Item 78

git_comments:

1. NativeS3FileSystem.rename won't overwrite, so we might need to delete the old index first
2. 256KB
3. * * Rename the files. This works around some limitations of both FileContext (no s3n support) and NativeS3FileSystem.rename * which will not overwrite * * @param outputFS The output fs * @param indexZipFilePath The original file path * @param finalIndexZipFilePath The to rename the original file to * * @return False if a rename failed, true otherwise (rename success or no rename needed)
4. This takes a long time due to retries

git_commits:

1. **summary:** Merge pull request #1434 from metamx/fix1433
   message: Merge pull request #1434 from metamx/fix1433 Wipe FileContext off the face of the earth

github_issues:

github_issues_comments:

github_pulls:

1. **title:** Wipe FileContext off the face of the earth
   body: - Fixes https://github.com/druid-io/druid/issues/1433 - Works arround https://issues.apache.org/jira/browse/HADOOP-10643 - Reverts to the prior method of renaming

github_pulls_comments:

1. LGTM

github_pulls_reviews:

1. can we reduce the time between retries?
2. Right now the retries are static in the class. To have them configurable is a big ask here.
3. minor thing, If these are just static util methods can they be moved to JobHelper, so that they can be reused by other jobs when required ?
4. This is in JobHelper
5. it seems a little weird to use the DataPusher interface here. RetryUtils might be cleaner so we don't have to abuse the method to return a boolean.
6. Fixed

jira_issues:

1. **summary:** Add NativeS3Fs that delegates calls from FileContext apis to native s3 fs implementation
   description: The new set of file system related apis (FileContext/AbstractFileSystem) already support local filesytem, hdfs, viewfs) however they don't support s3n. This patch is to add that support using configurations like fs.AbstractFileSystem.s3n.impl = org.apache.hadoop.fs.s3native.NativeS3Fs This patch however doesn't provide a new implementation, instead relies on DelegateToFileSystem abstract class to delegate all calls from FileContext apis for s3n to the NativeS3FileSystem implementation.

jira_issues_comments:

1. Added the implementation along with test case
2. this could be useful, when ready. We're looking for all our future S3 support -other than bug fixes- to be on HADOOP-10400 and the proposed new s3a filesystem, in a new module {{hadoop-tools/hadoop-aws}} # Can you look at the code proposed for HADOOP-10400 and build on that? And if not -what's wrong with that patch? # all new code will need to go into the proposed new module, which will have tests that only run if the relevant authentication details are provided. This gives you an opportunity to write contract compliance tests for the new FS

3. Sumit -why did you cut a check from {{AbstractFileSystem}}? That could have implications elsewhere. This should be retained, otherwise we've added a new way for things to fail, different error messages, etc. Can you add a new protected method to validate the URI; something which does the check —and which your code overrides if it really, really doesn't want a port to be specified. That won't impact anything else.
4. [~stevel@apache.org] I'm assuming your concern is on this portion of the patch (as part of AbstractFileSystem.java): {noformat} - // A file system implementation that requires authority must always - // specify default port - if (defaultPort < 0 && authorityNeeded) { - throw new HadoopIllegalArgumentException(- "FileSystem implementation error - default port " + defaultPort - + " is not valid"); - } {noformat} If so, s3's urls have a specific requirement that they don't contain any port (so defaultPort becomes -1 in this case) and they don't have any authority in the url as well. Does this work?
5. This request was originally for s3n. The functionality it requests is provided for s3a in HADOOP-11262. Can this be closed?
6. I see, feel free to resolve with appropriate closure code.