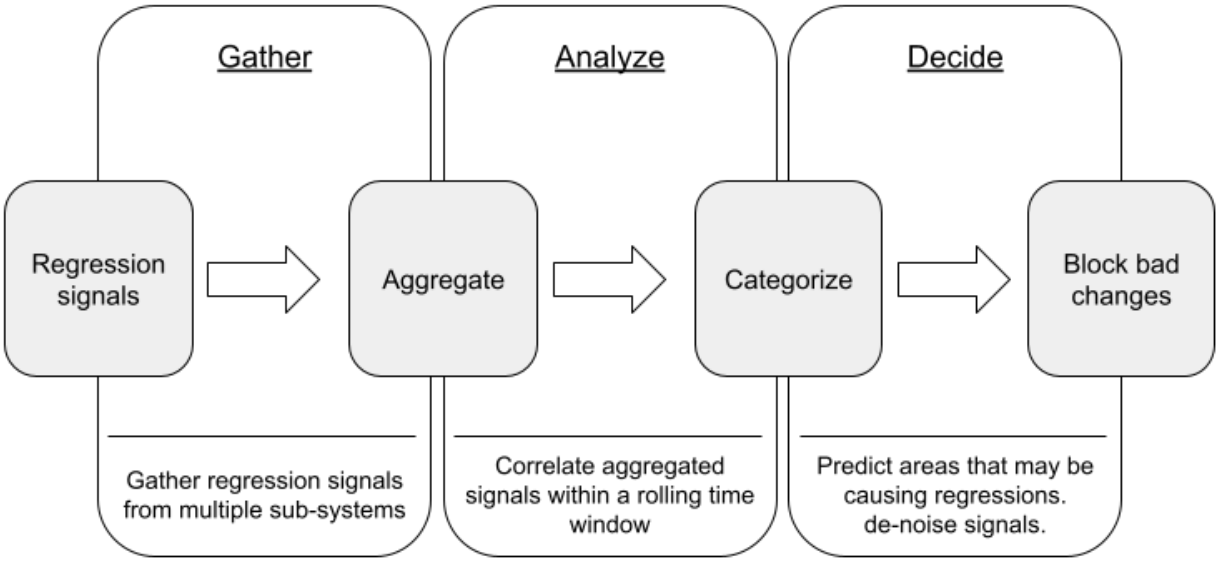

Fig. 1: A framework for categorizing regressions

Fig. 1 illustrates a framework for categorizing regressions, e.g., issues, in cloud infrastructure. The categorization can optimally be carried out in a testing region (private cloud region) of software development that closely resembles that of production software, e.g., with the presence of customer-representative workloads and traffic that increases signal reliability (as compared to pre-production environments). As illustrated, categorization occurs through a gather-> analyze->decide framework, explained below in greater detail.

Gather

Regression signals are collected from multiple subsystems running various kinds of workloads in the testing region for their respective tests. Some example signals are probers that

monitor the health of subsystems with running workloads; continuously running workloads that generate regression signals on performance degradation; functional end-to-end tests that generate metrics for running workloads; etc. Both push and pull mechanisms can be used to ingest signals. Push mechanisms can be based on standard contracts that other systems can use to structure their alert signals for consumption.

Analyze

Signals gathered in a given time frame, e.g., a rolling window of a certain number of minutes, are analyzed to discover correlations that may exist between the signals. The use of a time window allows for some latency between signals generated by distinct workloads tripped up by the same issue.

As explained below, correlations across signals are detected based on issue classification heuristics derived from known historical regression patterns. These patterns are improved iteratively (e.g., using machine learning) for increasingly reliable classification. Datasets of historical regression patterns are used to train machine learning models, enabling the transition from heuristics to ML-based decisions, which are of greater reliability and consistency.

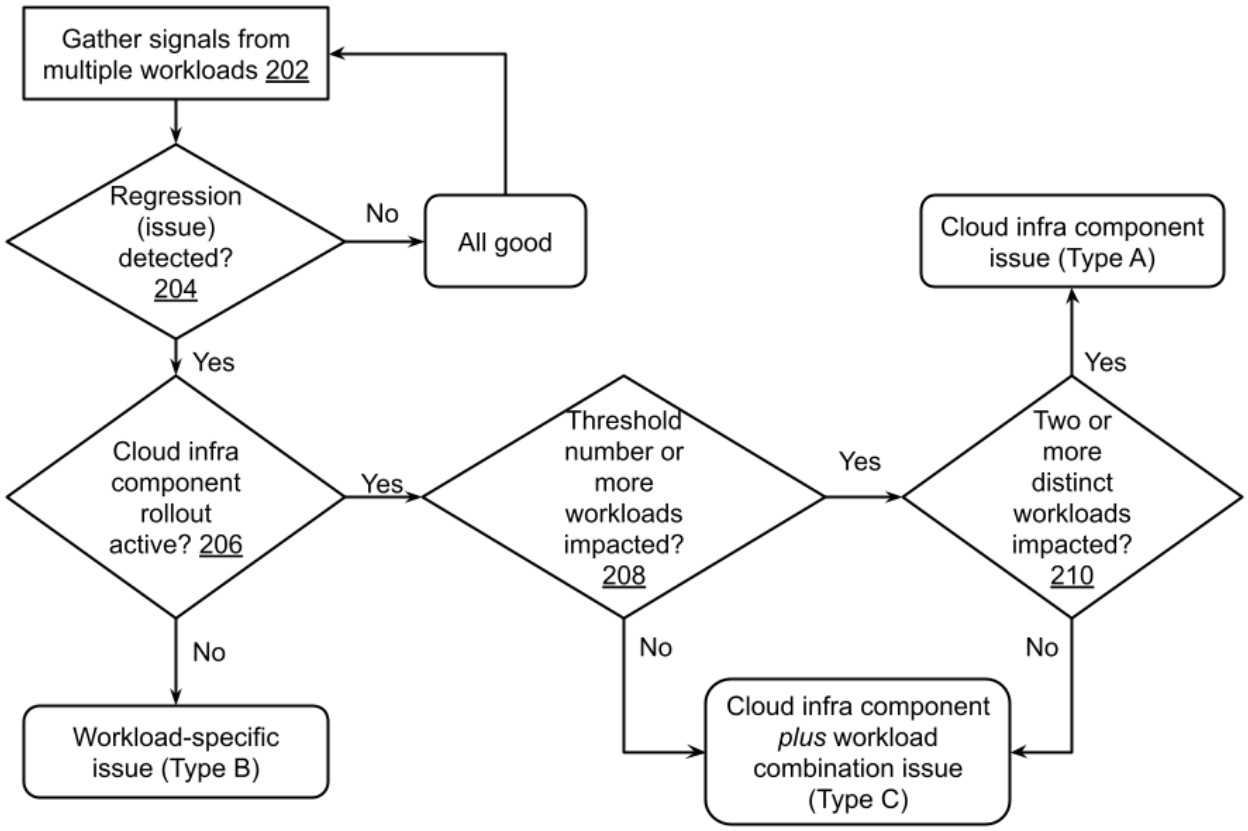

Fig. 2: Categorizing regressions by correlating signals across multiple workloads

Fig. 2 illustrates categorizing regression, e.g., issues, in cloud infrastructure by correlating signals across multiple workloads. Regression signals are detected across multiple workloads (202). Signals are gathered within a rolling time window to allow for some latency between signals generated by distinct workloads tripped up by the same issue. Upon detection of a regression or issue (204), rollouts of cloud infrastructure components are tested for activeness (206, as inferred by rollout events). If no cloud infrastructure component is active, then the regression is classified as type B (bad changes rolled out by a third-party software provider, the impact of which is felt by customers who use the third-party software).

If one or more cloud infrastructure components are active, additional signals from other operationalized workloads in the testing region are observed over the time window. If regression

signals are detected from greater than a certain threshold number (e.g., 50%) of workloads (208) and from at least two different kinds of workloads (210) then the regression is categorized as type A (newly rolled out changes in components of the cloud infrastructure platform, the impact of which is felt generally by all customers). If regression signals are detected from fewer than a certain threshold number (e.g., 50%) of workloads (208) or from the same type of workload (210) then the issue is not widespread, e.g., it is limited to specific cloud infrastructure components and workload combinations (type C).

Decide

While the aforementioned signal gathering and analyses result in inferences relating to the extent of the issue (e.g., attributable to cloud infrastructure rollout and affecting customers generally; attributable to third-party software; limited to certain workloads, environments, and third-party software), noise in the signals can lead to false positives. Also, it can be the case that only a subset of active cloud infrastructure component rollouts is causing the observed issues.

Therefore, prior to making a final decision, an issue attributed to cloud infrastructure components is analyzed to determine the precise component(s) (of all actively rolled out components) that are causing the issue. This also denoises the signal to reduce false positives. Categorization accuracy can also be improved as follows.

- For a given workload, historical datasets are queried for similar regression patterns attributed to specific components.
- Feedback mechanisms such as cube scores are used to detect false positives and to narrow down the search radius for bad rollouts.

In contrast to conventional end-to-end testing, which relies upon targeted testing of cloud infrastructure components for qualification, the described techniques gather signals from

multiple subsystems to arrive at a final decision. This enables countering noise to make reliable, data-driven, categorization decisions to pinpoint the origin of regression failures.

CONCLUSION

This disclosure describes techniques that correlate test signals across multiple sources to reliably categorize issues in cloud computing to identify the origin of bad rollouts in a timely and cost-efficient manner. An issue that affects a plurality of workloads can cause test signals generated by the workloads to become correlated. By discovering correlations between signals emitted by distinct workloads, determination can be made of the workloads, customer subsets, computing environments, and third-party software impacted by the issue.