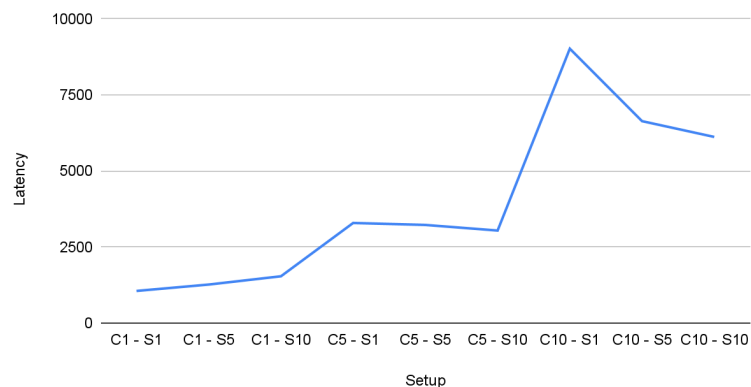


Performance Report

For the performance test, 10 different mail groups such as maildir\allen-p\all_documents are used as random put and gets. These documents are sent as serial puts first with randomness between clients and read by the same clients in random order. The latency unit is microseconds per k-v pair. The throughput unit is kv-pair per second.

Latency wrt Server-Client pairs



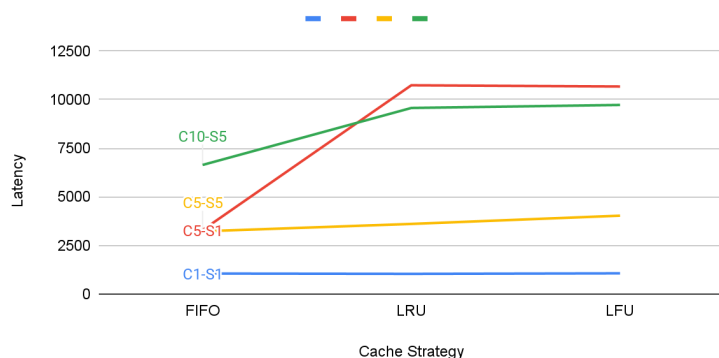
Latency wrt Server-Client Nodes

FIFO - Cache 100



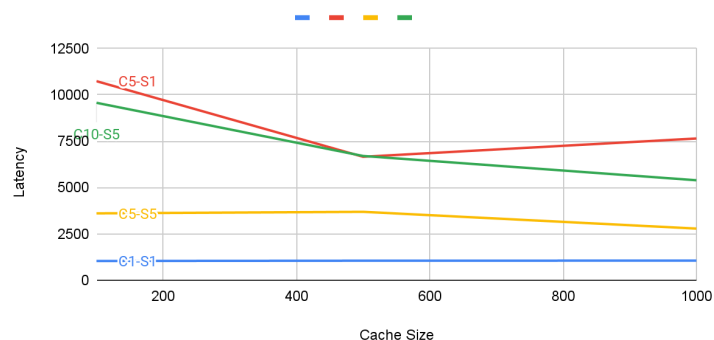
Latency wrt Cache Strategy

Cache Size 100



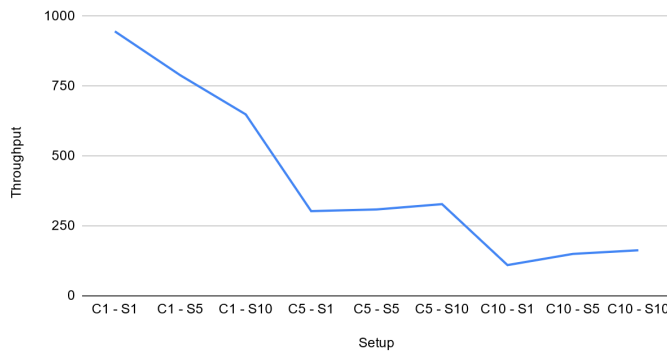
Latency wrt Cache Size

LRU

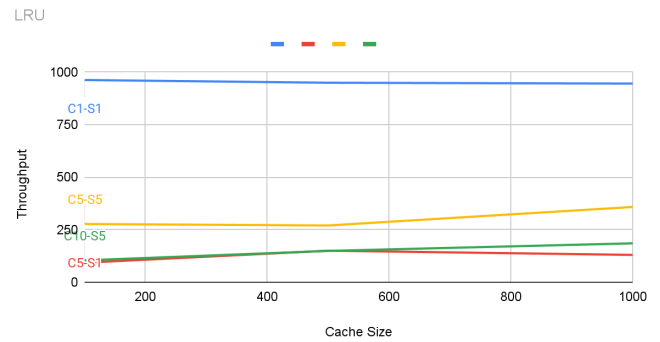


For the **Latency wrt Server-Client pairs** and **Latency wrt Server-Client Node** graphs, we can see that our latency is increasing in direct proportion to the client number when the server number is stable. This situation is expected because the workload of every server is increasing as well. Also, when the client number is stable but the server number is increasing, latency grows because of the disconnection-connection costs. In the **Latency wrt Cache Size** graph, we can see that latency is decreasing when the cache size is increasing as expected. However, after some point, latency is starting to increase because of a lack of memory. This happens because of running all servers on a single machine. In the **Latency wrt Cache Strategy** graph, it is seen that when we have just one server, the cache strategy doesn't have an impact on latency because, in performance tests, we don't call get for a value more than one time. Therefore, it always retrieves the value from the disk.

Throughput wrt Server-Client pairs

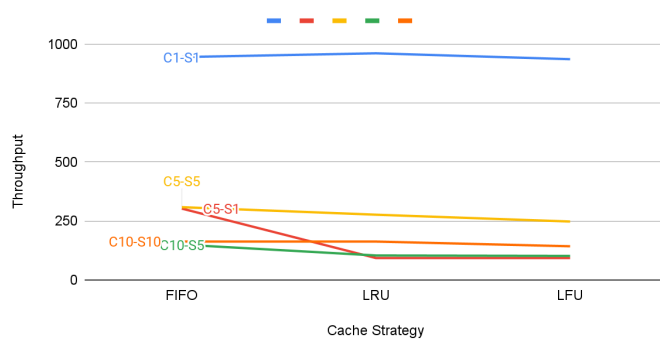


Throughput wrt Cache Size



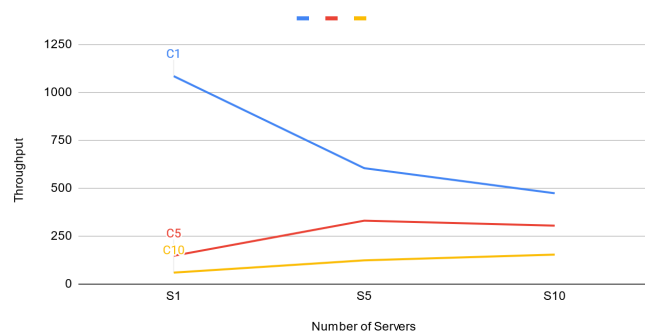
Throughput wrt Cache Strategy

Cache Size 100



Throughput wrt Server-Client Nodes

FIFO - Cache 100



For the throughput performance of the KV-Server, we can say that if the number of clients increases the throughput decreases since the number of streams that KV-Server has to deal with increases. However, as we talked about in the latency when the number of clients stays the same but the number of servers passes the number of servers, the disconnection-connection cost decreases the number of packets that can be sent. In terms of cache size, the cache size doesn't matter for the small number of nodes since they aren't using the same key for "get" commands in our performance tests in the same client. On the other hand, we can observe the increase of throughput by cache size in 5 servers since a key can be pulled from disk to cache by another client beforehand.