OpsCruise Use Case

Avoiding the Pain of Setting Thresholds

Problem Scenario: Operations' Pain and the Futility of Setting Thresholds

Abstract

DevOps and SRE teams face a big burden on selecting what metrics to monitor and how to set thresholds on them in order to detect performance problems with today's monitoring solutions. Current approaches generate a lot of noise, as well missing outages when those thresholds are relaxed. Furthermore, false alerts significantly increase the time to find root cause and isolate faults for resolutions. OpsCruise eliminates this wasted effort by detecting emerging problems using its ML-based modeling of the application without user-level involvement. This approach is both preemptive and adaptive since the model is automatically updated keeping up with the ever-expected application changes. The net result of using OpsCruise is significantly improved productivity of the Ops team, with increased organizational agility that also improves top line business performance.

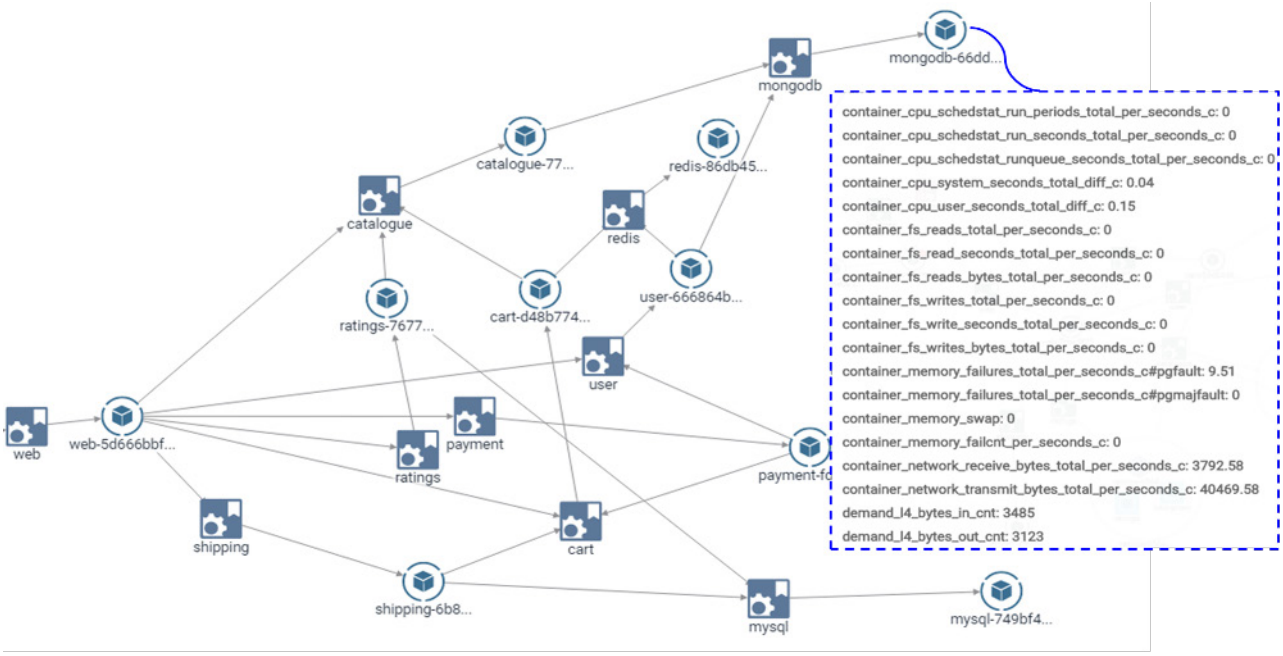
The Traditional Approach

Today's approach to detect application performance problems requires DevOps teams to both select metrics of all services and set the thresholds above which an anomaly is detected.

Consider when the Ops lead responsible for an e-commerce application has to decide thresholds on metrics for all containers in the application. While a generic container has over 30 metrics, some application components, such as a database can have as many as 100 metrics to choose from. This means selecting the right set of metrics to monitor and set thresholds on over 300 possible metrics. And, if a typical application has around 200 containers, this would mean deciding 3000 thresholds.

Choosing only a single metric threshold per container, such as using Prometheus alerts, would reduce the effort. However, such single metric thresholds are often insufficient since there are many complex interrelationships within a service. For example, request counts into a container can affect consumption of CPU, memory, disk, as well as requests it makes on other containers, so a threshold on a single metric, say in the case of the MongoDB service, does not capture the problem dimensions. As a result, a simple threshold Prometheus alert results in a high level of alert storms. Some commercial monitoring tools, including APMs have adopted multivariate statistical baselines for use in outlier detection. These are based on historical statistics, and consider a variety of unrelated metrics including location and time of day of user requests. Unfortunately, this leads to false alerts since those are independent of the application context, i.e., if it is operating correctly across different demand levels. As a result, these multivariate baselines create false alerts, and more significantly, lead to a longer incorrect conclusion in root cause analysis.

Another challenge with setting thresholds is how to tune them whenever even a single component image is updated. Ops faced with frequent changes in the application as desired by more agile frequent releases means previously set thresholds are rendered obsolete requiring setting, or often guessing, new values with every change.



Most importantly, the biggest drawback we keep hearing from Ops is their use of thresholds-based anomaly detection lead to a large number of false alerts, Because they are not related to how the service performs under different loads, relying on historical baselines results in frequent false positives, especially when new demands occur which the service is quite able to handle. A more unfortunate consequence is that of false negative alerts. When thresholds are not relevant they often miss the real problem.

The net result of the above is increased mean time to detect (MTTD), isolate the source of the problem, and then resolve, i.e., increased mean time to repair (MTTR).

Finally, using threshold-based anomaly detection, especially limits on performance, is always reactive resulting in avoidable downtimes. The above example is for one application service or container.

The OpsCruise Solution and an Example

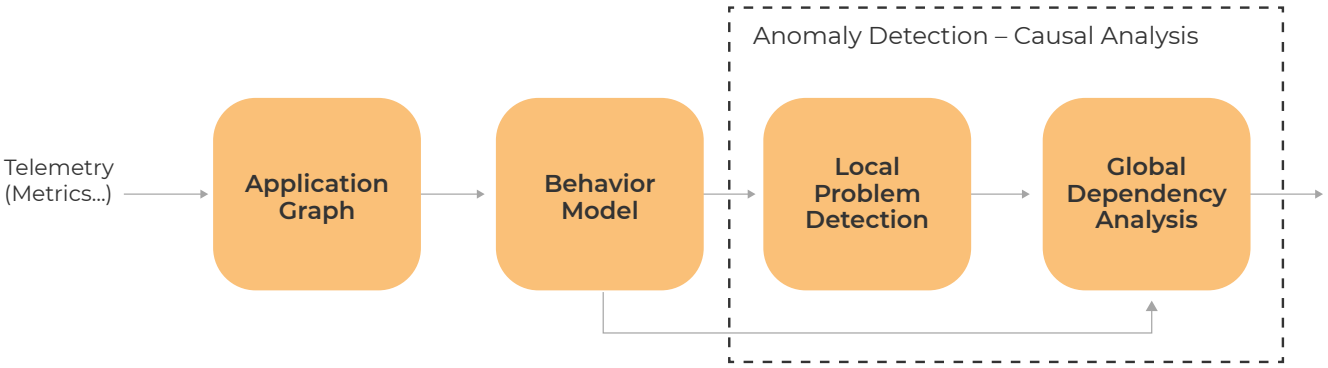
OpsCruise takes an unique application aware approach to proactively detecting problems: an automated ML-based approach that uses learned application behavior to catch application problems as they occur. Using real-time metrics, including flows, events and configuration from open source monitoring frameworks as well as cloud services, the structure, the Application Graph, and behavior of the application is learned. The behavior model is then deployed in runtime to detect deviations from the expected based on the current state, signaling an emerging problem.

In the case of the e-commerce application with 100s of components, the Ops team does not need to decide metrics to monitor nor set thresholds. OpsCruise's ML does the job, and based on learned behavior detects when MongoDB has a problem, and also surfaces the leading metrics that are likely explanations for the anomaly. Ops would have never guessed that this combination of metrics for MongoDB is an indication of an emerging problem.

The Business Impact

OpsCruise's application-aware ML-based anomaly detection:

- **Reduces the Ops hours wasted in setting thresholds that are ineffective and inconsistent.** This wasted effort becomes significant when the number of containers and services grow from 100s to 1000s. For the e-commerce customer, it was several FTEs of work annually.



- **Reduces significant Ops time that is spent on chasing false positives.** For the same customer example, they had several dozen infrastructure and application development employees who were spending 25%+ of their time chasing false positives.
- **Reduces the risk of downtime by decreasing the incidence of false negatives.** While hard to prove, anecdotally, this e-commerce firm acknowledged they uncovered many critical issues that their traditional monitoring tools would have ignored.

The net result of using OpsCruise is significantly improved productivity of the Ops team, with increased organizational agility that also improves top line business performance.

About OpsCruise

OpsCruise provides an observability platform for automated performance assurance of cloud applications.