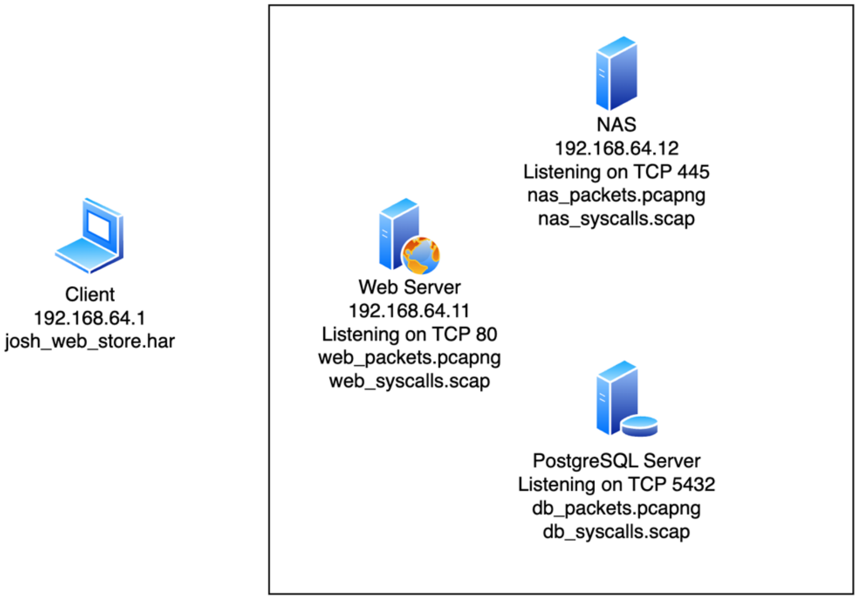
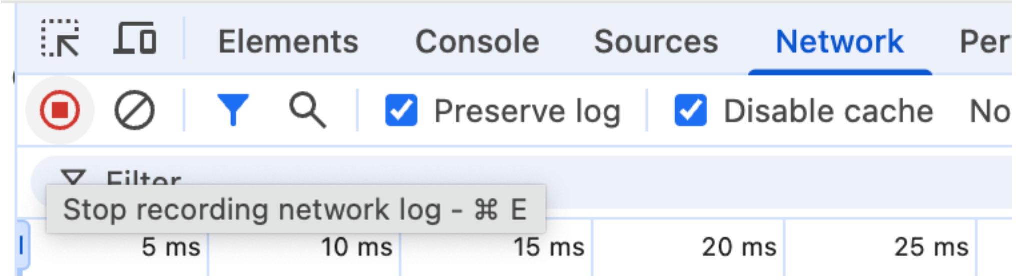
Distributed Application Troubleshooting with Wireshark and Stratoshark

This exercise uses the files in the distributed_application_captures folders to troubleshoot a distributed web application. For an extra challenge, use only the first page to complete the exercise.

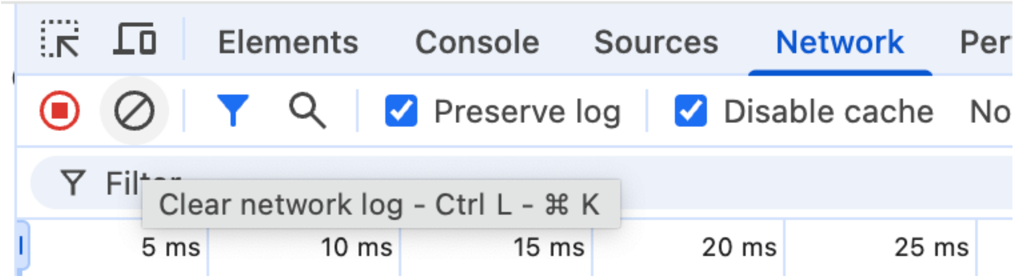


Use josh_web_store.har to identify the 3 slowest calls, and then use the included packet and system call captures to identify why each call is slow.

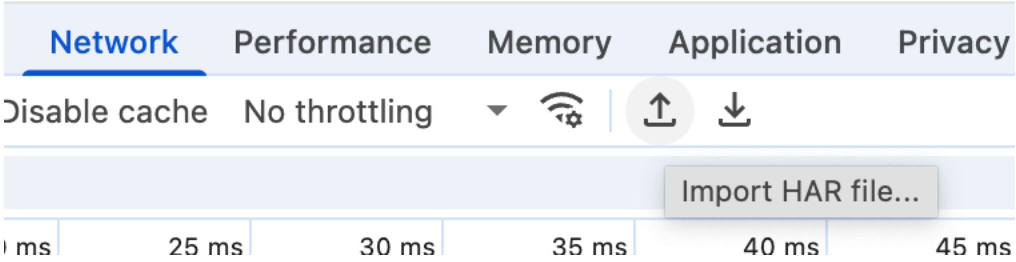
1. In your favorite web browser, open developer tools.
    a. In Chrome, you can go to View -> Developer -> Developer Tools. Click on the Network tab and click the “Stop Recording Network Log” Button



    b. If there are any network calls in the list, clear them



    c. Click the Up Arrow to import the file josh_web_store.har



    d. Click on the Time column header to sort by time. List out the 3 slowest calls, including their name and their overall load time

2. Open web_packets.pcapng. We want to look at the packets to see where the latency occurs
    a. Filter to HTTP traffic
    b. Craft a filter to look for the HTTP request of the slowest call

    c. Filter to that TCP session. What is the frame number of the last frame of the HTTP request?

    d. What is the frame number of the TCP acknowledgement of that request?

    e. What is the frame number of the first frame of the response data?

    f. The time difference between the TCP acknowledgement and the first frame of the response data is application delay. What is the application delay of this call?

    g. Change your display filter to look at all traffic that occurs between the acknowledgement and the first frame of the response data using the format ‘frame.number in {<first_frame>..<second_frame>}’

    h. Looking at this traffic, does it appear that this call relies on backend calls to either the NAS or the database?

i. Repeat this process for the second and third slowest calls

3. Troubleshooting NAS calls: use the following process to troubleshoot slow calls that rely on the NAS

    a. Create a filter that allows you to look at the HTTP request frames and the SMB2 traffic

    b. How long after the HTTP request does the web server send an SMB2 Create File Request for the requested file?

    c. How long does the web server take to download the file from the NAS? This will be the time between the Create File Request and the last Read Response for this file.

    d. Once the file is closed, how long does it take the web server to begin sending it back to the client. This will be the time difference between the last Read Response and the first frame of the HTTP response.

    e. Where is the bulk of the slowdown?

f. Look closely at the traffic for that portion of the backend call. What could be causing the slowdown?

g. If there are other slow NAS calls, repeat the troubleshooting process here.

4. Troubleshoot DB calls: use the following process to troubleshoot slow calls that rely on the database
    a. Open the db_syscalls.scap file

    b. Create a display filter to find network interactions with the web server using the format ‘fd.name contains “<web_server_IP>”

    c. Which event contains the backend request?

    d. What is the SQL query?

    e. Which event contains the response to the web server?

    f. Set up the Find tool to look for Event Details and search for that SQL query. Where else does it show up?

    g. Databases often record query performance. Can you find the query performance message for this query?

    h. Between the request and response, does postgres make any system calls that look like they could contribute to query latency? Look at the “Latency (s)” column.

i. If there are any other slow database calls, repeat the process here