OpsCruise Use Case

Change Management for Improved Agility

Problem Scenario: Tying Change Management for Faster Resolution

Changes in application are often the cause of problems. While not all performance incidents can be traced to such changes, Gartner estimates that approximately 85% of all performance incidents can be traced back to changes . With modern cloud applications that are more complex, the ability to track back to a specific change that may have caused the problem is now taking significantly longer and the process is more complex.

Abstract

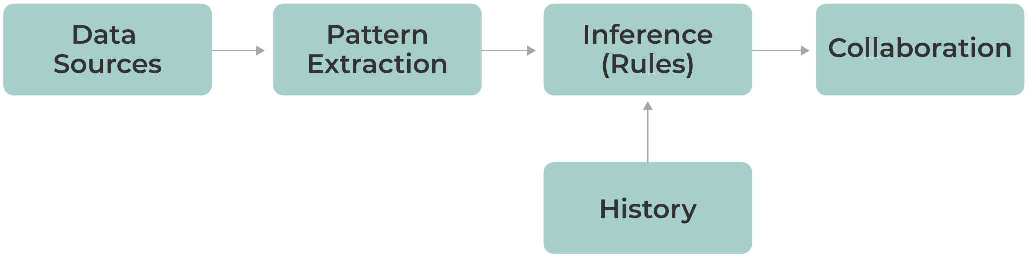
OpsCruise's application-aware approach to provide a coherent and integrated observability into cloud-native applications extends to getting real-time understanding of the application from its service to service interactions to all dependencies on infrastructure and the orchestration. Beyond detecting emerging problems using its ML-based behavior models, OpsCruise keeps a continuous track of the changes, both structural and behavioral, of the application using its time travel feature. In the event of a problem, OpsCruise can quickly point the anomaly or anomalies to a recent change and allow Ops to roll back to the last known good state, thereby significantly improving uptimes, as well as making blue-green and canary deployments more effective.

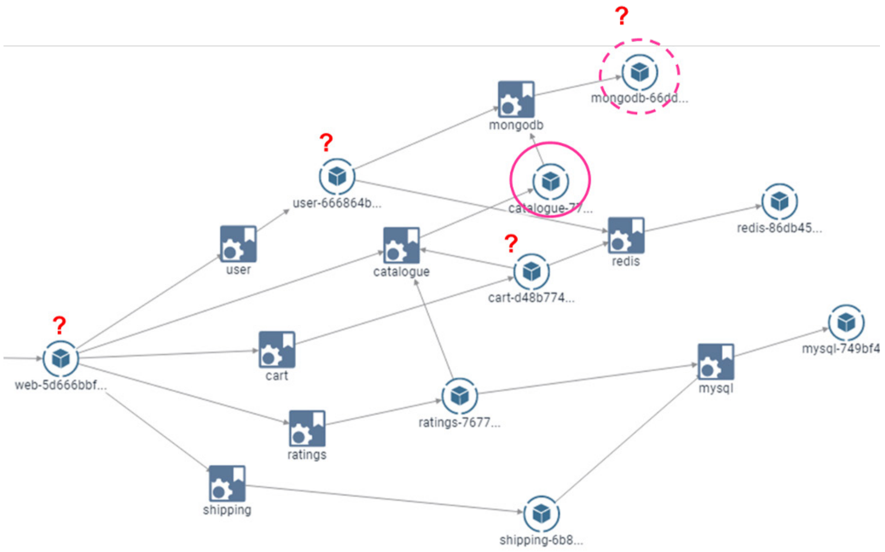
The Traditional Approach in Change Management

The current approach to linking a failure incident to a change requires multiple tools from CI/CD signal to application application monitoring to network devices. Consider the process used by event management tools: first, after a failure has occurred, data related to different entities that can affect the application are collected. Second, patterns are extracted primarily using correlation. Duplicate data are removed and a smaller set of data is analyzed with known inference rules, if available. Third, historical data is checked to look for previously known causes. The final set of data is then further analyzed by domain experts.

Here's an example scenario that we have witnessed. Before the big holiday season, the Catalog department of a Fortune 100 e-tailer releases a whole new list of product offerings and specials. The Catalog team Dev team pushes a new release of the code and associated collateral, including brand new product videos, testimonials, and reviews from suppliers into the repository database, all into the platform. Within hours after the code release, customers placing orders experienced long checkout times. The Ops team rushes into rescue mode and is concerned that the database is not keeping up. Is it more Users? More items in the Cart? Or, a slow database? If they only knew that the new Catalog had generated a lot more load on the database because of the new code push, and the backend store was right-sized, this could have been fixed in *minutes*. What Ops needed to know was *whether the new change in the application would break the platform*!

Today platform and Ops teams are adding new tools, often classified as AIOps that are good for fault management across diverse elements but they do not understand how small changes made in the application can break the platform. A more real-time application and change-aware solution is needed to make Change Management more effective and DevOps teams extremely agile. Enter OpsCruise.





The OpsCruise Solution and the Case of the E-Tailer's Holiday Rush Breakdown

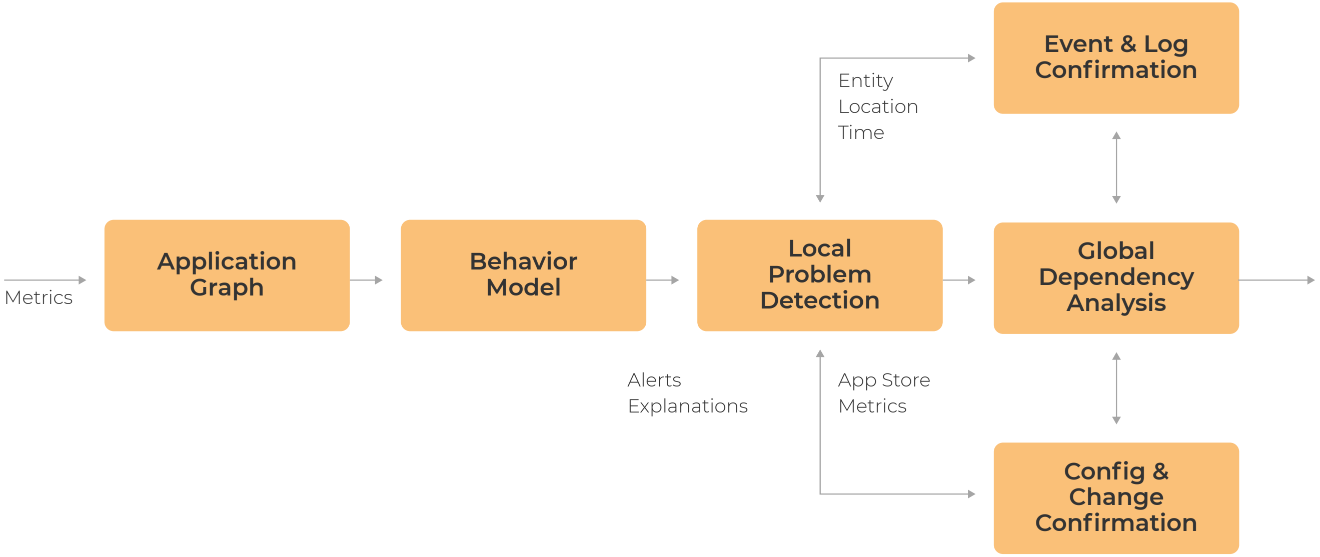
When OpsCruise is observing the e-tail application deployed in Kubernetes (K8s), it builds a deep application understanding. First, it auto-discovers and auto-builds the application from open source instrumented monitoring such as Prometheus for metrics and Fluentd and Loki for logs, besides using information from K8s and cloud configuration, and real-time flows between all services. In the above example, without application code instrumentation, OpsCruise discovers the dependency paths, such as the service path from Web to Catalog to the database MongoDB, as well as User to MongoDB. Subsequently, it builds ML-based predictive behavior models for all containers and services, so it knows what demands Catalog or User are expected to make on the database. Furthermore, it also records all events of all changes made to the applications.

When the Dev team pushed the new release on Catalog, OpsCuise detected the event as an image change in the Catalog container. It also initiated learning the behavior of the new Catalog container. The model captured that Catalog now was making more calls to the database just as User requests were also being sent to the same. So when customer requests started increasing, it noted that the increased load on MongoDB caused higher delays in checkout. Finally, in the causal analysis process, OpsCruise checked for any recent change that was made to the Catalog prior to the MongoDB performance problems, and identified that the most recent release was the change. DevOps teams can now either rollback to a working less I/O demanding Catalog, or reconfigure the database to meet the higher I/O load.

OpsCruise can directly tie-back a change to the problem because of three key reasons:

1. It continually tracks changes in the application environment, whether Kubernetes, the infrastructure, or a code change in any application component.
2. It can detect if any change results in an application problem within minutes, whether in infrastructure, K8s or a code update.
3. It captures snapshots of complete application as part of its Time Travel feature so it has a full structural component code level history.

[1] Will Cappelli, Causal Analysis Makes Availability and Performance Data Actionable, Gartner, 07 October 2015



The Ops team has immediate linkages from an emerging SLO breach in database to any change made in the application estate. Furthermore, the change is detected down to the source of the problem, e.g., specific container or service. This allows Ops to rollback the change immediately after verification. This avoids the

The Business Impact

Improve Customer Satisfaction. OpsCruise's use of early detection of problems with its use of predictive behavior model, understanding of application structure and ability to track and record changes in the application creates a significant positive impact for more agile DevOps. This e-tailer estimated performance degradations cost the company $50M+ year and believes OpsCruise can have a significant impact on getting ahead of issues.

Increase Business Agility and Staff Productivity. Instead of wasting hours debugging and not realizing that a small change has caused a problem, Ops teams can identify and make corrective changes quickly in minutes. This avoids crisis time in war rooms, hours of downtime, while gaining increased business agility with a more productive Ops team. The e-tailer would like to increase feature updates to multiple times/day with the same SRE staffing level, something OpsCruise will enable . . . saving several $M in FTE costs.

About OpsCruise

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