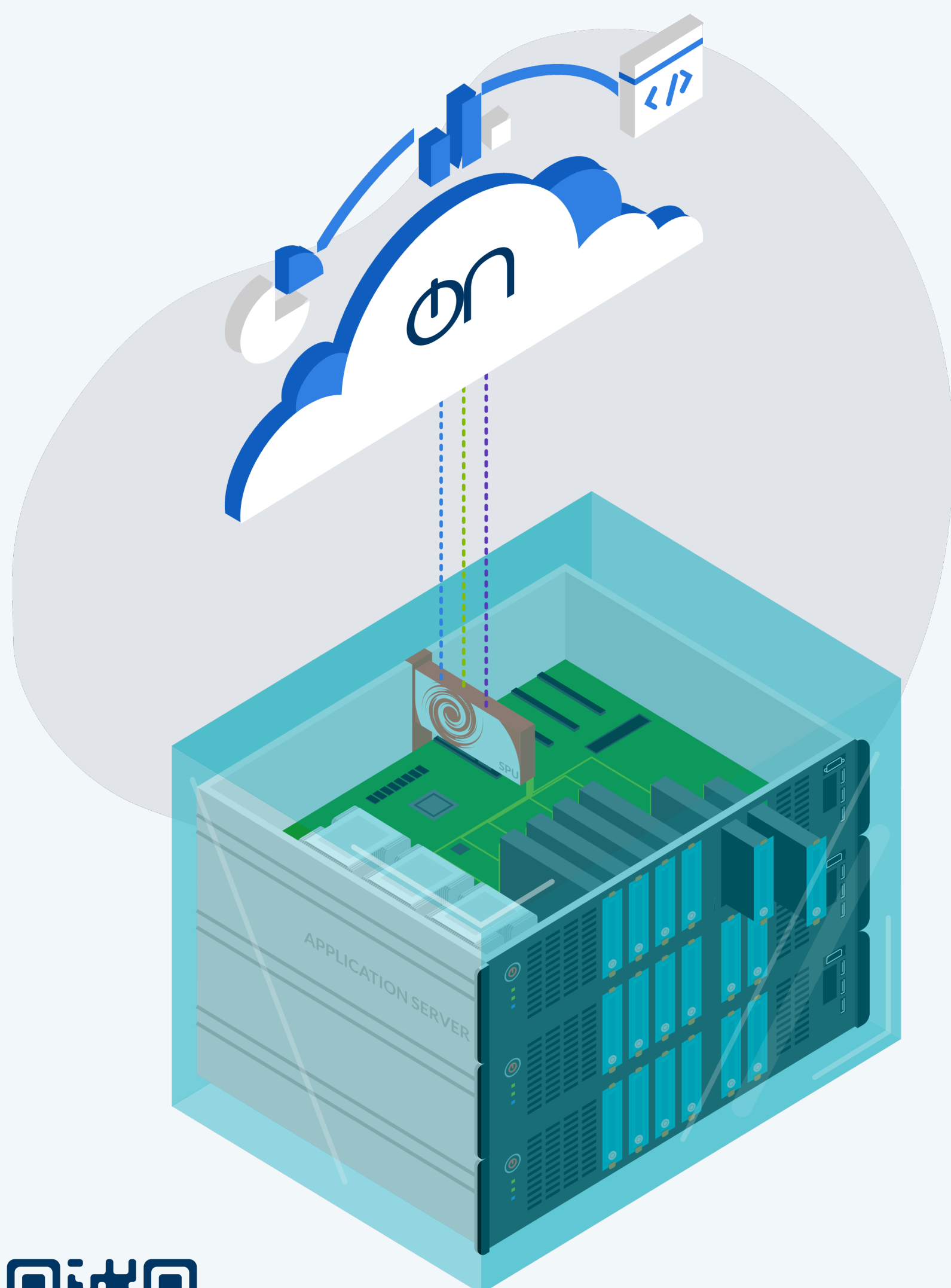


Description

I implemented error-logging and metrics on functions that queried databases in the cloud, tracking the duration and total accumulated count of these queries. These metrics were then used to create a dashboard which displayed both the top ten query counts and durations over a period of time to help pinpoint hotspots in CPU usage for heavy queries.



Technical Challenges

A technical struggle I faced was my lack of familiarity with metric-collecting interfaces such as Prometheus, PromQL, and Chronosphere. Although I was able to implement the metrics locally using documentation and examples from existing metrics in the codebase, the tough part was testing them to make sure that they worked as I intended them to. I couldn't figure out how to trigger my metrics from a test environment deployment in a Kubernetes pod and further verify they would be visible in Chronosphere. Since I couldn't rely on the internet or documentation for this specific problem, I asked my manager, other co-workers, and even the VP of Engineering how I could test my metric implementations and immediately received clear and helpful advice. I was able to access a custom port for metrics in the Kubernetes pod for test deployments, which I did not have permission to use before, and verify that my implementation worked.

Personal and Professional Growth

I've realized that the most important way to learn is to ask questions, especially in a restricted technical environment where answers aren't simply on the internet, but rather with the other engineers themselves. When introduced to new ideas, I wouldn't be able to produce good questions in the moment because I was too busy digesting all the new information handed to me. Throughout the course of my internship, I've gotten much better at recognizing the holes in my knowledge and asking questions based on them to yield better and more tailored answers from my managers. This change allowed me to become more efficient when given a new project, instead of getting lost in the details and wasting time which could be used to complete my task.

Ethics & Society

An important feature of the Nebulon services processing unit (SPU) is that it detects ransomware attacks on servers and offers near-instant restore capabilities for critical application data while recovering infected operating system volumes. This system allows client and customer data to be secured and not fall in the wrong hands of someone who might have malicious intents, all while saving the client millions of dollars in critical downtime and ransomware costs.

Connection to UCSD

CSE 12 (Basic Data Structures and OOP)

- Although CSE12 was primarily focused on Java, its lessons on JUnit and inheritance/ encapsulation have tremendously helped me dissect my company's Go codebase and create deterministic, specific, and comprehensive tests.

CSE 15L (Software Tools and Techniques)

- Having understood version control and git terminology such as merge, push, and commit in CSE15L made my transition into a production environment seamless.

What I'll bring back to UCSD

- I've noticed that although the contents of the classes at UCSD are important, their hidden lessons on problem solving, critical thinking, and time management will be far more valuable as I transition into the industry.

