

Signal-based Container Monitoring for Improved Microservice Availability

Microservices are a popular software architecture consisting of sets of containers managed by an orchestrator, e.g. Kubernetes. Orchestrators typically monitor container state with periodic probes, e.g. calling an HTTP endpoint to detect container failure or readiness to serve traffic. This **Poll-based Container Monitoring (PCM)** must be carefully tuned for each specific container and hardware, may be slow to detect state changes, and does not detect some failures such as intermittent database failures. An alternative is **Signal-based Container Monitoring (SCM)**, where the orchestrator receives a signal emitted by the container when it changes state. We believe SCM detects state changes faster than PCM while requiring less tuning.

To test this, we have implemented a prototype, **SCMKI**, of signal-based container monitoring for Kubernetes. SCMKI interprets container logs as signals instead of sending periodic probes. We compare the time to detect container failure and readiness under SCMKI and probes using the Sockshop benchmark. SCMKI detects a simple failure 86% faster than probes and readiness in a similar time. We find probes configured to detect failure quickly may detect false positives, leading to superfluous container restarts and a 4% reduction in service availability over SCMKI. Finally, we show a mathematical model that predicts SCM has a theoretical edge over probing. Future work will evaluate SCMKI using additional failures and benchmarks, e.g TrainTicket and TeaStore. We also plan to update SCMKI to reduce resource overhead.