Expanding Envoy Health Checking

Brian Gregg
Noah Picarelli-Kombert

Introduction to Envoy

- Envoy is an open source communication proxy designed for cloud systems
- High levels of scalability and transparency
- A common configuration is to have Envoy act as a network proxy between servers along with a sidecar on application servers
- Load balancing and health checking are key features of this configuration
- Three health check types:
  - L3/L4
  - HTTP
  - Redis
- Our project focuses on the Redis custom health check

Introduction to Redis Health Checking

- Redis is an open-source data structure store
- A Redis “database” of key-value pairs sits on the application server
- The Envoy front proxy can query the Redis server by key
- PING is an out-of-the-box Redis command that will always return PONG
- EXISTS will return a 0 if the passed key exists, and a 1 otherwise
- GET returns the value at a given key
- In the existing Envoy source code, two types of Redis health checks exist
    - PING - expects PONG
    - EXISTS - expects a 0 (key does not exist)

Project Goals

- Expand Redis health checks in Envoy to account for varying server health states
- Test our new health check to ensure functionality
- Learn about Envoy, its implementation in cloud systems, and related cloud technologies

Updates to the Envoy Source Code

Configuring Docker for Server Communication

Testing and Results

- Testing method
  - Set up 4 test containers and choose a health check key
  - Manually set each container's disk utilization based on the key to test healthy/unhealthy
    - Lower than key for healthy test
    - Higher than key for unhealthy test
  - Start Envoy in Docker environment
  - Check health status of test containers and load balancing distribution across containers
- Testing limitations
  - HTTP and Redis routing on the same endpoint proved too challenging, so load balancing between HTTP application servers using our health check results was not feasible
    - *However*, we were able to test load balancing using more Redis containers instead

Takeaways and Potential Improvements

- Learned about Docker and setting up a virtual Docker environment
- Utilized Cloudlab to compile and run Envoy
- Learned about the source code for Envoy
- Discovered Redis and its uses in a cloud environment

- Our new health check is functional but can be improved
  - Ideally, could be combined with an HTTP application to improve posterity
  - Could be made to allow more granular responses to health check requests
  - Could be built as a health check type separate from the Redis type to allow for integration into existing Envoy-using projects