

Abstract

Database as a Service (DBaaS) is transactional database model which has the advantages including: the cost expended by the users is cheaper because they do not need to buy the hardware; and they do not require a lot of resources to manage it. Generally, since DBaaS is frequently accessed by many users at once, overload happens. One way to prevent it is to divide them to multiple servers. This process is known as load balancing. In this study, the static load balancing with random and round robin algorithm is implemented. In the era of modern technology, its implementation requires correct design. One example of architecture that support database load balancing is Shared-nothing cluster (SN-cluster). The results of the study showed that randomized load balancing with SN-Cluster can improve DBaaS system performance in response time, throughput and error rate on select and insert operation, while the round robin load balancing with SN-Cluster can improve system performance in update and delete operation. On the other hand, load balancing with SN-Cluster can handle concurrency for 100 and 200 clients, but will not be able to handle consistency for all number of clients.

Keywords : DBaaS, over-load, load balancing, shared-nothing cluster, response time, throughput, error rate, concurrency, consistency.