1. General Feedback to All Groups

We first give feedback on things that many of you did and perhaps could use some feedback on. Specifically:

Overall Presentation

Make your presentation uniform! Make slides that look nice and make them all look consistent and clean. Spend time on this!

Data Presentation

Label all graph axes! You should never present a graph without clear labels on all axes – it just makes it harder to understand.

Use the right sized units! e.g., don’t put .0001 seconds when you can simply put .1 ms and don’t put 1000000 microseconds when you can put 1 second. Figure out what the best unit is for the granularity of measurement and use that.

Show all the data if you can. Not just averages – averaging hides information! Or, at least, look at all the data before you do the average, etc.

Graphs: generally, start at zero on the y-axis. Especially for bar graphs. Otherwise you could be misrepresenting the data.

Make graphs next to each other have same height/max value for y axes! This allows comparison across graphs.

Put the thing you’re comparing on the same graph near one another. Graphs facilitate comparison; thus, put the things you want the audience to compare near one another.

Use consistent colors across graphs, e.g., GRPC always blue, Thrift always green. That way, it’s easier to read graphs!

Empirical Methods

In general, you cannot meaningfully compare time across machines. Thus, when measuring, you have to think about what you can do on one machine.

A disk seek is not just an fseek! To get a disk to do a seek, you have to think about how to avoid caches (like the file system page cache, or the disk’s cache) and make sure the disk is read/writing blocks from far away spots on the disk.

Compare your measured numbers to spec sheets. What does the CPU/disk/SSD/DRAM expect to deliver? How do your numbers compare? Are there major discrepancies?

Testing should be done on the same setup, to facilitate comparison! Do not compare one system on one machine to another system on a completely different machine. This is not meaningful!

If the number looks weird, run it again. Ask why the number looks weird. Is it repeatable? What could be going on? What other numbers can you get to figure out what is going on? You can remove noise with careful experimentation!

Be curious! Figure out why things are the way they are. For example, a number of you used the C++ chrono::whatever timer. How does it work? You can find out!
Drawing Conclusions

Show data, and then try to draw conclusions. What does the data show? Make sure the reader/audience can see the same data, and draw the same conclusions from the data you have shown.

Don’t make guesses as to why - you’re probably wrong! Make a guess so as to then do subsequent measurements, and confirm/deny your guess. Having a guess is easy; showing that your reasoning is solid is hard and requires work.

Don’t draw strong conclusions when no strong conclusion is warranted! It’s better to say “I don’t know” than to put forward the wrong conclusion.

2. Feedback to Your Group

Overall, good effort. However, the presentation could have been more coordinated. Work together more, prepare slides, and practice timing (the presentation exceeded the time limit).

Part-1

- Using clock_gettime() along with clock_monotonic() makes sense
- Use of linked list for measuring memory access times to avoid prefetching makes sense.
- Some numbers look a little off (SSD random reads, for example, were actually measuring DRAM reads. You might wanna try some ways to clear caches before measuring IO numbers)

Part-2

- Drop rate graph looks good - might be better to use log scale for Y-axis though.
- Overall, the barebones library looks good!

Part-3

- Unclear why there is a latency jump when moving from 4k to 8k packet size (while it is relatively flat before and after)
- bandwidth numbers were not shown
- It’s a mystery why thrift is really slow compared to the barebones library

Handin

- Roster

<table>
<thead>
<tr>
<th>Member</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marvin Tan</td>
<td><a href="mailto:marvin.tan@wisc.edu">marvin.tan@wisc.edu</a></td>
</tr>
<tr>
<td>Sam Kottler</td>
<td><a href="mailto:skottler@wisc.edu">skottler@wisc.edu</a></td>
</tr>
<tr>
<td>Hao-Yu Shih</td>
<td><a href="mailto:hshih25@wisc.edu">hshih25@wisc.edu</a></td>
</tr>
<tr>
<td>Karan Grover</td>
<td><a href="mailto:kgrover2@wisc.edu">kgrover2@wisc.edu</a></td>
</tr>
</tbody>
</table>

- Checked handin: skottler