Lucene Index Creation on Existing Region

Overview

Allow users to add a Lucene index on a region that already exists and contains data. We also want to simplify the process to modify a Lucene index on a region. To modify an index, the user will need to first destroy the existing Lucene index and then add a new index, without having to destroy and recreate the data region, which is required today. Lucene indexes are currently only supported on partitioned regions.

Goals:

1. Lucene Index can be created before or after a data region has been created
2. Support an active cluster (puts in flight) when adding Lucene index
3. Index both existing data in region as well as new data events
4. Be able to add Lucene index from gfsh and have it stored in cluster config
5. Can handle HA events (members dying, new members added, rebalancing)
6. Need a public Java API to do distributed creation of Lucene Index
7. Queries on indexes that are in the middle of being initialized should throw an exception
8. Backward compatibility - adding this feature should not break apps using existing Lucene index creation flow: 1. create Lucene index, 2. create region.

Not in Scope

1. Modify a Lucene index on-the-fly; user will need to delete existing Lucene index and create a new one. This means queries will return an exception when old Lucene index is deleted and before the new Lucene index is created.

Approach

Our current design approach is as follows:

1. User initiates a create lucene index command from GFSH or a Java API
    a. A function is sent to all members in parallel that does the following:
        i. Create AEQ
        ii. Create index region
        iii. Add AEQ and listener to region
        iv. Return xml to be written to cluster config (if cluster config is enabled)
    b. Send xml to locator to be written into cluster configuration (if cluster config is enabled)
    c. Another function is sent to all members in parallel to:
        i. Set indexRepositories()
        ii. Modify computeRepo() to look for COMPLETE file; if doesn't exist
            1. Iterate and index existing region data
            2. Add COMPLETE file to fileAndChunkRegion
2. Query:
    a. Check for COMPLETE file
    b. If COMPLETE file not there, start async task to execute computeRepo(), and throw an exception back to the query caller to let them know the index is not yet ready

API Change

1. A new Java API to create the Lucene Index in a distributed manner. This Java API relies on a new Management API that does not currently exist in Geode.

Risks and Unknowns

1. Impact to memory usage after AEQ is added to each member, collecting events but not dispatching them until the index addition process is complete and the AEQ is unblocked.
2. Management API to add the API to create the lucene index does not exist at this time.