[HBASE-6721] Hbase Table Isolation for Multi-tenancy using Region Server Grouping

April 26, 2013

Francis Liu
Vandana Ayyalasomayajula

Motivation:

HBase table isolation is required for scenarios that involve multiple users sharing a common HBase instance. We want to isolate the impact of usage (like read,write patterns) and maintenance activity (like compaction, region splits) of different applications. In the current state, there is no way we can control what regions of a table get assigned to given region servers. A feature to group region servers and assigning tables based on groups will be very useful in multi-tenant environments. In this way, tables belonging to different applications can be assigned to different region server group and hence each application behavior can be isolated to good extent. This design is similar to existing work done in the JIRA HBASE-5169. The major difference being usage of command line to manage region server groups and the restriction that a table can only belong to a single region server group at a given time.

Proposal

The work related to this feature can be split into two major components:

1. Developing API to execute commands related to region server grouping.
2. Developing group based load balancer.

Region Server Group API

To provide commands through hbase shell to do the following:

- Create/Delete region server groups.
- Move region servers between groups.
- Move HBase tables between groups.

The following APIs will be required for operations related to grouping region servers and assigning region server groups to tables.

- NavigableSet<String> listTablesOfGroup(String groupName) throws IOException;
- GroupInfo getGroupInfo(String groupName) throws IOException;
- GroupInfo getGroupInfoOfTable(String tableName) throws IOException;
- void moveServers(Set<String> servers, String targetGroup) throws IOException;
- void moveTables(Set<String> tables, String targetGroup) throws IOException;
- void addGroup(String name) throws IOException;
- void removeGroup(String name) throws IOException;
- List<GroupInfo> listGroups() throws IOException;
- GroupInfo getGroupOfServer(String hostPort) throws IOException;
- Map<String, String> listServersInTransition() throws IOException;
- boolean balanceGroup(String name) throws IOException;

We plan to implement all the above functionality using a coprocessor end point and a custom LoadBalancer.

Similar to the existing hbase commands, the above api's can be called using command line interface. If the HBase instance is configured to not use the region server grouping features, appropriate error messages will be shown to the user.

Group Based Load Balancer

The load balancer is the component of HBase which decides the placement of regions onto region servers. The region server grouping feature essentially aims at a different logic than the existing one. The present "Default Load Balancer" assigns randomly a given region to a region server. The group based load balancer will assign the regions based on region server groups. One salient point is that the Group Balancer delegates actual balancing logic to a real balancer (DefaultLoadBalancer by default). The GroupBalancer merely tailors the set of regions and severs handed down to the real balancer by group.

During balancing of the cluster, if any regions servers have regions which do not belong to the group to which the region server belongs to, then the regions will be get unassigned and hence will not appear in the regions plans produced by the "balanceCluster" method. These unassigned

regions, will then get assigned to correct region servers in the next round of assignment.

General Points:

- All group information will be stored in the “0group0” table. The table is purposely named this way so that it will be the first user table that gets assigned. When HBASE-8015 goes in we can migrate this to become a system table.
- There will be one region group to begin with: **default**
- All the online region servers belong to the “Default” group, if they are not assigned to any group.
- As the groups will be created, region servers will be moved from default group to respective groups.
- Each table can be assigned to only **one** existing region server group.
- All the tables which do not have a region server group assigned will be considered assigned to the “Default” group.
- The “Default” group cannot be deleted.
- In line with the current behavior, balancing will not include the catalog tables.

HBase Shell Commands

The following command will be added to the existing hbase commands:

1. group_add ‘group_name’
2. group_remove ‘group_name’
3. group_move_servers 'dest',['host1:port','host2:port']
4. group_list
5. group_move_tables 'dest',['table1','table2']
6. group_get ‘group_name’
7. group_of_table ‘table_name’
8. group_of_server ‘regionserver_name’
9. group_list_transitions
10. group_balance ‘group_name’
11. group_list_tables 'default'

Required hbase-site.xml configuration

hbase.coprocessor.master.classes=org.apache.hadoop.hbase.group.GroupMasterObserver,org.apache.hadoop.hbase.security.access.GroupAdminEndpoint

hbase.master.loadbalancer.class=org.apache.hadoop.hbase.group.GroupBasedLoadBalancer

References:

- https://issues.apache.org/jira/browse/HBASE-4120
- https://issues.apache.org/jira/browse/HBASE-5169