ExaVenger Managers are a group of 5-10 machines with replicated states backed by Zookeeper. We need to be careful with each manager machine’s local state and the states kept by Zookeeper, to ensure manager fail-over won’t cause any trouble.

Global State

1. Machine status: one line per server

Server name, ip, port, chain, status (dead, reserved, active), capacity, primary partitions, secondary partitions.

1. Partition table: one line per partition

Partition number, primary, secondary, tertiary, candidates.

1. Reserves list
2. Events log

Managers need to process the following requests:

P

S

Report Primary Timeout

lease

lease

Report Secondary Failure

Manager timeout

Inactive

Manager timeout

New Primary

1. New server becomes available.

Here the manager node would insert a “new server” event in the event log, then reply to the server saying we received this notification.

Next the manager would need to locate the server on the machine status table, and for each primary partitions, do a new primary selection. For each secondary partitions, do a secondary removal.

Todo!! Add new machines to these partitions, update machine table and partition table.

Then we add the machine to the reserved server list and remove the item from the event list

1. Primary Timed Out

Note that we only accept requests with a up-to-date ballot (partition config version).

A secondary partition would respond to periodical beacons or replications from the primary. If after a *lease period*, no beacons or replications received from the primary, it would report “primary timeout” to the manager, and keep doing that unless beacon from primary is received again or partition configuration changed.

If within a window of say half second, 10 different machines complains about primary timeout regarding the same server ( or maybe one complained about primary timeout and other 9 about secondary timeout, we need an algorithm for that, leaky bucket?), the manager will deem the server as “dead”, and insert “machine dead” into the event log.

Then the manager need to send a kill signal to the server (wait for manager-time-out limit if there is no response). Next the manager assigns new primary to each of the primary partitions on that machine, and remove secondary to each of the secondary partitions. TODO!! Add new machines to these partitions.

Then we remove the item from the event list.

The earliest time a replica’s primary status can be taken away from it without its knowledge (because of network problem), is (lease + manager\_timeout) after the last time the reporting secondary replica received beacon from that seemingly problematic primary replica.

This can be sooner if the manager successfully send a “kill” command to the old primary, thus telling it to give up the primary status sooner.

In the case of very few reports about a primary timeout, the manager would do nothing, thinking the machine is healthy. Since there were timeouts, the primary replica would send “secondary failure” (see below) resulting the secondary replica’s be moved to another machine.

1. Secondary Failure

A primary replica would send periodical beacons or replications to all secondary replicas in the same partition. If one secondary does not reply after a *lease period*, or respond with an outdated decree. We send a “secondary failure” request to manager. Keep sending it until configuration change, or the secondary come back online with up to date decree.

If the primary replica do not get response from the manager after manager-timeout, it would give up primary status and stop processing read/write request.

When the manager receive the “secondary failure”, it insert into event log, and reply to the reporting server to shut it up. Then do a secondary removal of that partition. TODO!! Add new machine to that partition.

Then remove the item from the event list.

So the latest time the primary would give up its primary status, is (lease + manager\_timeout) after the starting point of the last successful communication with the problematic replica. Note this time should always be later than its primary status be taken away by the manager. So the Single primary still holds.

We may also consider letting the server restart if over half of its replica is removed from it.

We need to ensure fail-over safe, which means that we need to tolerate the loss of local manager state. So each action can be repeated without causing any harm. And request 1 and 3 needs to be repeated until an ack is received. In the case of 2, the server needs to keep complaining about time out until reconfiguration.