## CLOUD DATABASES SCALABLE STORAGE SERVER PEFORMANCE REPORT

#### PERFORMANCE WITH INCREASE IN CLIENTS:

From the first graph we can infer that as the number of clients increases the read and write time also, increases, as the server has to handle more threads (write generally takes more time than read). The increase is not so significant in this scenario since 20 clients is still a manageable number.

From the above graph we can see that the time for adding and removing nodes increases with the increase in the number of client just because of the multiple threads the servers then have to handle.

#### PERFORMANCE WITH INCREASE IN SERVERS:

With the increase in the number of servers, the read and the write time mainly remains the same with a slight decrease in both, as this time depends upon how well the hash values of the keys and the servers are distributed for the load to be balanced and thus take considerably less time.

Increase in servers greatly decreases the add and remove node times as the data to be moved from server to server is decreased and hence, the node is added or removed fast.

#### PERFORMANCE WITH INCREASE IN CACHE SIZE:

From the above graph, we can see that increase in cache size, while does not affect the write operation, greatly influences the read operation as the time for reads decreases with the increase in cache size, since the key value pair probability of being retrieved from the cache instead of the database increases.

#### PERFORMANCE WITH DIFFERENT CACHE STRATEGIES:

The graph demonstrates that the cache strategy, like before, really doesn’t affect the write operation. However, for the read operation, LRU fares the best, followed by LFU and then FIFO, for this particular scenario. The speed for the read operations in terms of strategy mainly depends upon the access order. If the access order is random then any which one could perform better, however if we tend to access again the elements that we just recently used, then LRU would perform better. If we access a select few elements continuously then LFU would do good and FIFO would be best if we don’t use elements that were sent in a long time ago. Therefore, we need complete knowledge of the future to know which strategy would fare best.