COMP 512 : zookeeper project3

Distributed coordination using zookeeper ensemble

Team 06

MASTER

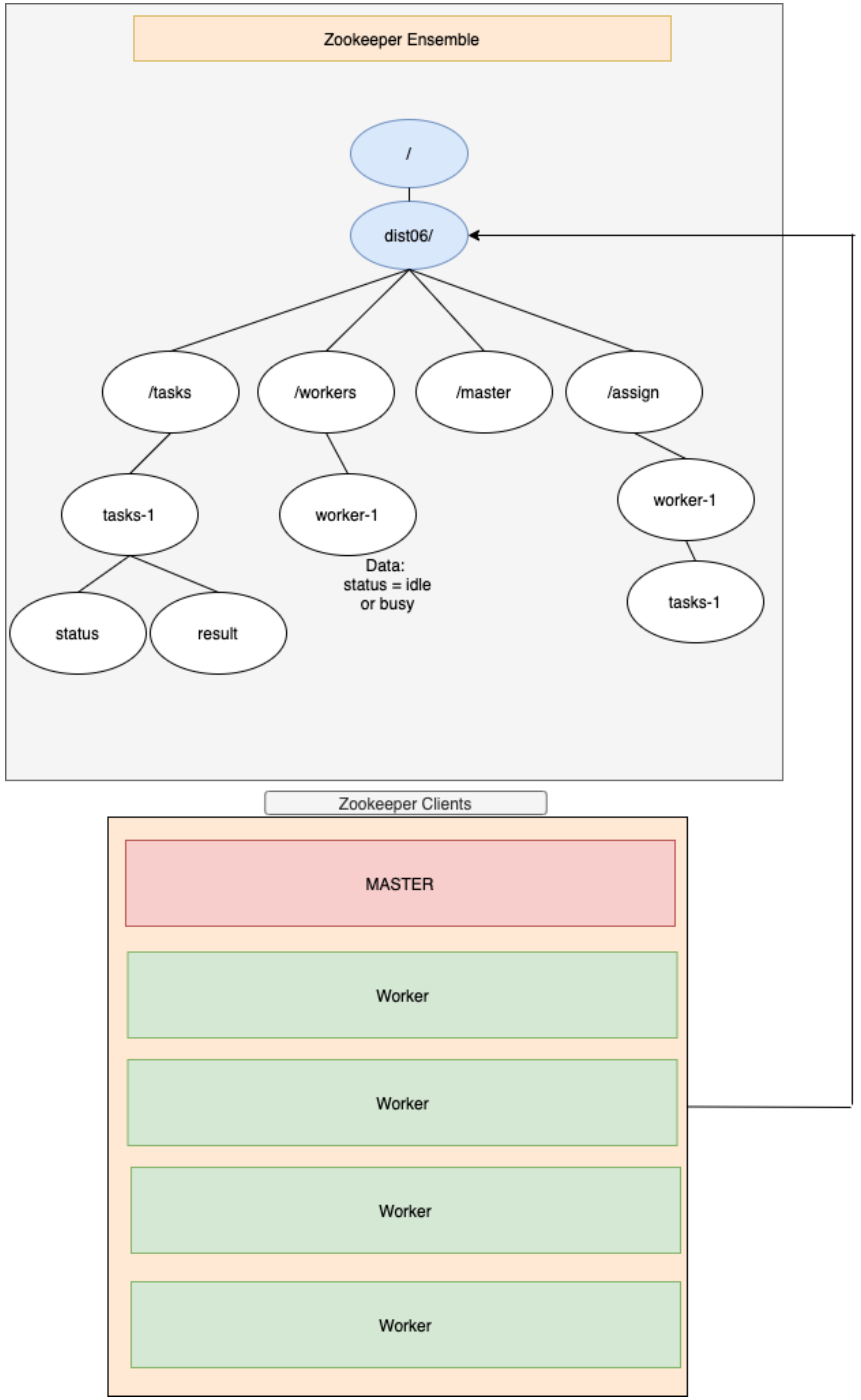
- Process declares itself as master by creating znode **/dist06/master**

- If master process process already up and running then new process will declare itself as worker and will create a new unique znode by creating **dist06/workers/worker-xx** where xx is worker ID

- master will change the status of the tasks to “assigned” after assigning the task to the worker
- This is done to prevent the reassigned of assigned tasks in case of failure during tasks assignment process

TASK ASSIGNENT

- after the worker is up and running it will update it’s status as “idle”
- Will create a znode **/dist06/assign/woker-xx/**
- Master will install a watch on **dist06/workers/** and check for status change for each worker
- As soon as the worker status = “idle” master will assign a tasks to that worker by creating and new znode **dist06/assign/worker-xx/tasks-x** and submit the task object in the tasks-x znode



WORKERS

- Worker-xx has watch installed under the znode **dist06/assing/worker-xx** so whenever worker is idle and a new tasks is assign to worker (i.e task-x) by master, worker-xx will be notified and it will change it status to “working “ and start the computation in different thread

WORKERS Cleanup

- after the result znode is created then worker-xx will delete the znode **/dist06/assign/worker-xx/tasks-x** node and install a new watch on the that znode and finally change it’s status back to idle to inform the master that’s worker process is ready for next round of execution

RESULT

- after the computation is done by the thread in worker will submit the result under the znode **dist06/tasks/task-x/result**
- Client will be notified (because of a watch) and client can safely consume the result and delete the tasks-xx node during shutdown (graceful cleanup)