Paper Summary

The paper “Evaluation of container orchestration systems for deploying and managing NoSQL database clusters” is about deploying and managing NoSQL database cluster on container orchestration systems where MongoDB have been chosen as the NoSQL database. The Author has described features of CO systems relevant to DB clusters, the modifications to be done while using DB instances with CO systems, and finally evaluates the performance of the system while using MongoDB as the NoSQL database.

Container Orchestration systems are a necessity for functionalities like resilience, resource cost efficiency, and elasticity when there is a large-scale system. Some of the features of CO systems useful for a database are: (i) customizable scheduler that is managed by user-specified placement constraints, (ii) automated node failure detection and container eviction, (iii) persistent volume for storing DB instances, (iv) virtual cluster IP address for containers, (v) stable service object on the client-side for each container, (vi) attach HostPorts to containers. These features are provided as built-in by K8s and Swarm.

The issue is that the built-in service proxy and replication scheme of CO system conflicts with database cluster system features like load balancing and replication. Hence four constraints need to be applied for container-based configuration of a database cluster, which are: (i) a specific IP address for each container, (ii) each database should be associated with a unique persistent volume, (iii) database instances should be created and destroyed in an ordered fashion, (iv) database instances need to be placed in different node but close to persistent volume node.

The overhead of the first three features of CO systems mentioned above is expected to be low and hence is not of concern. The overhead of the fourth and fifth features may increase performance overhead which may be eliminated by the sixth feature and a stable floating IP address. Hence, we evaluate and compare performances of different configurations and find answers to some of the research questions framed by the author in the paper.

The Author has used the YCSB benchmark to evaluate the performance of database clusters. In each deployment, the database cluster is set up with three database instances that store their state on a local directory of VM. Additional placement constraints are specified for a simple auto-recovery strategy. The deployment is configured in such a manner that the write operation would use at least two instances and read would use the invoked instance.

For the findings of Q1, the DockerOnly system has no significant performance overhead in the read operations but performs better than VM-based deployments in update operations. For Q2, the SwarmServiceIp deployment's performance overhead is exceptionally large on all workloads. The K8HostNameNodePort deployment performs best for update operations in read-heavy workloads. For RQ3, the SwarmServiceIp_overlay deployment performs best in update operations. For K8, there are no significant differences between the performances of Weave NET and the flannel plugin.

Hence, one of the conclusion is that databases with CloudProvisionedIp endpoints and HostPorts result in less performance overhead than ones with ClusterProvisioned endpoints.