**Zookeeper use cases:**

**Configuration:**

Init

zk.create(“/app/configs/configFile”, configFile)

The clients get the config files using:

configFile = getData(“/app/configs/configFile”, true)

if it not available the get will return null.

When admin update the configs:

setData(“/app/configs/configFile”,newConfigFile,v2) then clients will be notified to get the configuration file again.

**Group membership**

Zk.create(“/app/groups/group1”)

Clients add themselves to the group by

Zk.create(“/app/configs/configFile”/” + client\_id, client info, EPHERMAL)

Admin can get the members by

Zk.getChildren(“/app/configs/configFile”)

**Barriers**

Distributed systems use barriers to block processing of a set of nodes until a condition is met at which time all the nodes are allowed to proceed. Barriers are implemented in ZooKeeper by designating a barrier node. The barrier is in place if the barrier node exists. Here's the pseudo code:

* The client calls the ZooKeeper APIs **exists()** function on the barrier node, with watch set to true.
* If **exists()** returns false, the barrier is gone and the client proceeds.
* Else, if **exists()** returns true, the clients wait for a watch event from ZooKeeper for the barrier node.
* When the watch event is triggered, the client reissues the **exists( )** call, again waiting until the barrier node is removed.

1. Var res = Zk.exist(“…/barrier”, true)
2. If(!res) proceeds
3. Else waits for the notification and go to 1 again.