ASL Exercise 6

Good/Bad Plots & Explanations
System Under Test

Disclaimer: All Numbers in these slides are not related to this year project.
Maximum Throughput Achieved in the System

• Experiment:
  – Vary the number of clients.
  – Vary number of threads in the thread pool.

• Answer two questions:
  – What is the maximum throughput the system can achieve?
  – and what is the best number of threads to achieve that? Explain your choice?
What is Wrong with the Plot?

- Line colors are similar.
- Missing over saturated part of the graph.
Response time goes down and up

And this one?

“The response time grows significantly after 48 concurrent clients for most thread levels.”

What the student should do?
- Check if losing requests?
- Does run the experiments at the same time?
- Sleeps or timeouts in worker threads?
"Also, observe that the data fluctuates a lot but this is again due to the lack of repetitions. The experiments between 300 and 550 clients were performed at a different time than the others, implying that the virtual machines were shut down, which explains this performance drop."

**What the student should do?**
- The graph is overdone, the part after 400 clients is not necessary.
- Fluctuations will appear less if Y axis starts from 0.
- Should do repetitions.
**What is Wrong with the Plot?**

- Misses under and over saturated parts of the plot.
- Y axis does not start from 0. Lines will be flat if it starts from 0.

“I can report 60, 800 as my middleware highest throughput since the throughput doesn’t change much compared to previous reported configuration”
“Zero mean test suggests that Configurations 120, 160 and 200 Threads do not have statistically significant difference, hence we pick 120 threads configuration at 300 clients.”
Effect of Replication & Number of Servers

• Experiment:
  – Vary the number of Servers (3, 5, 7).
  – Vary replication factor (none, half, full).

• Answer two questions:
  – How throughput changes with respect to #Servers? Explain Why?
  – How throughput changes with respect to Replication? Explain Why?
  – How the GETs and SETs are impacted by the experiment parameters?
“for 7 servers, more threads compete for CPU resources than in the 5 servers case, hence throughput drops for both sets and gets.”

**Incomplete Explanation**

- What about 3 servers?
- Effect of replication?
Aggregated Middleware Throughput (95% GET, 5% SET, VC=300, T/S =24)

The left plots show the throughput of get requests and the right plot shows the throughput of set requests.
A Better Explanation

• “For a fixed number of servers increasing the replication factor reduces the throughput of both GETs and SETs. As visible in [the other figure], for SETs, the request response time increases as the writer thread spends more time to collect responses. For GETs, even though the service time should stay the same, servers are busier with more SET requests hence it increases the overall response time of GETs (due to longer queuing times).

• For fixed replication factor, increasing the number of servers leads to drop in performance for both GETs and SETs. This is because the number of thread pools is dependent on the number of servers and increasing the number of servers adds more threads to the system. As a result, more threads are competing to use the system resources leading to various inefficiencies. Since the middleware is saturated already for 3 servers, adding more threads thrash the CPU. As shown in Figure [other] in this case the response time of SETs ... ”
**What is Wrong with the Plots?**
- Queue and process times do not match total times.
- Response times contradict numbers in other figures.

**What the student should do?**
- Sanity check of all numbers and if they add up.
Effect of Writes

• Experiment:
  – Vary the number of Servers (3, 5, 7).
  – Vary replication factor (none, all).
  – Vary SETs percentage (1%, 5%, 10%).

• Answer two questions:
  – How throughput changes with respect to Writes? Explain Why?
What is Wrong with the Plots?

- Bars for different replication factor are not distinguishable.
- X-axis not totally clear.

*Student does not provide any discussion, explanation or mention of the graph.*
"We see how the write percentage affects performance. For the no replication configurations the small variations may be due to the experimental error; for the full replication configurations the performance drop is clearly visible when increasing the percentage of write operations. We can only say that, as expected, the percentage of write operations affects performance the most for full replication because of the high traffic of set requests."

Just describes the Graph with no explanation related to the system properties
Figure 15: Aggregated total mean middleware throughput of both operation types. For the response times, the colored bars are the median and the soft shaded bars indicate the 95th percentile.
A Better Explanation to the Graph

• “From previous experiments we concluded that a SET takes longer to process in our implementation than a GET. Therefore it is expected that as we increase the percentage of sets the median response time also increases. As shown in the figure, the 95th percentile stays stable for the un-replicated case, but increases slightly for the replicated one. This comes from the fact that not all servers respond in the same time when the system is more loaded. This is not because it is trashing. The overall throughput decreases with more writes (higher response time than for gets) and for larger number of servers (in the un-replicated case this is a result of the design decision in Section [...]). But the drop is more significant once full replication is used. The system is overall stable but combining slow SETs with replication significantly increases the response time for SETs, hence leading a significant drop in throughput”.