Low Latency Kernel Monitoring with XDP in Large Distributed Environments

Yuki Maruyama, Tomohiro Kano, Kenta Ishiguro, Kenji Kono
Keio University

Background. In large distributed environments, multiple servers run tasks in parallel while communicating with each other. To maintain health and performance of distributed systems, a monitoring service collects various metrics from the entire infrastructure, and is responsible for launching maintenance jobs such as load balancing and failure recovery. A monitor sends monitoring messages to each server, and the recipient server replies to the monitor with various metrics such as CPU load. The monitoring messages must be collected in a timely fashion; if the collected metrics are out of date, the monitor misjudges the system status and makes inappropriate decisions for the maintenance.

Problem. It is widely recognised that monitoring messages are often delayed due to overloading of the servers. In traditional monitoring systems such as NetData [1] and Prometheus [2], monitoring latency increases by more than 10× when CPU is saturated [5]. Amazon DynamoDB experienced a major outage caused by delays monitoring messages [3].

Goal. The goal of this research is to provide a lightweight, low-latency monitoring framework that can prevent delays in monitoring messages. In this framework, monitoring latency is unlikely to be affected by servers' loads. The framework can run with traditional Ethernet NICs and is not intrusive to existing operating system kernels.

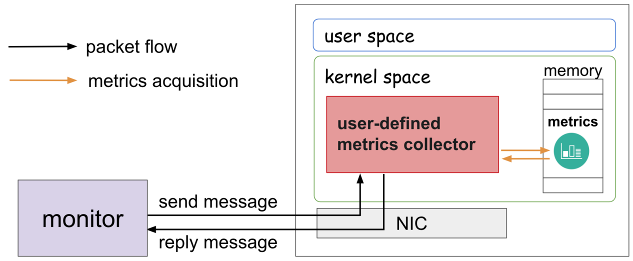

Figure 1. Overview of the proposed method

Approach. The key features of our framework are as follows: *In-kernel Monitoring:* Metrics are collected inside the kernel. This design leads to a lightweight design because the collection does not involve user/kernel context switches. *SoftIRQ Layer:* Metric collection is conducted in the soft IRQ layer. This layer is invoked immediately after hardware interrupt handling, and executed in the interrupt context. This design minimizes the latency involved in packet handling. *Generality:* All the kernel metrics procfs provides can be collected. Users can install a small piece of code inside the kernel that collects and manipulates kernel metrics. The safety of the user-defined code is validated before execution.

Fig. 1 shows the overall design of our framework. A monitoring message is delivered to the user-defined code that collects kernel metrics. The latency is minimized since this code is executed in the soft IRQ layer, and less likely to be affected by the server's load. Metric collection can be done via the procfs interface. No synchronization is necessary because the monitor only reads the metrics. Our framework has been implemented by slightly extending XDP [4], in-kernel packet processing in Linux. To allow XDP code to access to kernel metrics, a new helper function has been added, which reads kernel metrics through procfs interface.

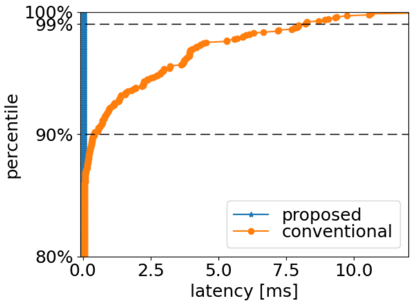

Figure 2. Monitoring latency (CDF)

Experiments. We conducted an experiment to measure the monitoring latency of RocksDB, using YCSB write-only workloads.[1] Fig. 2 shows CDF of the monitoring latency. The 50th, 90th, and 99th percentile are 28.0 μs, 31.0 μs, and 34.0 μs in the proposed approach, whereas those of the conventional approach are 43.0 μs, 424 μs, and 8.20 ms, respectively.

References

[1] [n. d.]. Netdata - Monitor everything in real time for free with Netdata. https://www.netdata.cloud/. Accessed: July, 2024.

[2] [n. d.]. Prometheus - Monitoring shstem & time series database. https://prometheus.io/. Accessed: July, 2024.

[3] [n. d.]. Summary of the Amazon DynamoDB Service Disruption and Related Impacts in the US-East Region. https://aws.amazon.com/jp/message/5467D2/. Accessed: July, 2024.

[4] Høiland-Jørgensen et al. 2018. The eXpress data path: fast programmable packet processing in the operating system kernel. In *Proc. of the 14th Int. CoNEXT '18*.

[5] Wang Zhe et al. 2022. Zero Overhead Monitoring for Cloud-native Infrastructure using RDMA. In *USENIX ATC 22*.

[1] Monitoring and monitored machines have a 3.80GHz-6core Intel Xeon E-2276G processor with 32 GB of RAM and Intel X540 NIC.