Overview

Unlike most roles in the cluster, coordinators must be configured statically, so selecting them requires more careful consideration. Coordinators carry state, and this state is not automatically re-replicated if a coordinator is lost. This means that you must select enough coordinators to fulfill your desired fault tolerance. Coordinators are also quorum-based, which means that you need additional coordinators to ensure that a quorum is available. Losing a majority of your coordinators will take the database completely unavailable. On the other hand, having more coordinators than you need can be a problem for performance and operations, so you want to avoid recruiting additional coordinators if it does not provide significant marginal benefit.

Upper Bounds on Coordinators

Coordinators are responsible for three things and all of these would become significantly slower if you have a large number of coordinators:

1. They elect the cluster controller (cluster controller is the leader of a cluster). This election process is done in an iterative way: election takes a short amount of time but if it fails, this election time is increased. Having more coordinators means that the probability of the election failing increases. Therefore you would need more iterations. I would imagine that with 100 coordinators election would be very very slow.
2. They store a global state. During recovery this state is read, locked, and rewritten to all coordinators. If the system has more than one master running, one of them will eventually fail. But as this protocol involves several round-trips to all coordinators, it would slow down the process. Therefore, your recovery times will probably go up significantly if you have many coordinators.
3. Clients connect through coordinators. This path is optimized to prevent clients from keeping unnecessary connections, but in the worst case a client will always need to talk to all coordinators. Therefore having many coordinators will make connection establishment slower.

Recommended Coordinator Counts

For a single-DC config with replication factor `R`, you want `2*R-1` coordinators. The expected fault tolerance with this configuration is `R-1`. Having `2*R-1` coordinators means that after losing `R-1` you will still have a majority of the original coordinators available, which is required for the database to be available.

For multi-DC configs, you will likely want to be resilient to at least 1 DC and 1 other machine going down simultaneously. Losing a DC is likely to be a failure mode that takes longer to recover from, and you will want some additional fault tolerance while you are in this failure mode. For these kind of configurations, we recommend 9 coordinators, spread evenly across at least 3 DCs.

This would ensure that no DC has more than 3 coordinators. In the event that you lose 1 DC and 1 machine, you would lose at most 4 of your 9 coordinators, leaving you with a majority available.

If you are only storing data in two DCs, we recommend that you provision 3 processes in a third data center to serve as coordinators.

In three data hall mode, we also recommend 9 coordinators, by the same logic.

Replication Mode	Coordinator Count
single	1
double	3
triple	5
three_data_hall	9
three_datacenter	9
multi-region	9

Determining Which Processes to Use

There are two things that you should guarantee when determining which processes will serve as coordinators:

- Every coordinator should be on a different fault domain, and thus have a different zoneid.
- Coordinators should be spread as evenly as possible across data centers and data halls.

These two rules should ensure that your cluster maintains its fault tolerance goals.