

Applied Resilience for Mission Systems

Orton Huang

Overview

How can we know when our systems are working properly? How can we monitor them? By adapting recent observability approaches from industry, the DoD can improve its ability to monitor and understand its mission systems and to diagnose and resolve mission performance issues.

Observability focuses on three core aspects: metrics, tracing, and logs. Applying Google's Golden Signals at multiple levels in a system architecture combined with distributed tracing provides meaningful understanding of system performance and supports identification of mission performance issues. Well-implemented logging provides the means to effectively diagnose and perform root cause analysis of these issues.

To support observability, leverage an adaptation platform (e.g., Nifi) and an aggregation and visualization platform (e.g., the Elastic Stack) to allow quick ingest of new data sources and provide the ability to explore issues.

Need

The DoD has become more dependent on software than ever before. As our mission systems have increased in complexity, it has become harder to understand and monitor system performance and diagnose issues. The DoD leverages log aggregators and tools like Nagios and Splunk to understand system and application performance, but these often fall short of giving insight into how well the mission is performing, understanding how much additional load or work can be handled, identifying bottlenecks in the system, or recognizing how the system can be improved. The DoD struggles with these questions, and they have become harder to answer with the complexity and distributed nature of our software today.

In order to resolve issues and improve the mission system, we need a more consistent approach to thinking about mission performance and a way to diagnose and explore problems as they occur.

¹ C. Sridharan, April 2017, "Logs and Metrics", <https://medium.com/@copyconstruct/logs-and-metrics-6d34d3026e38>

Approach

Observability simplifies monitoring by focusing on practicality, addressing the distributed nature of applications today, and approaching monitoring from the user's point of view. Observability focuses on three core aspects: metrics, tracing, and logs¹.

Well-implemented observability will help operators and developers understand **how** the mission is performing, **what** software is causing issues, and **why** there is a problem. This can be accomplished by adapting a practice used widely in the commercial sectors.

How is the mission performing? Google's Golden Signals² provide a framework for thinking about metrics. When applied at the workflow, business process, or user-facing services, they provide essential indicators to how well the system is performing. Google's Golden Signals focus on these five metrics:

- Rate: request throughput
- Utilization: average time that the resource was busy servicing work
- Latency: duration, response time
- Errors: count of error events
- Saturation: degree to which the resource has extra work that it can't service, often queued

These metrics are a combination of the USE³ and the RED⁴ approaches. The USE approach (utilization, saturation, and errors) is an internal, resource-centric view that focuses on key metrics about the resources handling the workload. The RED approach (rate, errors, and duration) is focused on measuring the workload that is being asked of the service and treats the service as a black box. Between these two groups of metrics, an observer can understand the users and the work they request as well as the resources and how they respond to the workload⁵.

² R. Ewaschuk, March 2016, Google Site Reliability Engineering Book, "Chapter 6: Monitoring Distributed Systems"

³ B. Gregg, 2012, "The USE Method", <http://www.brendangregg.com/usemethod.html>

⁴ T. Wilkie, May 2017, "The RED Method: Key Metrics for Microservices Architecture", <https://www.weave.works/blog/the-red-method-key-metrics-for-microservices-architecture/>

⁵ B. Schwartz, Oct 2017, "Monitoring and Observability with USE and RED", <https://www.vividcortex.com/blog/monitoring-and-observability-with-use-and-red>



What software is causing issues? Applying Google's Golden Signals at the service and process levels and implementing distributed tracing improves the ability to identify which service is performing poorly or being disrupted.

Distributed tracing is the process of tracking the activity resulting from the request of an application. With distributed tracing, you can:

- Trace the path of a request as it travels across a distributed system
- Discover the latency of the components along the path
- Identify which component is creating a bottleneck

Why is there a problem? Appropriate application logging combined with log aggregation and correlation provide the means to debug system problems. Logs are "immutable records of discrete events that happened over time" (similar to events). Good logging is essential to debugging issues in a system, but can be challenging and costly to aggregate and store correctly. The key is to utilize identifiers such that activity can be correlated across different components and services.

Infrastructure. In order to collect, store, index, and visualize these metrics, traces, and logs, the system needs infrastructure to flexibly adapt and ingest this information and make it available for visualization and analytics. Consider a tool like NIFI for additional flexibility in addition to your log aggregator. Also consider a tool like the Elastic Stack for aggregation, storage, and visualization. Figure 1 captures a potential way to bring these concepts together.

Impact

Understanding system performance. The most natural benefit of applying observability is to better understand the performance of the mission system and when things go awry: the who and the why.

Diagnosing problems. A well thought out infrastructure allows quick diagnosis of problems and the ability to explore new hypotheses to find relationships or causes of other problems.

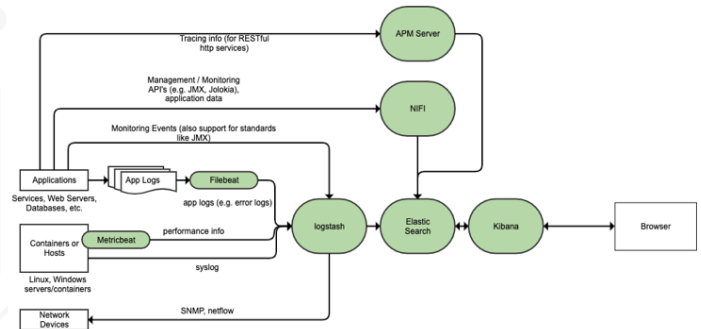


Figure 1: Observability Infrastructure

Performance against Service Level Objectives (SLOs). SLOs are a construct to manage and coordinate performance expectations to all stakeholders. The Golden Signals provides a foundation for Service Level Indicators that provide insight into whether the SLOs are being met.

Leading indicators. Many of the Golden Signals are also leading indicators for when the system may be about to fail and can provide warnings of these failures to admins and operators.

What Should Builders Do?

- Ensure software services and applications are instrumented to report at least the Golden Signal metrics
- Ensure that software provides log errors and basic lifecycle information and that these logs are aggregated
- Ensure that workflows, business processes, and mission functions are instrumented for tracing
- Perform disruptions and use your implemented observability to validate its ability to understand and diagnose the issue
- Ensure that observability infrastructure exists to ingest, adapt, store, and visualize the observability information for diagnosis and exploration.

⁶ C. Jones, March 2016, Google Site Reliability Engineering Book, "Chapter 4: Service Level Objectives"

