Item 142
git_comments:

1. fix length to relevant portion of har block
2. each line contains a hashcode range and the index file name
3. get all part blocks that overlap with the desired file blocks
4. the fields below are stored in the file but are currently not used by HarFileSystem permission = new FsPermission(Short.parseShort(propSplits[1])); owner = decodeString(propSplits[2]); group = decodeString(propSplits[3]);
5. close the master index
6. offset 1 past last byte of desired range
7. we got the right har Path- now check if this is truly a har filesystem
8. desired range includes beginning of this har block
9. propSplits is used to retrieve the metainformation that Har versions 1 & 2 missed (modification time, permission, owner group). These fields are stored in an encoded string placed in different locations depending on whether it's a file or directory entry. If it's a directory, the string will be placed at the partName location (directories have no partName because they don't have data to be stored). This is done because the number of fields in a directory entry is unbounded (all children are listed at the end) If it's a file, the string will be the last field.
10. * * Fix offset and length of block locations. * Note that this method modifies the original array. * @param locations block locations of har part file * @param start the start of the desired range in the contained file * @param len the length of the desired range * @param fileOffsetInHar the offset of the desired file in the har part file * @return block locations with fixed offset and length
11. pointer into the static metadata cache
12. desired range starts after beginning of this har block fix offset to beginning of relevant range (relative to desired file)
13. range ends before end of this har block fix length to remove irrelevant portion at the end
14. * * Get filestatuses of all the children of a given directory. This just reads * through index file and reads line by line to get all statuses for children * of a directory. Its a brute force way of getting all such filestatuses * * @param parent * the parent path directory * @param statuses * the list to add the children filestatuses to * @param children * the string list of children for this parent * @param archiveIndexStat * the archive index filestatus
15. * * Combine the status stored in the index and the underlying status. * @param h status stored in the index * @param cache caching the underlying file statuses * @return the combined file status * @throws IOException
16. decode the name
17. make it always backwards-compatible
18. the version is currently not useful since its the first version
19. * * @return null since no checksum algorithm is implemented.
20. offset 1 past last byte of har block relative to beginning of desired file
21. close the archive index
22. the archive has been overwritten since we last read it remove the entry from the meta data cache
23. offset of part block relative to beginning of desired file (may be negative if file starts in this part block)
24. the first line contains the version of the index file
25. make it a har path
26. check for existence of 3 part files, since part file size == 1
27. check block size for path files
28. check for existence of only 1 part file, since part file size == 2GB
29. check bytes in the har output files
30. test archives with a -p option
31. * * check if the block size of the part files is what we had specified
32. fileb and filec
33. * now try with different block size and part file size *
34. assuming all the 6 bytes were read.
35. *the size of the part files that will be created when archiving *
36. read the rest of the paths
37. * * the filestatus of this object * @return the filestatus of this object
38. assuming if the user does not specify path for sources the whole parent directory needs to be archived.

39. check to see if relative parent has been provided or not this is a required parameter.
40. * * constructor for filestatusdir * @param fstatus the filestatus object that maps to filestatusdir * @param children the children list if fs is a directory
41. * HarEntry is used in the {@link HArchivesMapper} as the input value.
42. * * get rid of / in the beginning of path * @param p the path * @return return path without /
43. * the size of the blocks that will be created when archiving *
44. add all the directories
45. find all the common parents of paths that are valid archive * paths. The below is done so that we do not add a common path * twice and also we need to only add valid child of a path that * are specified the user.
46. * * truncate the prefix root from the full path * @param fullPath the full path * @param root the prefix root to be truncated * @return the relative path
47. * * this method writes all the valid top level directories * into the srcWriter for indexing. This method is a little * tricky. example- * for an input with parent path /home/user/ and sources * as /home/user/source/dir1, /home/user/source/dir2 - this * will output <source, dir, dir1, dir2> (dir means that source is a dir * with dir1 and dir2 as children) and <source/dir1, file, null> * and <source/dir2, file, null> * @param srcWriter the sequence file writer to write the * directories to * @param paths the source paths provided by the user. They * are glob free and have full path (not relative paths) * @param parentPath the parent path that you wnat the archives * to be relative to. example - /home/user/dir1 can be archived with * parent as /home or /home/user. * @throws IOException
48. * * the children list of this object, null if * @return the children list
49. the largest depth of paths. the max number of times * we need to iterate
50. * * set children of this object * @param listStatus the list of children
51. * size of blocks in hadoop archives *
52. * * A static class that keeps * track of status of a path * and there children if path is a dir
53. * size of each part file size *
54. just take some effort to do it rather than just using substring so that we do not break sometime later

git_commits:

1. **summary:** HADOOP-7539. merge hadoop archive goodness from trunk to .20 (John George via mahadev)
   message: HADOOP-7539. merge hadoop archive goodness from trunk to .20 (John George via mahadev) git-svn-id: https://svn.apache.org/repos/asf/hadoop/common/branches/branch-0.20-security@1163079 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
2. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
3. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
4. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.

    label: test
5. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
6. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
7. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
8. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
9. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
10. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
11. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
12. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
13. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
14. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
15. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.
16. **summary:** merge hadoop archive goodness from trunk to .20
   description: hadoop archive in branch-0.20-security is outdated. When run recently, it produced some bugs which were all fixed in trunk. This JIRA aims to bring in all these JIRAs to branch-0.20-security.

jira_issues_comments:

1. The following JIRAs were the most interesting ones, but it made sense to bring in most of the others as well, not only because a bunch of them are dependencies of the JIRAs that were needed, but also because it is easier to merge. MAPREDUCE-1425 :archive throws OutOfMemoryError MAPREDUCE-2317 :HadoopArchives throwing NullPointerException while creating hadoop archives MAPREDUCE-1399 : The archive command shows a null error message MAPREDUCE-1752 :Implement getFileBlockLocations in HarFilesystem
2. No one has proposed making any more releases out of branch-0.20. Can you generate a patch for the branch-0.20-security line?
3. Sorry Owen, I meant to say branch-20-security (not branch-0.20). Fixed "Description". The patch is also meant for branch-.20-security.
4. **body:** John, Since this is a big patch, can you please do some manual testing on a real cluster (could be a single node cluster)? Just run a archive job and then a map reduce job to use the archives as input and verify the results. That should suffice.
   label: test
5. Yes, I will run the manual testing and post the results here. I ran "ant test" and it failed the same test that failed without the patch. The results of test-patch is as follows: [exec] BUILD SUCCESSFUL [exec] Total time: 6 minutes 23 seconds [exec] [exec] [exec] [exec] [exec] +1 overall. [exec] [exec] +1 @author. The patch does not contain any @author tags. [exec] [exec] +1 tests included. The patch appears to include 6 new or modified tests. [exec] [exec] +1 javadoc. The javadoc tool did not generate any warning

messages. [exec] [exec] +1 javac. The applied patch does not increase the total number of javac compiler warnings. [exec] [exec] +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings. [exec] [exec] [exec] [exec] [exec] == [exec] == [exec] Finished build. [exec] == [exec] ==

6. Manual tests run: - created a har file as follows: - hadoop fs -put test /tmp - hadoop archive -archiveName test.har -p /tmp test /tmp - ran the following manual tests: - wordcount on a couple of har files - streaming on the same har file with: hadoop jar hadoop-streaming.jar -Dmapred.reduce.tasks=1 -input har:///tmp/test.har/test/aa -output /tmp/aaa.2 -mapper cat -reducer "wc -l" Both of the above jobs completed successfully and had outputs in the corresponding output directory.
7. The only issue I see is that hadoop archives that already existed on the cluster will become obsolete since the new archive code wont be able to read it?
8. Maybe we want to add a utility to upconvert from 1 to 3 version?
9. Looks like I might be wrong. The patch seems to be able to read the old har archives as well. John, mind testing it out?
10. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12490263/HADOOP-7539-1.patch against trunk revision . +1 @author. The patch does not contain any @author tags. +1 tests included. The patch appears to include 6 new or modified tests. -1 patch. The patch command could not apply the patch. Console output: https://builds.apache.org/job/PreCommit-HADOOP-Build/67//console This message is automatically generated.
11. 1. Create HAR file using version 1 {quote} $ hadoop fs -cat /tmp/thisis1.har/_masterindex 1 0 2127535165 0 1856 {quote} 2. Install version 3 of HAR {quote} $ hadoop fs -cat /tmp/thisis3.har/_masterindex 3 0 2127535165 0 2610 {quote} 3. Run ls and wordcount on VERSION 1 {quote} $ hadoop fs -ls har:///tmp/thisis1.har $ hadoop jar hadoop-examples.jar wordcount har:///tmp/thisis1.har/x.sh /tmp/out.2 {quote}
12. looks good to me. Ill run some ant tests and check it in the 0.20 security branch.
13. I just committed this. Thanks a lot John!
14. Closed upon release of 0.20.205.0