Final Slack Root Cause Analysis (RCA) Report

Date: 2021-01-14
What: RCA for *Slack outage*
Date of Incident: 2021-01-04 7:00am PST - 10:40am PST

Issue Summary
Starting around 6:00am PST on 2021-01-04, some customers started experiencing occasional errors and increased latency while using Slack. Around 7:00am PST there was a rapid increase in errors and Slack was not usable for all customers. Around 8:45am PST some customers began to see improvements, but others who were trying to launch their Slack clients were not able to do so. By around 9:15am PST most customers were able to use Slack again normally. We continued to experience elevated errors until 10:40am PST, after which all customers were able to use Slack again normally.

Root Cause
Around 6:00am PST we began to experience packet loss between servers caused by a routing problem between network boundaries on the network of our cloud provider. By 6:30am PST the packet loss began to worsen, causing increased error rates from our backend servers. We were paged at 6:46am PST due to the high error rate.

In addition, many backend servers were busy servicing high latency requests due to network problems between our backend servers, other service hosts, and our database servers. While these high latency requests were only 1% of the incoming requests they used about 40% of the server time.

Subsequently, our service discovery system marked many backend servers as unhealthy due to the network problems. Our load balancers entered an emergency routing mode where they routed traffic to healthy and unhealthy hosts alike. The network problems worsened, which significantly reduced the number of healthy servers. By 7:00am PST there were an insufficient number of backend servers to meet our capacity needs. Customers either could not load their Slack clients, or saw error pages directing them to the Slack status page.

Our backend server fleet then began to automatically scale up to meet traffic demand. Between 7:01 and 7:15am PST our automation attempted to simultaneously add 1,200 servers to the backend fleet, a much higher rate of server provisioning than we normally handle. Our provisioning service, which configures new servers in our cloud environment, had three problems:

1. It was unable to keep up with the multiple tasks of configuring servers at the requested rate, such as setting up DNS configuration, resulting in unhealthy servers in the fleet. Provisioning began to take increasing amounts of time, until it reached 30 minutes which is when the operation timed out.
2. The service ran out of open file handles, because it kept an open file handle for each server that it attempted to provision, and it was provisioning more machines at once than we had load tested for.
3. The service ran into API rate limits from some of our cloud provider APIs, which slowed down the provision operations further.

This happened in large part because our provisioning service failed under the unprecedented load of provisioning requests. It was exacerbated by the ongoing networking problems, which caused some newly provisioned servers to have problems contacting the provisioning service or the configuration service, and those machines could not get far enough through the provisioning process to start their services. Both of these causes resulted in partially provisioned servers which could not take traffic. Eventually our provisioning service was no longer provisioning machines.

Our observability platform was also not reachable for most of the incident because its connection to its database was subject to the network instability. This complicated our debugging efforts and extended the timeline to recovery. We attempted to reprovision observability platform servers for troubleshooting, but were unable to do so due to the problems with the provisioning service. We began to do direct queries to our metrics backend.

Around 8:05am PST we diagnosed that our provisioning service had run out of open file handles. We fixed that by increasing the file handle limit at 8:13am PST. We were then able to successfully provision servers which entered service and served traffic, which indicated that our cloud provider's network instability had improved. We later were told by our cloud provider that during this time period they increased their network capacity in an effort to reduce the instability.

We then provisioned new backend servers and observed them successfully taking production traffic. We worked with our cloud provider to lift the rate limit which was restricting how quickly we could provision new machines. We steadily increased our backend server fleet size and began to see successful customer traffic. By 9:15am PST customers were able to use Slack again. We continued to experience elevated errors due to the network instability until 10:40am PST when our cloud provider had finished increasing their capacity. All customers were able to use Slack again successfully at this time.

Corrective Action

We have completed our detailed investigation and compiled our corrective actions to prevent future incidents of this kind.

- Our cloud provider has increased the capacity of their cross-boundary network traffic systems, as well as moving us from a shared to a dedicated system. They have provided us with a detailed RCA.
- We have a new runbook for how to debug our systems through direct queries to our metrics backend without our observability platform.
- We have prepared methods to configure some services to reduce cross-boundary network transit. If the problem occurs again we can use these methods.
- We have increased the open file handle limit on our provisioning service workers.
- Our cloud provider has increased the API rate limit on the cloud service APIs we call as part of the provisioning process.
- We will create an alert for packet rate limits between network boundaries on the network of our cloud provider. This work will be done by 2021-02-12.
- We will increase the number of provisioning service workers to improve our capacity to provision servers quickly. This work will be done by 2021-02-12.
- We will improve observability on our provisioning service. This work will be done by 2021-02-12.
- We will revisit our provisioning service design. This work will be done by 2021-04-13.
- We will load test our provisioning service. This work will be done by 2021-04-13.
- We will revisit our backend server scaling automation, to ensure we have the right settings for predictive scaling, rate of scaling, and metrics that we use to scale. This work will be done by 2021-04-13.
- We will improve our runbooks for debugging networking issues. This work will be done by 2021-02-12.
- We will investigate and test two settings for our backend servers. The first is the request timeout, the second is the number of simultaneous requests being handled. By adjusting these values we may be able to prevent a small number of requests from using the majority of the resources on the backend servers. This work will be done by 2021-04-13.