Item 147

git_comments:

1. We use a buffer of DEFAULT_BLOCKSIZE size. This might be extreme. Could maybe do with less. Study and figure it: TODO

git_commits:

1. **summary:** HBASE-3006 Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
   message: HBASE-3006 Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance git-svn-id: https://svn.apache.org/repos/asf/hbase/trunk@997968 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
   description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,

srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.

2. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
   description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging,

log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.

3. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
   description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:

/10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.
label: code-design

4. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16

09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO

org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.
label: code-design

5. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
   description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,

srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.
label: test

6. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
   description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src:

/10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768;

bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.
label: code-design

7. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:

blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.
label: code-design

8. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
   description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID:

DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO

org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.
label: code-design

9. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO

org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.

10. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
    description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336

```
INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 139000 {code}
```

This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk.

```
{code} 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711
decompressedSize = 92324 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768;
bytes: 11005 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773;
bytes: 1 2010-09-16 09:21:27,912 INFO
```

org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.

11. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
    description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid:

blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.

12. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
    description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:

blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775;

bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.

13. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
    description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by

createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.

14. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
    description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,

srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.

15. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
    description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was 64k, and on disk block size after compression should generally have been around 6k. Some print debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer. The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a 11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO

org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711 decompressedSize = 92324 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768; bytes: 11005 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.
label: code-design

16. **summary:** Reading compressed HFile blocks causes way too many DFS RPC calls severly impacting performance
description: On some read perf tests, we noticed several perf outliers (10 second plus range). The rows were large (spanning multiple blocks, but still the numbers didn't add up). We had compression turned on. We enabled DN clienttrace logging, log4j.logger.org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace=DEBUG and noticed lots of 516 byte reads at the DN level, several of them at the same offset in the block. {code} 2010-09-16 09:28:32,335 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:38713, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 203000 2010-09-16 09:28:32,336 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40547, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 119000 2010-09-16 09:28:32,337 INFO org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest: /10.30.251.189:40650, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800, srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid: blk_-4686540439725119008_1985, duration: 149000 2010-09-16 09:28:32,337 INFO

org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:40861, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 135000 2010-09-16 09:28:32,338 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:41129, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 117000 2010-09-16 09:28:32,339 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:41691, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 148000 2010-09-16 09:28:32,339 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:42881, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 114000 2010-09-16 09:28:32,341 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:49511, bytes: 516, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.3\ 0.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 153000 2010-09-16 09:28:32,342 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src: /10.30.251.189:50010, dest:
/10.30.251.189:51158, bytes: 3096, op: HDFS_READ, cliID: DFSClient_-436329957, offset: 39884800,
srvID: DS-1757894045-10.\ 30.251.189-50010-1283993662994, blockid:
blk_-4686540439725119008_1985, duration: 139000 {code} This was strange coz our block size was
64k, and on disk block size after compression should generally have been around 6k. Some print
debugging at the HFile and BoundedRangeFileInputStream (which is wrapped by
createDecompressionStream) revealed the following: We are trying to read 20k from DFS @ HFile layer.
The BounderRangeFileInputStream instead reads several header bytes 1 byte at a time, and then reads a
11k chunk and later a 9k chunk. {code} 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 34386760 compressedSize = 20711
decompressedSize = 92324 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386760;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386761;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386762;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386763;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386764;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386765;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386766;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386767;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34386768;
bytes: 11005 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397773;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397774;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397775;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397776;
bytes: 1 2010-09-16 09:21:27,912 INFO
org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397777;
bytes: 1 2010-09-16 09:21:27,912 INFO

org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397778; bytes: 1 2010-09-16 09:21:27,912 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397779; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397780; bytes: 1 2010-09-16 09:21:27,913 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### seeking to reading @ 34397781; bytes: 9690 {code} Seems like it should be an easy fix to prefetch the compressed size... rather than incremental fetches.
label: code-design

jira_issues_comments:

1. @Kannan You are my new hero. I love the debug trace above. You think though this will add up to the 10 seconds?
2. Stack: This was just a snippet of the trace. Some rows span a lot more blocks. I am now looking at this in hfile/Compression.java: {code} public InputStream createDecompressionStream(InputStream downStream, Decompressor decompressor, int downStreamBufferSize) throws IOException { CompressionCodec codec = getCodec(); // Set the internal buffer size to read from down stream. if (downStreamBufferSize > 0) { Configurable c = (Configurable) codec; c.getConf().setInt("io.file.buffer.size", downStreamBufferSize); } CompressionInputStream cis = codec.createInputStream(downStream, decompressor); BufferedInputStream bis2 = new BufferedInputStream(cis, DATA_IBUF_SIZE); return bis2; } {code} Off-hand don't understand all the params to the various Stream abstractions :) I tried passing the compressed size of the block to above function instead of the current value 0 (for downStreamBufferSize) but that didn't do it. Still digging, but if you know off-hand what setting to change, let me know.
3. **body:** stack: To clarify regarding your: <<I love the debug trace above. You think though this will add up to the 10 seconds? >> The snippet was for a much smaller row. The outliers we saw were on larger blocks. But this should just help even the small row case. Basically the cost of missing any block in the block cache when compression is turned on was quite steep right now (will do about 5x-10x more DFS RPCs than needed).
   label: code-design
4. **body:** Wrapping a BufferedInputStream around did it. {code} 2010-09-16 11:30:40,842 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 32060739 compressedSize = 6197 decompressedSize = 65800 2010-09-16 11:30:40,842 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### reading @ 32060739; bytes: 6197 2010-09-16 11:30:40,843 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 32066936 compressedSize = 6083 decompressedSize = 65658 2010-09-16 11:30:40,843 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### reading @ 32066936; bytes: 6083 2010-09-16 11:30:40,844 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 32073019 compressedSize = 5003 decompressedSize = 65708 2010-09-16 11:30:40,844 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### reading @ 32073019; bytes: 5003 2010-09-16 11:30:40,844 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 32078022 compressedSize = 4834 decompressedSize = 65700 2010-09-16 11:30:40,844 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### reading @ 32078022; bytes: 4834 2010-09-16 11:30:40,845 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 32082856 compressedSize = 6137 decompressedSize = 65566 2010-09-16 11:30:40,845 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### reading @ 32082856; bytes: 6137 2010-09-16 11:30:40,846 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 32088993 compressedSize = 4727 decompressedSize = 65766 2010-09-16 11:30:40,846 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### reading @ 32088993; bytes: 4727 2010-09-16 11:30:40,876 INFO org.apache.hadoop.hbase.io.hfile.HFile: ### fs read @ offset = 32093720 compressedSize = 5025 decompressedSize = 65760 2010-09-16 11:30:40,876 INFO org.apache.hadoop.hbase.io.hfile.BoundedRangeFileInputStream: ### reading @ 32093720; bytes: 5025 {code} and perf is screaming back again. We can debate about whether the buffer size for the BufferInputStream should be the compressedSize (read everything) or some uniform sized chunks like 16k to play well with JVM. But basically the fix seems to work really well. Will post a patch shortly.
   label: code-design

5. **body:** Patch submitted. Fixes perf problems we were seeing. Averages dropped from 1-2 seconds to 100ms for my test runs. No outliers > 10 seconds. Previously we had many, and some which took 40-50 seconds. Yet to run unit tests (will report back on that once it completes). Thoughts on what BufferedInputStream's buffer setting would be? The disadvantage of using compressedSize (as i n provided patch) might lots of varying length allocations. Does JVM prefer allocations to all be of a uniform size? If we set a different value, say 16k, it'll incur more DFS RPCs (but at least not silly 1 byte RPCs).
   label: test
6. **body:** Patch looks great to me (This code was robbed from tfile a long time back IIRC). My sense is that though this a short-lived allocation, allocating compressedSize is overdoing it and as you suggest, it can vary, possibly widely. If a cell was large, then we could have an hfile much larger than 64k. A buffer of 8 or 16k would just as well save against the 1 byte rpc reads. If you agree, I can add this in on commit (allocate a 16k buffer by default)?
   label: code-design
7. **body:** Rather than adding buffering, can we just avoid using the random-access API to DFSInputStream? If you use the other APIs, you will take advantage of the internal buffering of DFSInputStream instead of double buffering, and it should be faster.
   label: code-design
8. **body:** Looks good to me.. nice catch Kannan.. This will significantly improve read durations
   label: code-design
9. Caught up with Todd in IRC. Avoiding double buffering sounds good. But not clear what the alternate DFS api is. I am planning to try with min(16k, compressedSize);
10. if the objects are short lived, it doesnt matter the object allocation size. If they are used only for the lifecycle of reading a single block, i'd say size it to the compressed Size.
11. For our test case, min(16k, compressedSize) will be a no-op since our compressed sizes were almost always smaller than 16k. In our internal branch, we're planning to go with 64k. Thought, for large objects, 16k batching might be too small. Will upload the new patch.
12. Committed. Thanks for nice fix Kannan (I left it at 64k -- use the HFile.DEFAULT_BLOCKSIZE define instead -- thinking that less rpc'ing is a better saving than a bit of memory in local heap).
13. Adding to the latest 0.89
14. This issue was closed as part of a bulk closing operation on 2015-11-20. All issues that have been resolved and where all fixVersions have been released have been closed (following discussions on the mailing list).