Master vs Region Server, Current State, and Where We Are Going

With HBASE-10569 (meta co-locates with master), it is possible to simplify HBase deployment a lot. However, it also causes some confusion/misunderstanding. In this doc, I list out some important facts as where we are currently, and some possible enhancements we can do.

Current State

1. All masters are also region servers

Since the active master needs to host meta and some other system regions, it needs to speak the region server protocol, so it is also a region server.

2. A normal region server can't become a master

If a region server is started as a normal region server, it can't become a master since some master objects are not loaded.

One reason for this is to be backward compatible. We don't want to surprise users that a region server becomes a master suddenly.

3. Backup master vs normal region server
    a. At runtime, backup master = region server + one thread trying to become the next active master + one redirect web UI server listening to the original master info port (if enabled)
    b. Backup master can be configured to host regions just like a normal region server, or less, or none at all. This is for backup compatibility.
    c. Backup master can be the next active master, while a region server can't.

4. Meta is always on the active master

I filed HBASE-10923 so that we can put meta not on master via configuration. But I closed it as Won't Fix to avoid extra configurations. If we need to put this feature in 1.0.0, we may need to get this fixed and do proper configuration in 1.0.0. Matteo suggested not to fix it, just change the code a little in 1.0.0, which works too. My concern is that (1) this may prevent user trying to put meta on the active master, (2) is this against our open source policy?

5. No API change, or wire protocol change

Function-wise, no change to break compatibility, rolling-upgrade/restart. Clients look for the master from ZK as before.

6. Deployment change should be aware

By default, there are big deployment impacts since master is going to serve regions. We can configure backup master not to host any regions by default. But we can't configure not to put any region on the active master yet, without HBASE-10923, or a similar change.

7. No scripts change. Master is master, region server is region server

There is no change to any start/stop scripts. So starting up a master, it remains the same to start up an active/backup master; starting up a region server, it remains the same to start up a region server that won't become a master.

Where We Are Going

1. All masters

One suggestion is to make all region servers masters so that any one of them can be a master if needed. So there is no difference between a master and a region server. They run the same code. In a HBase cluster, all nodes are the same. We can call it simply HBase node instead of master/region server, or something similar.

One simple fix is to change the code so that we won't be able to start a normal region server any more. Starting a region server essentially starts up a (backup) master.

In this solution, we still rely on ZK to select the active master.

2. Master quorum

Instead of making all nodes to be masters, we can have a collection of nodes that can be masters. Other nodes are still normal region servers. The collection of nodes that can be masters form a quorum. Instead of ZK quorum, we use master quorum.
Clients find out the active master from the master quorum instead of the ZK quorum. This is one more step further for HBase to be independent of ZK.

If all nodes could be masters, the quorum may be too big to be efficient.