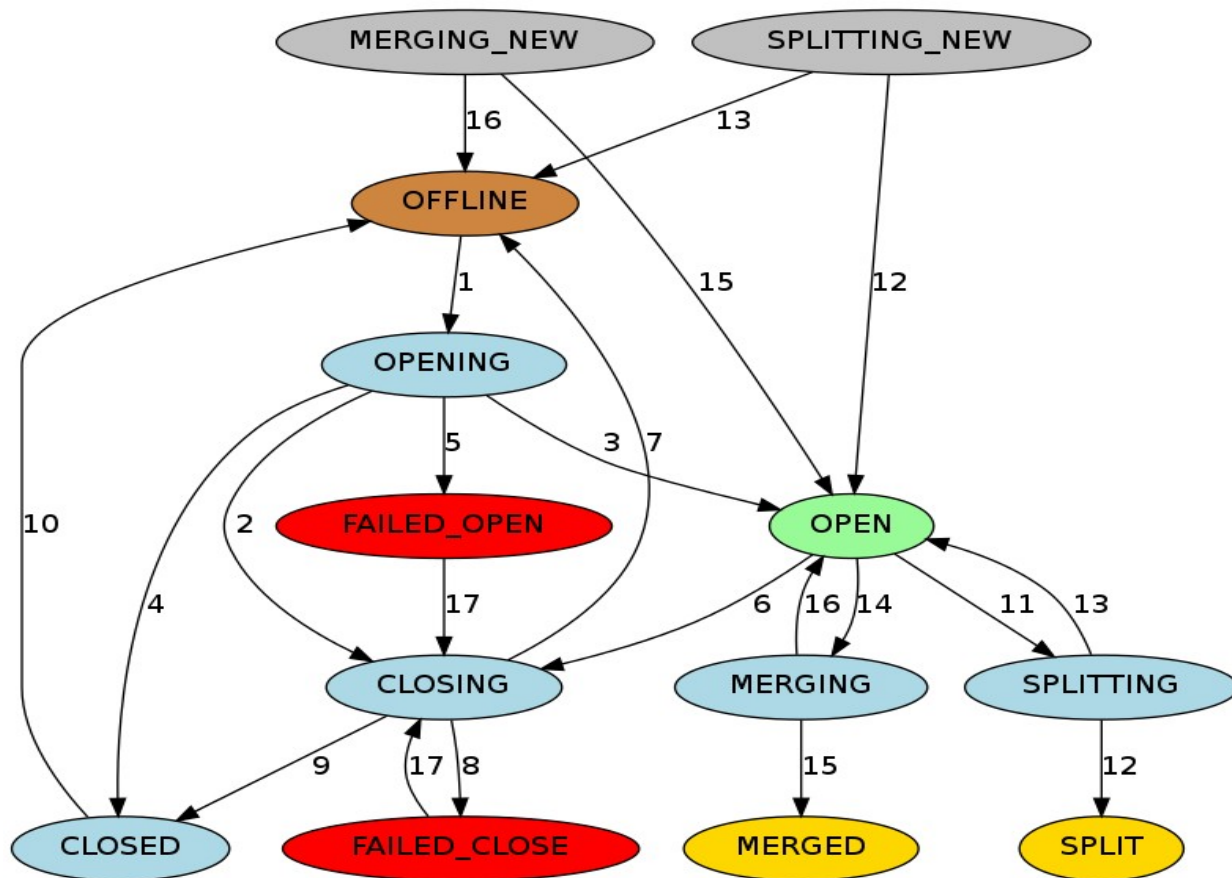


## Region State Transitions



Offline state, a special state that can be transient (after closed before opening), terminal (regions of disabled tables), or initial (regions of newly created tables)

- **OFFLINE:** Region is offline and not opening

Online state that regions can serve requests

- **OPEN:** Server opened region, notified master

Transient states

- **OPENING:** Trying to open a region but not done yet
- **CLOSING:** Trying to close a region but not done yet
- **CLOSED:** Server closed region, notified master
- **SPLITTING:** Server notified master to split a region
- **MERGING:** Server notified master to merge two regions

Failure states that need OPS attention

- **FAILED\_OPEN:** Failed to open a region
- **FAILED\_CLOSE:** Failed to close a region

Terminal states of regions split/merged

- **SPLIT:** Server completed split of a region, notified master
- **MERGED:** Server completed merge two regions, notified master

Initial states of regions created through split/merge

- *SPLITTING\_NEW*: New regions to be created when a server splits a region
  - *MERGING\_NEW*: A new region to be created when a server merges two regions
1. Master moves a region from OFFLINE to OPENING, tries to assign a region to a server. The server may or may not have received the open region request. Master retries to send the open region request to the server till the RPC goes through or the master runs out of retries. After got the open region request, the server starts to open the region.
  2. If running out of retries, the master makes the sure the server isn't opening the region by moving the region to CLOSING and trying to close it, no matter if the region is opening or not on the server.
  3. After the server opens the region, it keeps trying till notifies the master. The master moves the region to OPEN when notified. Now the region is open.
  4. If the server can't open the region, it notifies the master. The master moves the region to CLOSED and tries to open the region on a different server.
  5. If the master can't open the region for a certain number in a row, it moves the region to FAILED\_OPEN, and won't do anything further until OPS does something from the HBase shell, or the server is dead.
  6. Master moves a region from OPEN to CLOSING. The server holding the region may or may not have received the close region request. Master retries to send the close request to the server.
  7. If the server is not online, or throws NotServingRegionException, master moves the region to OFFLINE and re-assigns it.
  8. If the server is online, but not reachable after running out of retries, master moves the region to FAILED\_CLOSE, and won't do anything further until OPS does something from the the HBase shell, or the server is dead.
  9. If the server gets the close region request, closes the region, and notifies the master. The master moves the region to CLOSED and re-assigns it.
  10. Before assigning a region, the master moves the region to OFFLINE automatically if it is in CLOSED.
  11. When a server is about to split a region, it notifies the master at first. The master moves the region to be split from OPEN to SPLITTING, and add the two new regions to be created to the region states. These two regions are in SPLITTING\_NEW initially.
  12. After notifying the master, the server starts to split the region. Once passed PONR, the server notifies the master again so the master can update the meta properly. However, the master updates the region states only after is notified by the server the split is done. If the split is succeeded, the splitting region is moved from SPLITTING to SPLIT and the two new regions are moved from SPLITTING\_NEW to OPEN.
  13. If the split is failed, the splitting region is moved from SPLITTING back to OPEN, and the two new regions to be created are moved from SPLITTING\_NEW to OFFLINE.
  14. When a server is about to merge two regions, it notifies the master at first. The master moves the two regions to be merged from OPEN to MERGING, and add the new merged region to be created to the region states. The new region is in MERGING\_NEW initially.
  15. After notifying the master, the server starts to merge the two regions. Once passed PONR, the server notifies the master again so the master can update the meta properly. However, the master updates the region states only after is notified by the server the merge is done. If the merge is succeeded, the two merging regions are moved from MERGING to MERGED and the new region is moved from MERGING\_NEW to OPEN.
  16. If the merging is failed, the two merging regions are moved from MERGING back to OPEN, and the new region to be created is moved from MERGING\_NEW to OFFLINE.
  17. For FAILED\_OPEN/FAILED\_CLOSE region, the master tries to close it again upon re-assign by OPS from HBase shell.